ORACLE
SOLARIS

Oracleホワイト・ペーパー
2011年7月

Oracle Solarisゾーン・クラスタでの Oracle® Real Application Clustersの実行

Oracle Solarisゾーン・クラスタへのOracle RACのデプロイ 1
ゾーン・クラスタの概要 1
Oracle Solaris Clusterを使用するメリット 4
クラスタのメンバーシップ 4
ストレージ・フェンシング 5
擬似STREAMネットワーク・ドライバ**clprivnet** 5
一貫性のあるディスク・デバイスの名前空間 5
アプリケーション・コンポーネントの制御 5
幅広いエージェント・ポートフォリオ 6
ディザスタ・リカバリ・ソリューションの構築 6
ゾーン・クラスタ内のOracle RAC 6
Oracle Solaris ClusterとOracle RACの統合 7
Oracle RACリソースの分離 8
共通リソース・コンポーネント 9
Sun QFS共有ファイル・システム 12
Oracle UDLMおよびネイティブSKGXNのサポート 13
Oracle RAC用の構成ウィザード 13
例：ゾーン・クラスタによるOracle RACのホスティング 14
ハードウェア構成の例 14
個々のゾーン・クラスタ構成 15
Oracle SolarisおよびOracle Solaris Clusterソフトウェアのインストール 17
ゾーン・クラスタの構成 18
Oracle Databaseの構成 26
Oracle RAC構成 36
Oracle Solaris Volume Manager上のQFSを使用したOracle RAC

10g/11g 37
RAID上のQFSを使用したOracle RAC 10g/11g 39
Oracle Solaris Volume Manager上のOracle RAC 10g/11g 42
ハードウェアRAIDを使用したOracle ASM上のOracle RAC 10g/11g . . 43
Oracle Solaris Volume Managerを使用したOracle ASM上のOracle RAC 10g/11g 45
NAS上のOracle RAC 10g/11g 47
Oracle Solaris Volume Manager上のQFSを使用したOracle9i RAC . . 49
RAID上のQFSを使用したOracle9i RAC 51
Oracle Solaris Volume Manager上のOracle9i RAC 52
NAS上のOracle9i RAC 54
ハードウェアRAID上のOracle9i RAC、Oracle RAC 10g/11g 55
単一システム上の複数構成 55
著者について 57
謝辞 57
参考資料 58

Oracle Solarisゾーン・クラスタへのOracle RACのデプロイ

仮想化テクノロジーは、複数のアプリケーションを単一システムに集約してシステム使用率を上げるための一般的な手段です。Oracle® Solaris Clusterで使用できるOracle Solarisゾーン・クラスタ（別名Oracle Solaris Containersクラスタ）は仮想クラスタを提供しており、複数のクラスタ・アプリケーションの単一クラスタへの集約をサポートしています。本書では、ゾーン・クラスタへのOracle Real Application Clusters（Oracle RAC）のデプロイ方法を具体的に説明します。

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。

- ″ゾーン・クラスタの概要″ではゾーン・クラスタの全体像を紹介します。
- ″Oracle Solaris Clusterを使用するメリット″では、Oracle RACをゾーン・クラスタにデプロイするときに、Oracle Solaris Clusterを使用することでもたらされるメリットを簡単にまとめます。
- ″ゾーン・クラスタでのOracle RAC″では、ゾーン・クラスタがOracle RACと連携する方法を説明します。
- ″例：ゾーン・クラスタによるOracle RACのホスティング″では、ゾーン・クラスタ上のOracle RAC構成例について段階的に説明します。
- ″Oracle RACの構成″では、ゾーン・クラスタでサポートされる各種のOracle RAC構成について詳しく説明します。

本書は、読者がOracle Solaris Clusterとゾーン・クラスタの基礎を理解していることを前提としています。ゾーン・クラスタについて、詳しくはOracleホワイト・ペーパー『ゾーン・クラスタ-仮想クラスタの配備方法と配備する理由』を参照してください。

ゾーン・クラスタの概要

Oracle Solaris 10オペレーティング・システムには、グローバル・クラスタとゾーン・クラスタの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、一連のOracle Solarisホストに含まれるすべての大域ゾーンを含みます。また、任意で、メンバーシップ投票権を持たない非大域ゾーンを含むことができます。ゾーン・クラスタに対する管理操作に必要な権限を持つのは、グローバル・クラスタ内の大域ゾーンのみです。
- **ゾーン・クラスタ。**ゾーン・クラスタは1つまたは複数の非大域ゾーンで構成されます。これらすべての非大域ゾーンのゾーン・ブランド・タイプはクラスタになります。ゾーン・クラスタに含まれるクラスタ・ノードは、それぞれ異なるOracle Solarisホスト上に存在します。

ゾーン・クラスタ・ノードを稼働できるようにするには、同じOracle Solarisホスト上にある大域ゾーンをクラスタ・モードで起動する必要があります。すべてのゾーン・クラスタ・ノードが、同じグローバル・クラスタに属するOracle Solarisホスト上に存在している必要があります。ゾーン・クラスタ・ノードは、この同じグローバル・クラスタに属するOracle Solarisホストの一部である場合もあります。ゾーン・クラスタが1つのグローバル・クラスタに依存する一方で、グローバル・クラスタはゾーン・クラスタに依存しません。

図1に、Oracle Solaris Cluster製品でサポートされる4台のマシンのハードウェア構成を示します。図2では、この同じ4台のマシンのハードウェア構成上で複数のクラスタが実行されています。

重要なのは、クラスタ・ノードがクラスタ・ソフトウェアの構成要素であり、必ずしもハードウェアと1対1関係を持つ必要はないため、単一マシンが複数のクラスタ・ノードをホストできるという点です。

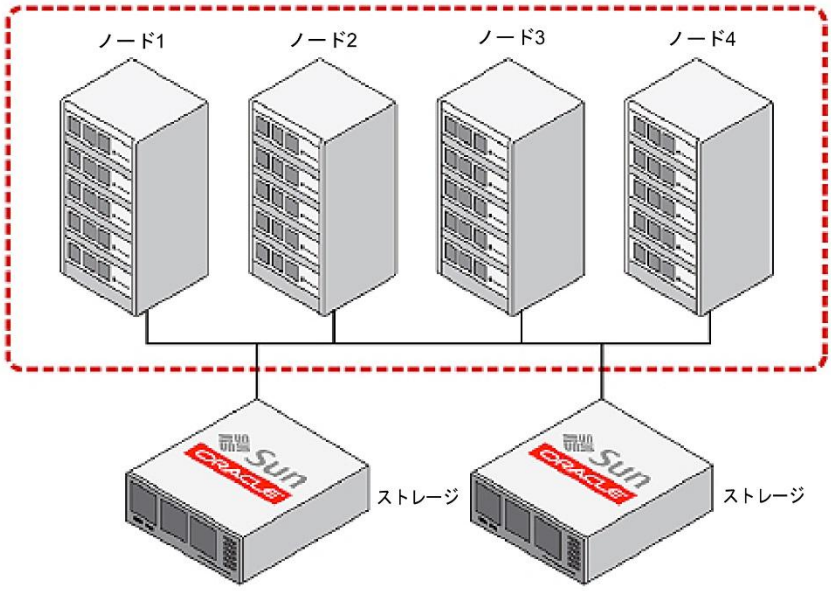

図1. 4台のマシンのハードウェアからなるクラスタ構成

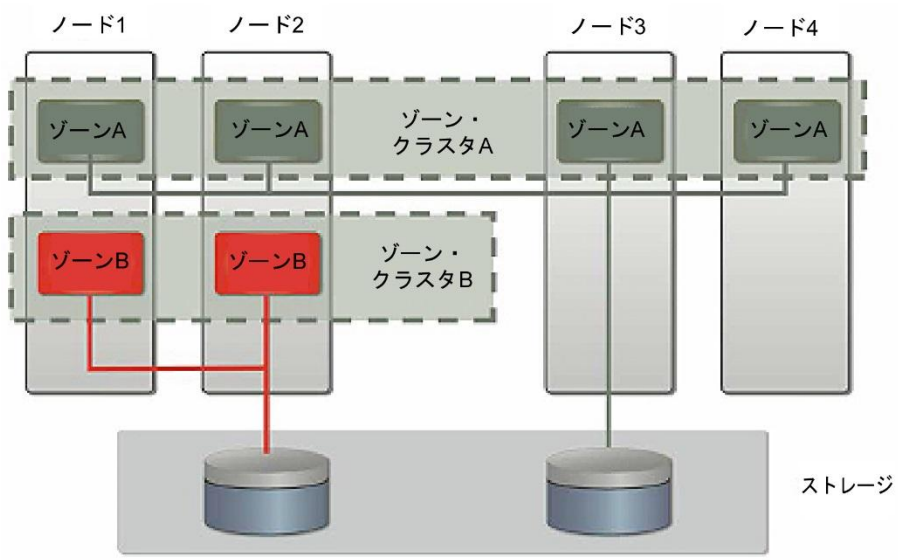

図2. 4マシン構成にデプロイされた複数クラスタ

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・クラスタはクラスタ・アプリケーションに対して、あたかもゾーン・クラスタが従来のクラスタであるかのように見せます。

ゾーン・クラスタは次の機能を提供します。

- **セキュリティの分離。**あるゾーン・クラスタ内のアプリケーションおよびユーザーが、異なるゾーン・クラスタで実行されているアプリケーションやデータを表示またはアクセスすることはできません。
- **アプリケーション障害の分離。**あるゾーン・クラスタ内のアプリケーションによって実行されたアクション（リブート・コマンドの発行など）が、このゾーン・クラスタ外のアプリケーションに影響を与えることはありません。
- **リソース管理。**リソースまたは共有リソースをゾーン・クラスタに割り当てることができます。これには、CPU、メモリ、デバイス、ファイル・システムなどのリソースが含まれます。
- **専用クラスタ・モデル。**ゾーン・クラスタは、ゾーン・クラスタ内のクラスタ・アプリケーションが専用で使用できる従来型クラスタであるかのように見せかけます。アプリケーションが、ハードウェア上にあるその他のクラスタを認識することはありません。従来型クラスタとは、仮想マシンやゾーンが存在する前にマシン上で実行されていたタイプのクラスタとして見なすことができます。

ゾーン・クラスタをデプロイする重要な動機の1つは、複数のクラスタ・アプリケーションを1つの物理クラスタ構成に集約することです。集約により、しばしば、異なる組織に属するアプリケーションが同じハードウェア上で実行されることがあります。たとえば、ゾーン・クラスタを使用すると、金融、販売、エンジニアリングの各組織用のOracle RACデータベースを同じハードウェア上に配置できます。また、ゾーン・クラスタは幅広いアプリケーションの集約をサポートしています。例を挙げると、複数のゾーン・クラスタを使用して、従来型データセンターのフロントエンド層、アプリケーション層、データベース層をすべて同じ一連のハードウェア上に集約できます。

データベースは多くの組織にとってクリティカルなアプリケーションです。信頼性を確保するため、多くの組織は本番、テスト、開発用にハードウェアを切り離しています。ゾーン・クラスタを使用すると、テスト機能と開発機能を1つのハードウェア構成に集約できます。エンジニアたちは全員が同時に作業を実施でき、タイム・スライスも排除できます。管理者はゾーン・クラスタを動的に作成および破棄することができ、その他のゾーン・クラスタに影響を与えることもありません。

ゾーン・クラスタは段階的なデータベースのアップグレードをサポートしています。たとえば、同じハードウェア上の別々のゾーン・クラスタで、Oracle9i RAC、Oracle RAC 10g、Oracle RAC 11gを同時に実行することも可能です。管理者は各自の都合に合わせて、データベースを1つずつ新しいデータ・ソフトウェア・リリースにアップグレードできます。

多くのソフトウェア・ベンダーは、CPUの数に基づいてソフトウェアのライセンス料金を請求します。管理者はゾーン・クラスタ内のCPU数を調整できるため、ソフトウェアで使用できるCPUの数もコントロールできます。一部のベンダーはこのCPU数をライセンス料金のベースとして使用しています。多くの組織はアプリケーションとデータベースを併用しています。データベース・ソフトウェアをその他のアプリケーションとは別のゾーン・クラスタに配置できるため、管理者はそれぞれ異なるソフトウェア・システムに対して個々にCPU数を調節できます。このアプローチにより、大幅なコスト節約が可能になります。

これらのユースケースとそれに伴うコスト節約について、詳しくは『*ゾーン・クラスタ-仮想クラスタの配備方法と配備する理由*』を参照してください。

Oracle Solaris Clusterを使用するメリット

一連のマシン上にOracle Solaris Clusterソフトウェアをデプロイすると、これらのマシンの大域ゾーンにデプロイしたアプリケーションと、ゾーン・クラスタを構成してデプロイする利点があります。このメリットは、デプロイされているOracle RACデータベースと、完全なアプリケーション・スタックを構成するその他のソフトウェア・コンポーネントのどちらにも当てはまります。

含まれる機能は次のとおりです。

- 堅牢なカーネル・ベースのメンバーシップ・メカニズム
- 信頼できるストレージ・フェンシング・メカニズム
- 擬似STREAMネットワーク・デバイス・ドライバ(`clprivnet`)による、単一ネットワーク・アドレス経由での高可用性トランキング接続の提供
- すべてのクラスタ・マシンを通じて一貫したディスク・デバイス名前空間
- フル機能付きの成熟したフレームワークによるアプリケーション・コンポーネント制御
- 幅広いエージェント・ポートフォリオによるアプリケーション・コンポーネントの制御と監視
- アプリケーション・スタック全体を含むディザスタ・リカバリ・ソリューションの構築プラットフォーム

次のセクションから各機能について説明していきます。詳しくは、『*Oracle Solaris Cluster Essentials*』を参照してください。

クラスタのメンバーシップ

Oracle Solaris Clusterのメンバーシップはカーネル・ベースのメンバーシップ・モニターによって決定されます。カーネル・ベースであるため、同様のフレームワークが一連のユーザー・プロセスとして実装された場合に発生する問題をほとんど回避できます。

メンバーシップ・モニターはプライベート・インターコネクトを構成するネットワークを使用して、クラスタ内のマシン間でハートビート・メッセージを送信します。10秒間にわたってすべてのプライベート・インターコネクトがハートビート・メッセージを受信しない場合、クラスタ・マシンの障害が検出されます。このイベントはクラスタの再構成を引き起こすため、残りのクラスタ・マシンに構成されたすべてのゾーン・クラスタに新しいメンバーシップ情報が反映されます。その結果として、デプロイ先がグローバル・クラスタであるかゾーン・クラスタであるかに関係なく、Oracle RACデプロイメントは素早く正確に新しいメンバーシップ情報を認識できます。新しいクラスタに対して参加資格のないマシンは強制的に終了され、クラスタから除外されます。この際、マシン上に存在するすべてのゾーン・クラスタ・ノードも終了されます。

メンバーシップ・モニターは新しいメンバーシップ情報を提供するだけでなく、データを破損から保護するストレージ・フェンシング・プロセスと、スペース（″スプリットブレイン″）と時間（″アムネジア″）の両方のクラスタ分割を防止するクォーラム・メカニズムを起動します。

ストレージ・フェンシング

時として、誤った状態にあり、割当て時間内にピア通信に失敗したマシンがクラスタから除外される場合があります。このような場合は、クラスタ再構成プロセスが開始されたら、そのマシンから共有ストレージに書込みを継続できないようにすることが重要です。Oracle Solaris Clusterが採用しているSCSIフェンシング・メカニズムは、除外された1つまたは複数のノードによる書込みがあった場合、すぐにパニックを発生させます。

Oracle Solaris ClusterはNASユニット、ディスク、RAIDデバイスのフェンシングをサポートしています。データ破損を防止するこの重要なデータ整合性機能は、長年にわたってOracle Solaris Clusterでサポートされており、現在はゾーン・クラスタでもサポートされています。

擬似STREAMネットワーク・ドライバ`clprivnet`

Oracle Solaris Clusterは`clprivnet`ドライバを使用して、プライベート・インターコネクトを介したノード間の通信を改善しています。プライベート・インターコネクトは最大で6つまで使用できます。`clprivnet`ドライバを使用する場合、Oracle Solaris Clusterは使用できるすべての通信パス間でトラフィックをストライピングします。

Oracle Solaris Clusterは通信パス障害を透過的に回復し、少なくとも1つのパスが使用できる限り配信を保証します。`clprivnet`ドライバはOracle9i、Oracle 10g、Oracle 11g Release 1、Oracle 11g Release 2で使用でき、すべてのケースで、Oracleソフトウェアに対して仮想ネットワーク・アダプタとIPアドレスを1つずつ提示します。

また、ゾーン・クラスタ内でこのインタフェースに対して必要なネットワーク・アドレスの構成は、ゾーン・クラスタの作成時に自動的に処理されるため、構成ミスのリスクが解消されます。

一貫性のあるディスク・デバイスの名前空間

複数のマシン・タイプを含むクラスタを作成またはアップグレードすると、しばしば、マシンごとに異なるcXtYdZデバイス・パスが共有ストレージ・デバイスに設定されます。Oracle Solaris Clusterはクラスタ内の全マシンに対して一貫性のある名前空間を作成することで、自動的にこの違いに対処します。これらの新しいデバイス名は、共有QFSまたはOracle Automatic Storage Managementから使用できます。この機能により、デバイスにシンボリック・リンクを作成して管理する必要がなくなります。

アプリケーション・コンポーネントの制御

Oracle Solaris Clusterに統合されているリソース・グループ・マネージャを使用すると、アプリケーション固有エージェントを通じてアプリケーション・コンポーネントの開始、停止、状態監視を制御できます。リソース・グループ・マネージャは、クラスタへのノードの追加や削除といったイベントによって駆動されるか、または管理者がコマンドラインから操作できます。リソース・グループ・マネージャでは、アプリケーション・コンポーネント・グループ間にプラスとマイナスのアフィニティを定義でき、個別コンポーネント間には依存性を定義できます。いずれの関係もゾーン・クラスタの境界をまたがって定義できます。この機能を使うと、特定のゾーン・クラスタ内の特定の層にあるアプリケーション・コンポーネントが、異なるゾーン・クラスタ内の別の層（RACデータベースなど）にあるコンポーネントに依存するように指定できます。このような設定は、Oracle ClusterwareやOracle Grid Infrastructureフレームワークでは実現できません。

幅広いエージェント・ポートフォリオ

Oracle Solaris Clusterには、アプリケーション・コンポーネントの起動や停止、状態監視を実行するエージェントの豊富なポートフォリオが付属しています。Oracle ClusterwareとOracle RACに関係するエージェントの一部については後述します。その他に、Oracle WebLogic、Oracle E-Business Suite、Siebel、Peoplesoft、SAP、Oracle Business Intelligenceなどの幅広いOracleアプリケーション・コンポーネントを制御できる50以上のコンポーネントが提供されています。これらのコンポーネントはすべてゾーン・クラスタに対応しているため、アプリケーション・スタック全体を1つのマシン・クラスタに集約できます。

既存のエージェントがないアプリケーションに対しては、Generic Data Serviceに、起動、停止、状態監視のスクリプトを組み込むことで、アプリケーション・コンポーネントの再起動とフェイルオーバーに関するベスト・プラクティスを含むサービスをすぐに実装することができます。

ディザスタ・リカバリ・ソリューションの構築

Oracle Solaris Clusterをデプロイすると、Oracle Solaris Cluster Geographic Editionソフトウェアを介して複数のクラスタを連携させることができます。この組合せはすべてのアプリケーション層に対して、一般的なデータ・レプリケーション・メカニズムの多くを含む共通ディザスタ・リカバリ・フレームワークを提供します。対象メカニズムには、ストレージ・レプリケーション、ホスト・ベース・レプリケーション、アプリケーション・ベース・レプリケーションが含まれます。Oracle RACに関しては、Geographic Editionフレームワークによって Oracle Data Guardがサポートされています。

Geographic Editionフレームワークの個々のインスタンスは、物理的なクラスタ上に構成された任意のグローバル・クラスタおよびゾーン・クラスタ上で実行できます。

ゾーン・クラスタ内のOracle RAC

Oracle RACの観点から言うと、ゾーン・クラスタはあたかも専用クラスタであるかのように見えます。言い換えると、Oracle RACから見てゾーン・クラスタ内での実行とゾーンなしのクラスタ上での実行に違いはありません。一般に、グローバル・クラスタ内で稼働するOracle RACソフトウェアは、変更なしでゾーン・クラスタ内でも実行できます。ただし、ゾーン・クラスタにOracle RACソフトウェアをインストールする前に最新のサポート・マトリックスをチェックして、必要なSolarisパッチとOracleパッチを適用する必要があります。

ゾーン・クラスタは以下のOracleリリースに対応しています。

- Oracle9i RAC（9.2.0.5以上）
- Oracle RAC 10g（10.2.0.3以上）
- Oracle RAC 11g Release 1（11.1.0.6以上）
- Oracle RAC 11g Release 2（11.2.0.2以上）

ゾーン・クラスタは独立した環境であるため、リリースごとに異なるゾーン・クラスタを使用することで、同じOracle Solaris Cluster上で任意の組合せのOracle RACバージョン（9i/10g/11g）を実行できます。

大域ゾーンで実行されるOracle RACは複数のデータベースをサポートします。同様に、ゾーン・クラスタで実行されるOracle RACも、1つのゾーン・クラスタ内での複数データベースの実行をサポートします。

ここからは、ゾーン・クラスタがどのようにOracle RACと連携するのかについて詳しく説明していきます。

Oracle Solaris ClusterとOracle RACの統合

Oracle Solaris Clusterは単にOracle RAC向けのプラットフォームを提供するにとどまらず、クラスタにOracle RACを統合します。さらに、Oracle RACを管理するだけでなく、Oracle RACを使用して動作するアプリケーションも管理します。Oracle RACを含む一連のアプリケーション全体を、Oracle Solaris Clusterを介して管理できます。また、Oracle Solaris Clusterは、デバイスやファイル・システムなどの重要リソースに関する情報をOracle Clusterware[1]（10g Release 2）またはGrid Infrastructure（11g Release 1および11g Release 2）に提供します。この統合により、Oracle Solaris Clusterはデータベース・インストールに関するリソースとアプリケーションの全体を、よりインテリジェントに管理できるようになるため、総合的な可用性が向上します。ここからは、Clusterwareという用語には、特に指定しない限りGrid Infrastructureが含まれるものとします。

図3に、クラスタ・ファイル・システムまたはRAWデバイスにOracle RACデータベースが格納されている場合の、ClusterwareとOracle Solaris Clusterの両方から見たリソースの管理を示します。図4は、11g Release 2データベースがOracle Automatic Storage Management（Oracle ASM）上に格納されている場合の同じ構成図です。10g Release 2と11g Release 1の構成では、Oracle ASM用のClusterwareリソースはありません。その代わりに、Oracle Solaris ClusterがOracle ASM用のリソースを提供し、プロキシASMリソースをClusterwareの制御下にインストールします。

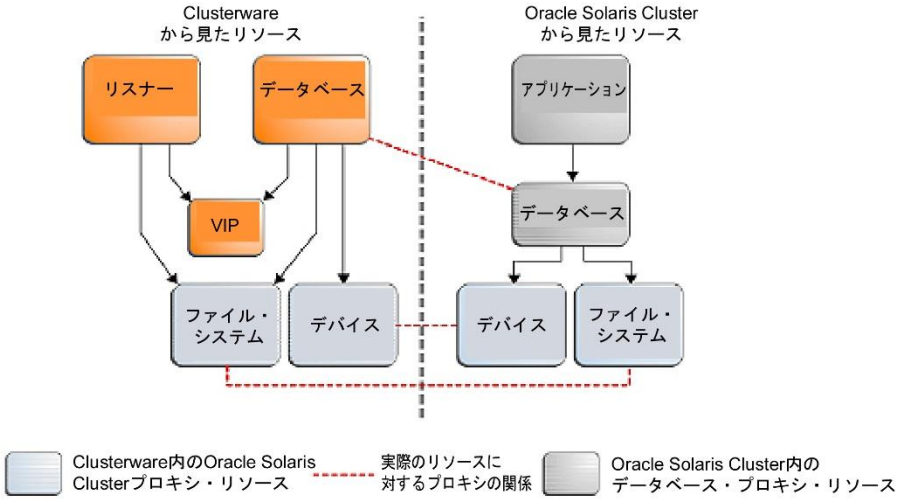

図3. データベースがクラスタ・ファイル・システムまたはRAWデバイス上にある場合のOracle Solaris ClusterおよびOracle Clusterwareから見たリソース図

[1] Oracle Clusterwareは当初、Cluster Ready Services（CRS）と呼ばれていました。

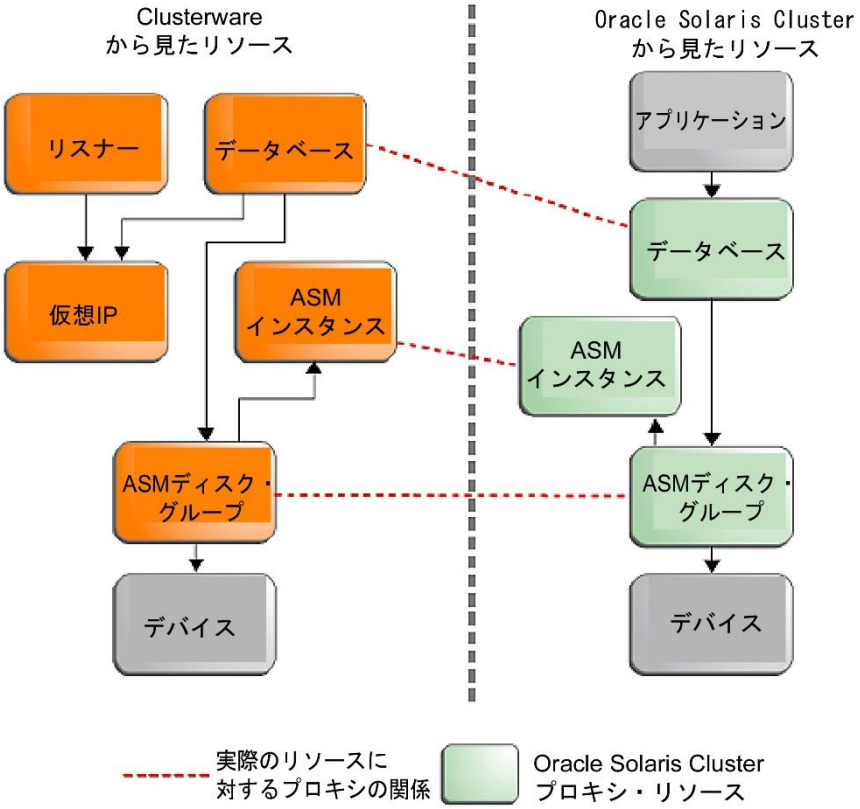

図4. データベースがOracle ASM上にある場合のOracle Solaris ClusterおよびOracle Grid Infrastructure (11g Release 2) から見たリソース図

Oracle RACリソースの分離

Oracle RACは以下のリソースを使用します。

- パブリック・ネットワークにアクセスするためのネットワーク・リソース（IPアドレス）
- アプリケーションと通信するためのネットワーク・リソース（IPアドレス）
- その他のデータベース・インスタンスと通信するためのネットワーク・リソース（IPアドレス）
- データベース情報を格納するためのクラスタ・ファイル・システムまたは完全接続されたストレージ・デバイス

図5に、同じ物理クラスタ上の2つの異なるOracle RACインストールに対して、2つのゾーン・クラスタでリソースを分離した例を示します。この例では、ゾーン・クラスタZC1がOracle Solaris Cluster向けのOracle Solaris Volume Manager上でQFS共有ファイル・システムを使用するOracle RACをサポートしており、ゾーン・クラスタZC2がAutomatic Storage Managementを使用するOracle RACをサポートしています。

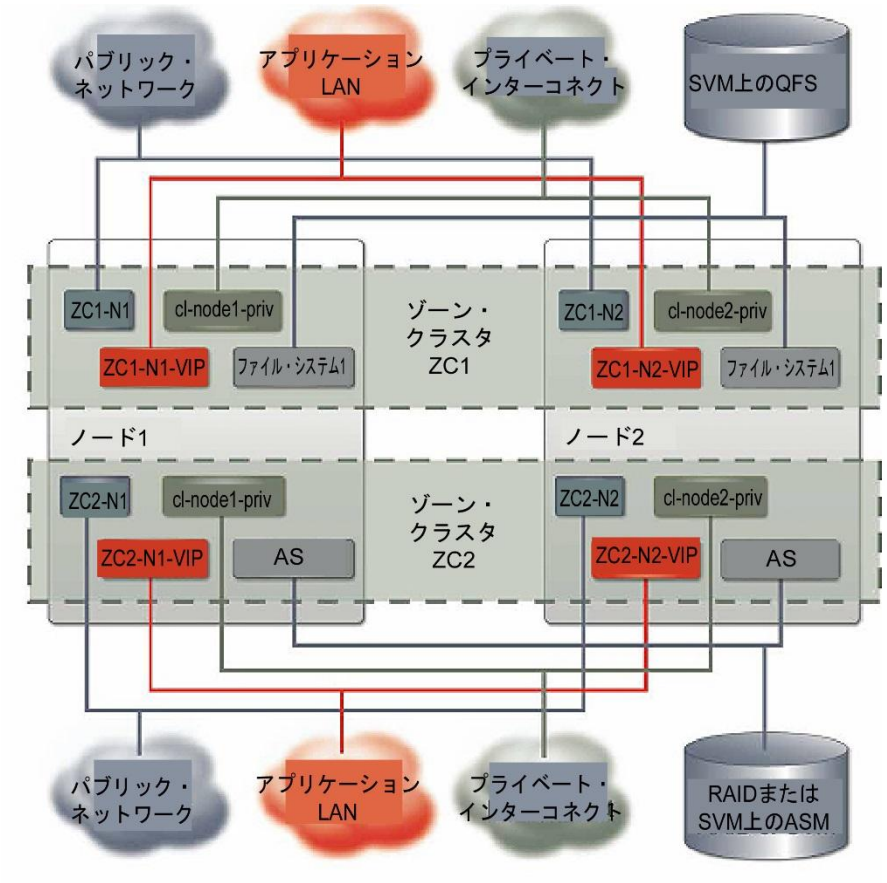

図5. ゾーン・クラスタ内でのOracle RACリソースの分離

共通リソース・コンポーネント

Oracle RACをサポートするすべての構成で多数のリソース・コンポーネントが使用されています。説明の重複を減らすため、このセクションではこれらの共通コンポーネントについて説明します。特定の構成でのみ使用されるリソース・コンポーネントについては、それぞれの構成セクションで説明します。

Oracle RACフレームワーク

スケーラブルなOracle RACフレームワーク・リソース（`SUNW.rac_framework`）は、Oracle RACをサポートするためのサービスを提供しています。Oracle RACフレームワーク・リソースはメンバーシップなどのプラットフォーム情報をOracle RACに提供し、Oracle RAC関連のソフトウェア再構成をとりまとめます。Oracle RACフレームワーク・リソースを再起動する場合は、リソースの構成場所に応じてグローバル・クラスタまたはゾーン・クラスタを再起動する必要があります。

VUCMMフレームワーク

VUCMMフレームワーク・リソース（SUNW.vucmm_framework）は、1つのOracle Solaris Clusterに異なるクラスタ・ボリューム・マネージャを構成できるようにするフレームワークを表しています。このリソースによってフレームワーク・ステータスの監視が可能になります。VUCMMフレームワーク・リソースを再起動する場合は、リソースの構成場所に応じてグローバル・クラスタまたはゾーン・クラスタを再起動する必要があります。

CRSフレームワーク

スケーラブルなCRSフレームワーク・リソース（SUNW.crs_framework）はOracle Clusterwareを監視し、残りのクラスタに対してClusterwareの状態を提示します。このClusterwareフレームワーク・リソースはClusterwareの自動開始プロパティを無効化して（もう一度有効化された場合、再度無効化はしません）、Clusterwareに開始と停止の両方を指示できるようにします。ただし、このClusterwareリソースは障害発生後にClusterwareを再起動しません。これは、Clusterwareには固有の再起動機能が組み込まれているためです。このClusterwareリソースはOracle RAC 10g/11gをサポートしています。Oracle RACがゾーン・クラスタ内に構成されている場合、CRSフレームワーク・リソースはゾーン・クラスタ内に存在します。

Oracle UNIX Distributed Lock Manager（Oracle UDLM）

スケーラブルなOracle UDLMリソース（SUNW.rac_udlm）は、Oracle RACに対応した分散ロック・マネージャ・サービスを提供するOracle UDLMを管理します。ゾーン・クラスタ内のOracle RACをサポートする場合、Oracle UDLMリソースはゾーン・クラスタ内に構成されます。Oracle UDLMはSPARC®プロセッサ・ベースのマシンでのみ稼働します。

Oracle RACインスタンス・プロキシ・リソース

スケーラブルなOracle RACインスタンス・プロキシ・リソース（SUNW.scalable_rac_server_proxy）に設定されたOracle Clusterwareからのコールバックは、データベースの状態に関する情報を提供します。Oracle RACインスタンス・プロキシ・リソースはClusterwareからデータベースの状態に関する情報を受け取ります。データベースはファイル・システムまたはストレージ・デバイスのいずれかに配置されています。このリソースはOracle Clusterwareを介してOracleデータベースの起動と停止を実行できます。このリソースはOracle RAC 10g/11gで使用できます。Oracle RACがゾーン・クラスタ内に構成されている場合、Oracle RACインスタンス・プロキシ・リソースはゾーン・クラスタ内に存在します。

Oracle RACサーバー・リソース

スケーラブルなOracle RACサーバー・リソース（SUNW.scalable_rac_server）はOracleデータベース・サーバーを監視します。Oracle RACサーバー・リソースはOracleデータベースの起動と停止を実行できます。このリソースを使用できるのはOracle9i RACのみです。Oracle RACがゾーン・クラスタ内に構成されている場合、Oracle RACサーバー・リソースはゾーン・クラスタ内に存在します。

Oracle RACリスナー・リソース

スケーラブルなOracle RACリスナー・リソース（SUNW.scalable_rac_listener）はOracleリスナー・プロセスを開始または停止します。このリソースを使用できるのはOracle9i RACのみです。Oracle RACがゾーン・クラスタ内に構成されている場合、Oracle RACリスナー・リソースはゾーン・クラスタ内に存在します。

Oracle ASMインスタンス・プロキシ・リソース

スケーラブルなOracle ASMインスタンス・プロキシ・リソース（SUNW.scalable_oracle_asm_instance_proxy）はプロキシ・リソースであり、Oracle RAC 10gおよび11g構成でのみ使用できます。このプロキシ・リソースはOracle Clusterwareが管理するリソースの代理となることで、Oracle Solaris Clusterユーティリティから Oracle Clusterwareを介してOracle RACサーバー・インスタンスを管理できるようにします。こうすることで、SUNW.scalable_asm_instance_proxyリソース・タイプは、Oracle Solaris ClusterとOracle Clusterwareが提供するそれぞれのクラスタ化フレームワークが相互運用できるようにします。Oracle ASMインスタンス・プロキシ・リソースはOracle Clusterwareを介してOracle ASMデータベースを起動または停止できます。また、モニタを使用して、SUNW.scalable_asm_instance_proxyリソースがプロキシの役割を果たすClusterwareリソースからステータス情報を受け取ります。Oracle RACがゾーン・クラスタ内に構成されている場合、Oracle ASMインスタンス・プロキシ・リソースはゾーン・クラスタ内に存在します。

Oracle ASMディスク・グループ・リソース

Oracle ASMディスク・グループ・リソース（SUNW.oracle_asm_diskgroup）を使用できるのはOracle 10gおよび11g Release 1のみです。このリソースは、Oracle Solaris Cluster構成に含まれる単一インスタンスまたはクラスタ・インスタンスのOracle ASMディスク・グループを表します。それぞれのSUNW.scalable_asm_diskgroupリソースが、クラスタ化されたOracle ASMディスク・グループを表します。クラスタ化されたOracle ASMディスク・グループは、インスタンスの実行ノードに対するasm_diskgroups拡張プロパティの値によって一意に識別されます。Oracle RACがゾーン・クラスタ内に構成されている場合、ASMディスク・グループ・リソースはゾーン・クラスタ内に存在します。

Oracle ASMディスク・グループ・プロキシ・リソース

スケーラブルなASMディスク・グループ・プロキシ・リソース（SUNW.scalable_asm_diskgroup_proxy）を使用できるのはOracle 11g Release 2のみです。このリソースは、Oracle Solaris Cluster構成に含まれる単一インスタンスまたはクラスタ・インスタンスのOracle ASMディスク・グループを表します。それぞれのSUNW.scalable_asm_diskgroup_proxyリソースが、クラスタ化されたOracle ASMディスク・グループを表します。クラスタ化されたOracle ASMディスク・グループは、インスタンスの実行ノードに対するasm_diskgroups拡張プロパティの値によって一意に識別されます。Oracle RACがゾーン・クラスタ内に構成されている場合、Oracle ASMディスク・グループ・プロキシ・リソースはゾーン・クラスタ内に存在します。

ゾーン・クラスタ起動待機リソース

このスケーラブル・リソース（SUNW.wait_zc_boot）は、マシン上で指定されたゾーン・クラスタ・ゾーンの起動が完了するのを待機してから、同じマシン上で自身の起動メソッドを完了します。このリソースを使用すると、ゾーン・クラスタのゾーンが起動してから特定のリソースがオンライン状態になるようにできます。特に、このゾーン用のQFS共有ファイル・システムがマウントされる前に、ゾーン・クラスタ・ゾーンが確実に起動しているようにするために使用されます。

以下の例では、SUNW.wait_zc_bootリソースを使用することで、ゾーン・クラスタ・ゾーンの起動後にQFS共有ファイル・システムがマウントされるように保証しています。これらのリソースとリソース・グループはすべて、大域ゾーン内に存在しています。

次のコマンドは、リソース・グループが大域ゾーン内にあらかじめ作成されており、ゾーン・クラスタの名前はzc-racであることを前提としています。

```
# clresourcetype register SUNW.wait_zc_boot
# clresource create -t SUNW.wait_zc_boot \
-g scal-wait-zc-rg \
-p ZCName=zc-rac \ wait-zc-rs
# clresource create -t SUNW.qfs \
-g mds-rg \
-p resource_dependencies=wait-zc-rs \
-p QFSFileSystem=/zone/zc-rac/root/db_qfs/Data \ mds-rs
#
```

・最初のコマンドのclresourcetype registerによって、SUNW.wait_zc_bootリソース・タイプが登録されます。この手順は、このリソース・タイプを使用するすべてのゾーン・クラスタに対して1回だけ実行します。

- 次のコマンドは、スケーラブル・リソース・グループであるscal-wait-zc-rg内に、wait-zc-rsという名前のSUNW.wait_zc_bootリソースを作成します。リソース拡張プロパティZCNameには、SUNW.wait_zc_bootリソースが待機するゾーン・クラスタの名前を指定します（この例では、zc-rac）。
- 最後のコマンドは、mds-rgフェイルオーバー・リソース・グループ内にmds-rsという名前のQFSメタデータ・サーバー（SUNW.qfs）リソースを作成します。この例では、ゾーン・ルート・パスが/zones/zc-racであり、リソースwait-zc-rsに対するリソース依存性があり、QFS共有ファイル・システムが/zones/zc_rac/root/db_qfs/Dataにマウントされていることを前提としています。

Sun QFS共有ファイル・システム

QFS共有ファイル・システムはデータの読取りと書込みを各ノードから直接実行します。各ノード上のQFS共有ファイル・システムのステータスは、スケーラブルなマウント・ポイント・リソース（SUNW.ScalMountPoint）によって監視されます。このリソースはゾーン・クラスタ内に存在します。

QFS共有ファイル・システムはクラスタ全体に対して、ファイル・システムごとに1つのアクティブ・メタデータ・サーバー（MDS）を持っており、このMDSが特定のCluster File Systemのメタ情報を管理します。このMDSはどのノードでもホストできます。MDSは同時に1ノードでのみアクティブになるフェイルオーバー・リソースです。ノード障害が発生した場合、システムは自動的に稼働中のノードでMDSをアクティブ化します。MDSリソースはOracle RAC構成で使用される唯一のフェイルオーバー・リソースです。QFSメタデータ・サーバー・リソース（SUNW.qfs）はQFSメタデータ・サーバーを管理し、大域ゾーン内にのみ存在します。

このゾーン・モデルは、基盤となるストレージ・デバイスに対するアクセス権を直接与えることなく、ゾー

ン内にファイル・システムを配置するように促進するものです。悪意のあるプログラムが悪質なioctl関数を呼び出すと、マシン全体が停止する場合があります。ファイル・システムを介してデータ・ストレージにアクセスする場合、この処理は不可能であるため、システムの総合的な信頼性が向上します。基盤となるデバイスをゾーン内に配置しないことで追加のオーバーヘッドが発生することはありません。

QFS共有ファイル・システムは、大域ゾーンのみで稼働するOracle Solaris Volume ManagerまたはハードウェアRAID上で実行されます。Oracle Solaris Volume Managerリソース（SUNW.vucmm_svm）はOracle Solaris Volume Managerを制御します。Oracle Solaris Volume Managerディスク・セットに対するすべての管理操作は大域ゾーン内で実行されます。

Oracle UDLMおよびネイティブSKGXNのサポート

SKGXN（システム・カーネル汎用インタフェース・ノード）は、Oracle RACにプラットフォーム情報を提供するためのOracle RACインタフェースです。SPARCプロセッサ・ベース・マシン上の任意のゾーン・クラスタは、Oracle UDLMまたはネイティブSKGXNサポートを使用できます。このため、1つの物理クラスタで両方のオプションをサポートします。ただし、各ゾーン・クラスタで使用できるのはOracle UDLMまたはネイティブSKGXNサポートのいずれか一方です。

Oracle RAC用の構成ウィザード

Oracle Solaris Cluster ManagerにはゾーンでOracle RACを構成するためのウィザードが含まれています。このデータ・サービス構成ウィザードは、本書で説明するさまざまなストレージ構成と各種のOracle RACバージョンをサポートしています。Oracle RAC構成には多数のリソース、リソース・グループ、リソース依存性、リソース・グループ・アフィニティが含まれています。Oracle RACデータ・サービス構成ウィザードはこれらすべての項目の正しい構成方法を認識しているため、管理者の負担を軽減できます。Oracle RACデータ・サービス構成ウィザードは、可能な場合は自動的に検出を実行します。また、Oracle RACデータ・サービス構成ウィザードは、各構成の必須プロパティを設定します。この構成ウィザードは管理者の時間と負荷を節約し、Oracle RAC構成の設定時のエラーを削減します。

Oracle RAC用の構成ウィザードには2つの形式があります。Oracle Solaris Cluster ManagerはOracle RACの構成手段としてブラウザ・ベースのユーザー・インタフェースを提供し、clsetupコマンドはテキスト・ベースのユーザー・インタフェースを提供します。両方のウィザードで実行される処理は同じです。

図6に、ブラウザ・ベースのウィザードを使用してゾーン・クラスタ内にOracle RACを構成する手順の1つのスクリーンショットを示します。この例では、管理者がOracle RAC用のファイル・システム・マウント・ポイントを選択しています。

図6. Oracle RAC用構成ウィザード

例：ゾーン・クラスタによるOracle RACのホスティング

このセクションでは、ゾーン・クラスタ内で複数のOracle RAC構成をホストする単一クラスタの例を紹介します。はじめにクラスタ構成について説明したのち、ゾーン・クラスタのセットアップ方法を詳しく説明します。読者はOracle Solaris、Oracle Solaris Cluster、Oracle Solaris Volume Manager、Sun QFS共有ファイル・システム、Oracle RACの管理操作についての知識があるものとします。以下の説明では、ゾーン・クラスタ向けの新しい操作に焦点を合わせています。

ハードウェア構成の例

このクラスタ・ハードウェアは6つのSun Fire T2000サーバーで構成されており、各サーバーに1GHz 6コア/24スレッドのUltraSPARC®T1プロセッサと8GBの物理メモリが搭載されています。サーバーに構成された`e1000g1`および`e1000g5`のNICがプライベート・ネットワーク・インターコネクト・インタフェースとして機能し、ジャンボ・フレームが有効化されています。また、各サーバーが`e1000g0`および`e1000g2`のNICをパブリック・ネットワーク・インタフェースとして利用しており、これらの同じ2つのNICが単一IPマルチパス（IPMP）グループ内に構成されています。合計で12のLUNを持つ2台のSun StorageTek 3510ストレージ・アレイがディスク・ストレージに使用されています。

クラスタ・ハードウェア以外に、1つのSun Fire T1000サーバーがデータベース上でSwingbench Order Entryワークロードを実行しています。

この構成は一例にすぎず、同様のデプロイでより新しいサーバーを使用することもできます。

個々のゾーン・クラスタ構成

図7に示すとおり、この例ではクラスタ・ハードウェア上で3つのゾーン・クラスタが同時に実行されています。

- zc1-6n – 10.2.0.3のOracle RACデータベースをホストする6ノードのゾーン・クラスタ
- zc2-4n – 10.2.0.4のOracle RACデータベースをホストする4ノードのゾーン・クラスタ
- zc3-2n – 11g Release 1のOracle RACデータベースをホストする2ノードのゾーン・クラスタ

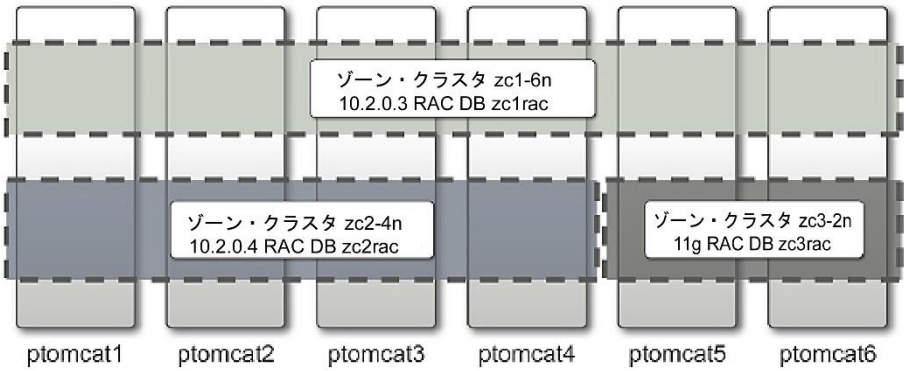

図7. デモ構成の概要

この例では、各ゾーン・クラスタ・ノードに対して、マシンごとに8つの専用CPUと4GBのメイン・メモリが構成されています。今回はすべてのゾーン・クラスタでネットワーク構成オプションip-type=sharedを使用しているため、ゾーン・クラスタはNICを大域ゾーンと共有しており、それぞれのゾーン・クラスタが固有のIPアドレスを使用しています。ゾーン・クラスタはパブリック・ネットワーク通信用に同じIPMPグループを共有していますが、NICには異なる論理インタフェースが接続されています。

同様に、ゾーン・クラスタは同じクラスタ・プライベート・インターコネクトを共有していますが、`clprivnet` NICには異なる論理インタフェースが接続されています。

この例では、各ゾーン・クラスタに4つのLUNが付与されています。各ゾーン・クラスタは2つのSun QFS共有ファイル・システムに対して2つの異なるLUNを使用して、Oracleの実行可能ファイルを格納しています。1つのLUNにはClusterware実行可能ファイルを、もう1つのLUNにはOracle RACデータベースの実行可能ファイルを格納しています。ゾーン・クラスタzc1-6nは、クラスタ機能を持つOracle Solaris Volume Manager上のQFS共有ファイル・システム用に2つの追加LUNを使用して、それぞれにClusterwareデータとデータベース・データを格納しています。

ゾーン・クラスタzc2-4nとzc3-2nは、Oracle Solaris Clusterボリュームを管理するOracle Solaris Volume Managerに対して追加のLUNを使用して、それぞれにClusterwareデータとデータベース・データを格納しています。

図8にゾーン・クラスタzc1-6nの構成を示します。この図には6ノードのうちの2ノードのみ示していますが、残りの4ノードも同様の構成になります。

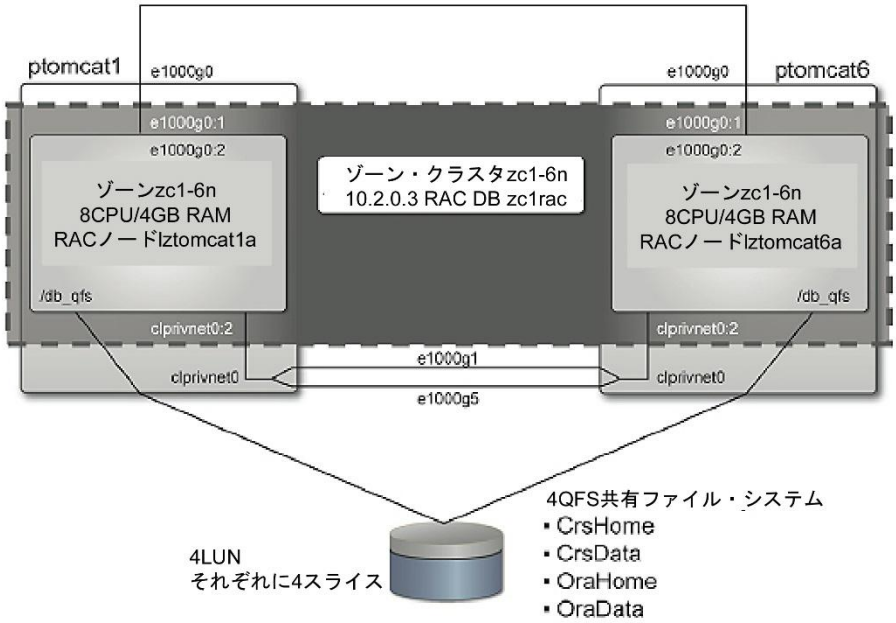

図8. ゾーン・クラスタzc1-6nの構成

図9にゾーン・クラスタzc2-4nの構成を示します。この図には4ノードのうちの2ノードのみ示していますが、残りの2ノードも同様の構成になります。

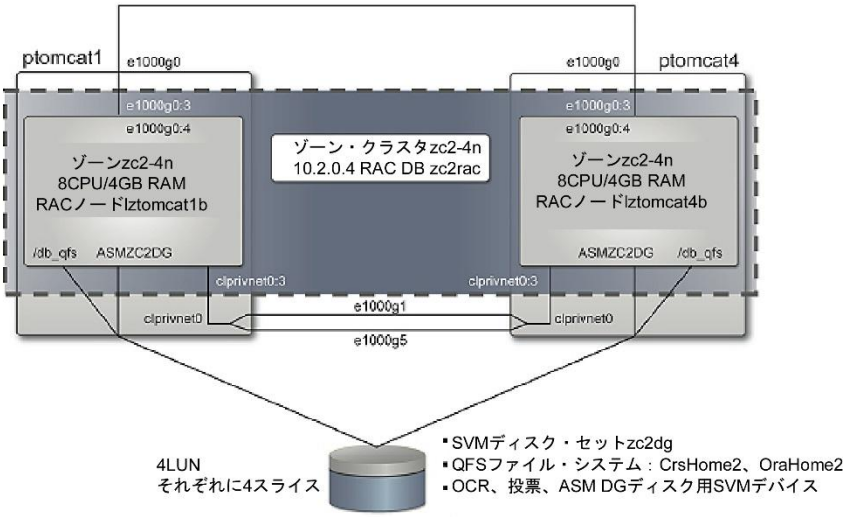

図9. ゾーン・クラスタzc2-4nの構成

図10にゾーン・クラスタzc3-2nの構成を示します。

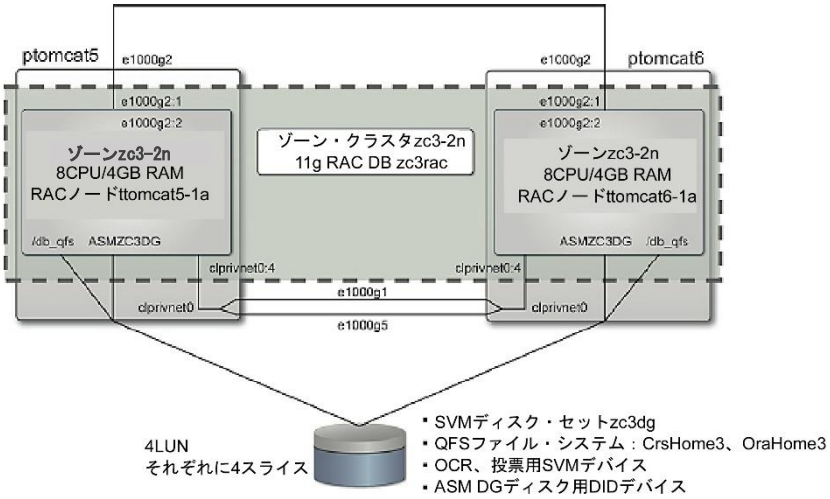

図10. ゾーン・クラスタzc3-2nの構成

この構成例では、バージョンの異なる3つのOracle RACソフトウェアが異なるゾーン・クラスタで独立して実行されています。また、それぞれのゾーン・クラスタに異なる数のノードが含まれます。さらに、Oracle RACデータベースをサポートするために2種類のストレージ・トポロジが使用されています。

Oracle SolarisおよびOracle Solaris Clusterソフトウェアのインストール

それぞれのマシンには、Oracle Solarisソフトウェアと Oracle Solaris Clusterソフトウェアが厳密に1つずつインストールされています。このソフトウェアはマシン上のグローバル・クラスタと全ゾーン・クラスタをサポートします。後からパッチまたは更新を適用する場合、Oracle SolarisとOracle Solaris Clusterに対してそれぞれ別々に、マシンごとにパッチまたは更新を1つだけ適用します。その他にゾーン・クラスタに対して必要なパッチまたは更新作業はありません。

以下の手順に従って、Oracle SolarisおよびOracle Solaris Clusterソフトウェアをインストールします。

1. はじめに、すべてのマシンにOracle Solaris 10 10/09以降をインストールします。最新のパッチをインストールすることを推奨します。
2. 次に、Oracle Solaris Cluster 3.3以降をインストールします。こちらも最新のパッチをインストールすることを推奨します。
3. Oracle Solaris Clusterソフトウェアのインストール中に、管理者はグローバル・クラスタを構成します。構成中に、パブリック・ネットワーク用にIPMPグループが作成され、クラスタのプライベート・インターコネクト用に`clprivnet`インタフェースが作成されます。この手順はゾーン・クラスタの数に関係なく、1回だけ実行されます。

以上で、個別のゾーン・クラスタを作成する準備が整いました。

ゾーン・クラスタの構成

このセクションでは、4ノードのゾーン・クラスタ（zc2-4n）の構成方法について詳しく説明します。

1. はじめに、このゾーン・クラスタで使用するrootパスワードの暗号化された値を特定します（詳しくは`sysidcfg`のマニュアル・ページを参照してください）。
2. ゾーン・クラスタ内にリソースを構成する前に、ストレージ・ボリュームまたはファイル・システムが存在している必要があります。Oracle Solaris Volume Managerボリュームを作成します。これらのボリュームは、グローバル・クラスタのサポートに使用される場合とまったく同じ方法で大域ゾーン内に作成されます。この例で作成されるディスク・セットの名前は`zc2dg`であり、作成されるメタデバイスは以下のとおりです。

```
# metaset -s zc2dg -M -a -h ptomcat1 ptomcat2 ptomcat3 ptomcat4
# metaset -s zc2dg -a /dev/did/dsk/d2[0369]
# metainit -s zc2dg d11 1 1 /dev/did/dsk/d20s0
# metainit -s zc2dg d12 1 1 /dev/did/dsk/d29s0
# metainit -s zc2dg d21 1 1 /dev/did/dsk/d26s0
# metainit -s zc2dg d22 1 1 /dev/did/dsk/d23s0
# metainit -s zc2dg d31 1 1 /dev/did/dsk/d20s1
# metainit -s zc2dg d32 1 1 /dev/did/dsk/d29s1
# metainit -s zc2dg d41 1 1 /dev/did/dsk/d26s1
```

```
# metainit -s zc2dg d42 1 1 /dev/did/dsk/d23s1
# metainit -s zc2dg d51 1 1 /dev/did/dsk/d20s3
# metainit -s zc2dg d52 1 1 /dev/did/dsk/d29s3
# metainit -s zc2dg d61 1 1 /dev/did/dsk/d26s3
# metainit -s zc2dg d62 1 1 /dev/did/dsk/d23s3
# metainit -s zc2dg d71 1 1 /dev/did/dsk/d20s4
# metainit -s zc2dg d72 1 1 /dev/did/dsk/d29s4
# metainit -s zc2dg d73 1 1 /dev/did/dsk/d26s4
# metainit -s zc2dg d74 1 1 /dev/did/dsk/d23s4
# metainit -s zc2dg d1 -m d11 d12
# metainit -s zc2dg d2 -m d21 d22
# metainit -s zc2dg d3 -m d31 d32
# metainit -s zc2dg d4 -m d41 d42
# metainit -s zc2dg d50 -m d51 d52
# metainit -s zc2dg d5 -p d50 120m
# metainit -s zc2dg d60 -m d61 d62
# metainit -s zc2dg d6 -p d60 120m
# metainit -s zc2dg d7 -p d50 1m
```

3. 次に、QFS共有ファイル・システムのCrsHome2（Clusterware用）とOraHome2（Oracleホーム用）をOracle Solaris Volume Managerボリューム上に作成します。これらのファイル・システムは、グローバル・クラスタのサポートに使用される場合とまったく同じ方法で作成されます。これらのファイル・システムのマウント・ポイントはゾーンのルート内になります。
   a. ptomcat1からptomcat4までの各マシンの大域ゾーン内で、対応するファイルに以下のコンテンツを作成または追加します。

```
# more /etc/vfstab
#device     mount   FS    fsck   mount
#to mount to fsck  point   type  pass   at boot  options
...
CrsHome2 - /zones/zc2-4n/root/db_qfs/CrsHome samfs - no shared
OraHome2 - /zones/zc2-4n/root/db_qfs/OraHome samfs - no shared
```

```
# more /etc/opt/SUNWsamfs/mcf
...
#
# File system CrsHome2
#
CrsHome2 5 ms CrsHome2 on shared
/dev/md/zc2dg/dsk/d1 50 md CrsHome2 on
/dev/md/zc2dg/dsk/d2 51 md CrsHome2 on
#
# File system OraHome2
#
OraHome2 6 ms OraHome2 on shared
/dev/md/zc2dg/dsk/d3 60 md OraHome2 on
/dev/md/zc2dg/dsk/d4 61 md OraHome2 on
```

```
# more /etc/opt/SUNWsamfs/samfs.cmd
...
fs=CrsHome2
sync_meta=1
fs=OraHome2
sync_meta=1
```

```
# more /etc/opt/SUNWsamfs/hosts.OraHome2
ptomcat1 clusternode1-priv 1 0 server
ptomcat2 clusternode2-priv 1 0
ptomcat3 clusternode3-priv 1 0
ptomcat4 clusternode4-priv 1 0
```

```
# more /etc/opt/SUNWsamfs/hosts.CrsHome2
ptomcat1 clusternode1-priv 1 0 server
ptomcat2 clusternode2-priv 1 0
ptomcat3 clusternode3-priv 1 0
ptomcat4 clusternode4-priv 1 0
```

   b. それぞれのマシンで、2つのQFS共有ファイル・システムに対してマウント・ポイントを作成します。

```
# mkdir -p /zones/zc2-4n/root/db_qfs/CrsHome
# mkdir -p /zones/zc2-4n/root/db_qfs/OraHome
```

c. これら2つのQFS共有ファイル・システムは複数のクラスタ・ノードにまたがります。このため、次のコマンドは1つのマシン上でのみ実行します（この例ではptomcat1）。

```
# /opt/SUNWsamfs/sbin/sammkfs -S CrsHome2
# /opt/SUNWsamfs/sbin/sammkfs -S OraHome2
```

注 – Oracle Solarisでは、ゾーンの起動時に、ファイル・システムのルート内にマウントされているファイル・システムがあることは許されません。これらのファイル・システムの処理が完了したら、次に進む前にすべてがアンマウントされていることを確認します。

1. 次の`clzonecluster`コマンドを使用してゾーン・クラスタzc2-4nを作成します。

```
# clzonecluster configure zc2-4n
clzc:zc2-4n> create
clzc:zc2-4n> set zonepath=/zones/zc2-4n
clzc:zc2-4n> set autoboot=true
```

上記の1行目では、新しく作成したゾーン・クラスタにzc2-4nという名前を割り当てています。残りのコマンドはゾーンのルートの場所を指定し、マシンの起動時にゾーン・クラスタ・ゾーンを自動的に起動するようにソフトウェアに指示しています。この構成情報をコマンド・ファイルに格納して、clzoneclusterコマンドに渡すこともできます。

2. 次に、`add sysid`サブコマンドを使用して`sysidcfg`情報を指定します。

```
clzc:zc2-4n> add sysid
clzc:zc2-4n:sysid> set root_password=ZiitH.NOLOrRg
clzc:zc2-4n:sysid> set name_service="NIS{domain_name=example.com
  name_server=ns8-137(123.4.5.6)}"
clzc:zc2-4n:sysid> set nfs4_domain=dynamic
clzc:zc2-4n:sysid> set security_policy=NONE
clzc:zc2-4n:sysid> set system_locale=C
clzc:zc2-4n:sysid> set terminal=xterm
clzc:zc2-4n:sysid> set timezone=US/Pacific
clzc:zc2-4n:sysid> end
```

rootパスワードは暗号化されないため、暗号化された状態で入力する必要があります。指定する必要があるフィールドはrootパスワードのみです。その他のフィールドに対して値が指定されていない場合、対応する大域ゾーンの値に基づいて、clzoneclusterコマンドが値を提供します。この例では複数の値を設定していますが、ほとんどのケースでデフォルト値が使用されます。clzoneclusterコマンドによってゾーン・クラスタ内の各ゾーンに対してsysidcfg情報が設定されるため、管理者がマシンごとに同じ処理を繰り返す必要はありません。

3. 次に、add nodeサブコマンドを使用してゾーン・クラスタのノードを特定します。

```
clzc:zc2-4n> add node
clzc:zc2-4n:node> set physical-host=ptomcat1
clzc:zc2-4n:node> set hostname=lztomcat1b
clzc:zc2-4n:node> add net
clzc:zc2-4n:node:net> set address=lztomcat1b
clzc:zc2-4n:node:net> set physical=e1000g0
clzc:zc2-4n:node:net> end
clzc:zc2-4n:node> end clzc:zc2-4n> add node
clzc:zc2-4n:node> set physical-host=ptomcat2
clzc:zc2-4n:node> set hostname=lztomcat2b
clzc:zc2-4n:node> add net
clzc:zc2-4n:node:net> set address=lztomcat2b
clzc:zc2-4n:node:net> set physical=e1000g0
clzc:zc2-4n:node:net> end
clzc:zc2-4n:node> end clzc:zc2-4n> add node
clzc:zc2-4n:node> set physical-host=ptomcat3
clzc:zc2-4n:node> set hostname=lztomcat3b
clzc:zc2-4n:node> add net
clzc:zc2-4n:node:net> set address=lztomcat3b
clzc:zc2-4n:node:net> set physical=e1000g0
clzc:zc2-4n:node:net> end
clzc:zc2-4n:node> end clzc:zc2-4n> add node
clzc:zc2-4n:node> set physical-host=ptomcat4
```

```
clzc:zc2-4n:node> set hostname=lztomcat4b
clzc:zc2-4n:node> add net
clzc:zc2-4n:node:net> set address=lztomcat4b
clzc:zc2-4n:node:net> set physical=e1000g0
clzc:zc2-4n:node:net> end
clzc:zc2-4n:node> end
```

physical-hostは該当するグローバル・クラスタ・ノードの名前です。hostnameはゾーン・クラスタ・ノードの名前であり、Resource Group Manager（RGM）ノード・リストにクラスタ・アプリケーションを実行する場所を指定するときに使用します。/etc/hostsファイルにはIPアドレスと併せてホスト名を入力することを推奨します。

netリソースには、ゾーンの起動時に指定されたNICに接続するIPアドレスを指定します。管理者は指定されたIPアドレスまたはホスト名を使用して、ゾーン・クラスタ・ノードにログインできます。

ゾーン・クラスタ・ゾーンの名前は常にゾーン・クラスタ名と同じであり、ゾーン名はホスト名とは異なります。

4. Oracle RAC 10g以降では各ノードにIPアドレスが必要です。このIPアドレスはVIPアドレスと呼ばれます。これらのIPアドレスを指定するには、クラスタ・レベルのコンテキスト内でadd netサブコマンドを使用します。

```
clzc:zc2-4n> add net
clzc:zc2-4n:net> set address=lztomcat1b-vip
clzc:zc2-4n:net> end
clzc:zc2-4n> add net
clzc:zc2-4n:net> set address=lztomcat2b-vip
clzc:zc2-4n:net> end
clzc:zc2-4n> add net
clzc:zc2-4n:net> set address=lztomcat3b-vip
clzc:zc2-4n:net> end
clzc:zc2-4n> add net
clzc:zc2-4n:net> set address=lztomcat4b-vip
clzc:zc2-4n:net> end
```

注 – Oracle9i RACを使用している場合、管理者は論理ホストで使用するIPアドレスを指定します。

5. 次に、「CrsHome2」および「OraHome2」（ディレクトリはそれぞれ「/db_qfs/CrsHome」 および「/db_qfs/OraHome」）を含む２つのQFS共有ファイル・システムは、ゾーン・クラスタzc2-4nの構成に追加されます。11g Release 2より前のリリースでは、Oracle Clusterwareインストールに対して共有QFSを使用できますが、11g Release 2でOracle Grid Infrastructureインストールに共有QFSを使用することはできません。ただし、Oracle RAC RDBMSインストールに対しては、すべてのリリースで共有QFSを使用できます。

```
clzc:zc2-4n> add fs
clzc:zc2-4n:fs> set dir=/db_qfs/CrsHome
clzc:zc2-4n:fs> set special=CrsHome2
clzc:zc2-4n:fs> set type=samfs
clzc:zc2-4n:fs> end clzc:zc2-4n> add fs
clzc:zc2-4n:fs> set dir=/db_qfs/OraHome
clzc:zc2-4n:fs> set special=OraHome2
clzc:zc2-4n:fs> set type=samfs
clzc:zc2-4n:fs> end
```

ファイル・システムを使用するときのベスト・プラクティスは、このファイル・システムのみをゾーン・クラスタ内に構成することです。ゾーン・クラスタ内でストレージ・デバイスが明示的に構成されていない場合、このストレージ・デバイスに対して危険なioctl操作が実行される可能性はありません。ゾーン・クラスタ内に構成されているストレージ・デバイスがなくても、ファイル・システムはパフォーマンスが低下することなく適切に動作します。

dirプロパティはゾーン・クラスタのルート・パスに関連するマウント・ポイントを指定します。このマウント・ポイントはゾーン・クラスタ内部からファイル・システムにアクセスするために使用されます。ゾー

ン・クラスタzc2-4nはオペレーティング・システム・ボリューム上のOracle ASMを使用します。ここで、このボリュームはクラスタ機能を使用したOracle Solaris Volume Managerによって提供されています。Oracle ASMはDIDデバイスを使用して構成することもできます。

注 – Oracle RACをゾーン・クラスタ内で実行する場合、inherit-pkg-dirディレクティブを/optファイル・システムに対して使用することはできません。Oracle RACでは、ゾーン・クラスタ内の/optファイル・システムに対する読取り能力と書込み能力の両方が必要です。競合を防止するため、各ゾーン・クラスタに独自の/optファイル・システムを構成する必要があります。

6. ゾーン・クラスタzc2-4nは、Oracle Solaris Volume Managerがミラー化したd5、d6、d7ボリュームをOracle Cluster Registryディスクおよび投票ディスクとして使用します。Oracle ASMはミラー化されていないOracle Solaris Volume Managerデバイスのd71～74を使用します（ミラー化デバイスが必要な場合、デバイス間でのミラー化をOracle ASMが行うように設定できます）。

```
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/zc2dg/rdsk/d5"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/zc2dg/rdsk/d6"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/zc2dg/rdsk/d7"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/zc2dg/rdsk/d71"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/zc2dg/rdsk/d72"
```

```
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/zc2dg/rdsk/d73"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/zc2dg/rdsk/d74"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/shared/1/rdsk/d5"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/shared/1/rdsk/d6"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/shared/1/rdsk/d7"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/shared/1/rdsk/d71"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/shared/1/rdsk/d72"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/shared/1/rdsk/d73"
clzc:zc2-4n:device> end
clzc:zc2-4n> add device
clzc:zc2-4n:device> set match="/dev/md/shared/1/rdsk/d74"
clzc:zc2-4n:device> end
```

7. Oracle Solarisはすべての非大域ゾーンに一連の権限を提供しています。set limitprivコマンドを使用すると追加の権限を指定できます。Oracle RACではデフォルト提供されていない権限が必要であり、その権限は次のコマンドで設定します。

```
For 9i, 10gR2 and 11gR1 use:
clzc:zc2-4n> set limitpriv="default,proc_priocntl,proc_clock_highres"
 For 11gR2 use:
clzc:zc2-4n> set
  limitpriv="default,proc_priocntl,proc_clock_highres,sys_time"
```

proc_lock_memoryはメモリのロック・ダウン能力を付与する権限であり、データベース・インスタンスは構成済みのSGAメモリをロックします。この権限はデフォルトでゾーン・クラスタ内に存在しています。

8. CPUリソースを割り当てます。リソース管理はクラスタ・アプリケーション集約での重要な部分です。CPUの数を指定することで、管理者はゾーン・クラスタ内のアプリケーションに対するソフトウェア・ライセンス費用を調整できます。Oracle Solaris OSはゾーン・クラスタ内でのCPU管理に対して複数のアプローチをサポートしています。ライセンス費用を考慮に入れない場合は、フェアシェア・スケ

ジューラを使用すると相対的なCPUシェアを割り当てることができます。importanceパラメータには、Oracle Solaris OSを使用するときの重要度を指定します。これにより、CPUの割当てでどのゾーンがより重要であるかが決まります。

```
clzc:zc2-4n> add dedicated-cpu
clzc:zc2-4n:dedicated-cpu> set ncpus=8
clzc:zc2-4n:dedicated-cpu> set importance=20
clzc:zc2-4n:dedicated-cpu> end
```

9. メモリ・リソースを割り当てます。データベースをサポートするゾーン・クラスタに対しては、メモリ・サイズを明示的に制御することを推奨します。メモリ制限を設定する場合、実行可能ファイルとデータの両方に対する領域を見込む必要があります。次のコマンドはゾーン・クラスタが使用するメモリのサイズを明示的に指定しています。

```
clzc:zc2-4n> add capped-memory
clzc:zc2-4n:capped-memory> set physical=4G
clzc:zc2-4n:capped-memory> set swap=5G
clzc:zc2-4n:capped-memory> set locked=4G
clzc:zc2-4n:capped-memory> end
```

10. 次のコマンドは構成の完了を知らせるために使用します。

```
clzc:zc2-4n> commit
clzc:zc2-4n> exit
```

11. 以上で、ゾーン・クラスタzc2-4nの構成が完了しました。次のコマンドはゾーン・クラスタにシステム・ソフトウェアをインストールします。

```
# clzonecluster install zc2-4n
```

ゾーン・クラスタ内のすべてのゾーンを並行してインストールしますが、処理にはかなりの時間がかかります。

12. 以上で、ゾーン・クラスタの構成とシステム・ソフトウェアのインストールが完了しました。次のコマンドはすべてのゾーン・クラスタ・ノードを起動します。

```
# clzonecluster boot zc2-4n
```

はじめてゾーン・クラスタを起動する場合、sysidcfg情報の設定を含む初期化処理が実行されます。初期化処理が完了すると、ゾーン・クラスタは自動的に再起動されます。

13. ゾーン・クラスタzc2-4nが実行中になりました。ゾーン・クラスタのステータスを参照するには、Oracle Solaris Cluster Managerのグラフィカル・ユーザー・インタフェース（GUI、ブラウザ・ベースのユーザー・インタフェースとも呼ばれる）を使用します。

図11に、ゾーン・クラスタzc2-4nのステータスを調べる場合のOracle Solaris Cluster Managerの出力画面を示します。

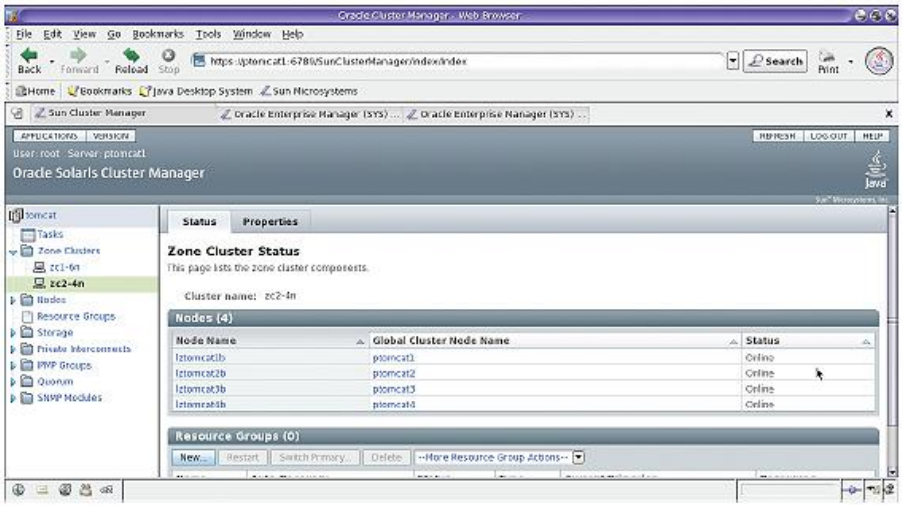

図11. スクリーンショット – ゾーン・クラスタのステータス

Oracle Databaseの構成

ここまでに、ゾーン・クラスタzc2-4nを使用してアプリケーションをサポートする準備が整いましたが、まだどのアプリケーションも構成されていません。Oracle RACデータベースはゾーン・クラスタ内に手動でインストールして構成できますが、Oracle Solaris Clusterは作業を容易にするために、ゾーン・クラスタ内のOracle RACに対してデータ・サービス構成ウィザードを提供しています。構成ウィザードはゾーン・クラスタ内で実行中のOracle RACの要件を認識し、管理者が実行する必要のある手順を減らします。構成ウィザードはゾーン・クラスタ内で使用できるリソースを自動的に検出し、適切な段階で提示します。構成ウィザードを使用することを強く推奨します。

1. Oracle UDLMを使用している場合、はじめにOracleユーザー・アカウントを作成してからOracle UDLMソフトウェアをインストールします。

注 – SKGXNを使用する場合、この時点で必要な作業はありません。SKGXNソフトウェアは残りのOracle RACソフトウェアに付属しています。

2. Oracle Solaris Cluster Manager GUIを起動します。
   Oracle RAC用Oracle Solaris Cluster構成ウィザードはOracle Solaris Clusterでサポートされているすべての Oracle RACバージョンに対応しており、サポートされるすべてのストレージ・トポロジを構成できます。このウィザードはOracle Solaris Cluster Manager GUIの一部に含まれています。図12に示すウィザードの最初の画面では、ゾーン・クラスタzc2-4nが選択されています。

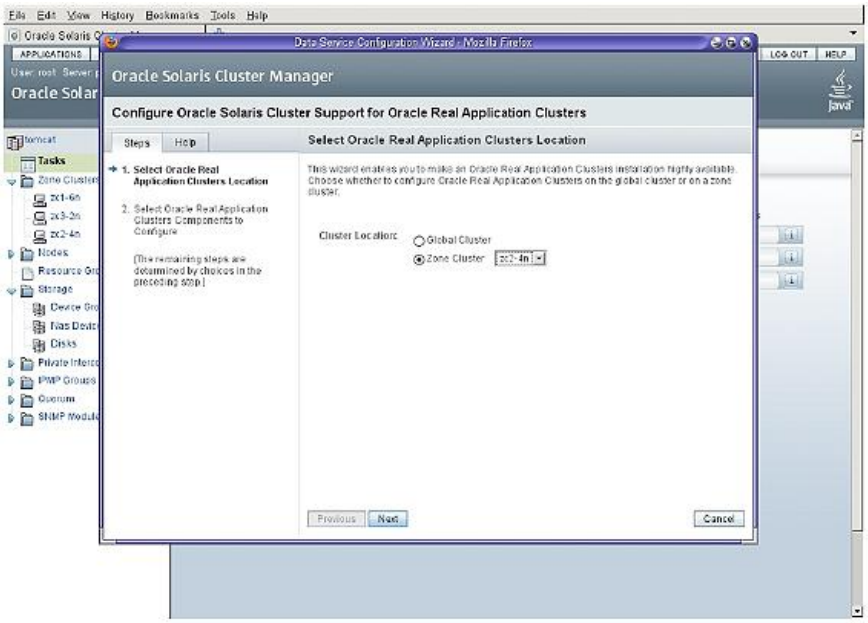

図12.スクリーンショット – Oracle RACロケーションの選択

Oracle Solaris Cluster Manager GUIと、Oracle Solaris Cluster Managerの下で稼働するすべてのウィザードは、大域ゾーン内のグローバル・クラスタ内で実行されます。

3. 次のOracle RAC用Oracle Solaris Cluster構成ウィザードの手順では、以下の6つの構成領域を設定します。
   - Oracle RACフレームワークのリソース・グループ
   - マルチオーナー・ボリューム・マネージャ・フレームワークのリソース・グループ
   - Oracleファイル用のストレージ・リソース
   - Oracle Clusterwareフレームワークのリソース
   - Automatic Storage Management
   - Oracle RACデータベース・インスタンス用のリソース

   それぞれの領域が、必須作業の各部分に対応する一種のウィザードを構成しています。これらのサブウィザードは一連の画面を通じてその時点で必要なものを明らかにし、可能な選択肢を表示して決定を下すように要求します。

処理の終わり近くで、サブウィザードは選択済みの項目を表示します。管理者はここで一部の項目を変更できます。変更するには元に戻る必要のある項目もあります。すべてに満足したら、選択を承認します。ウィザードは適切なコマンドを実行し、そのコマンドと最終的な結果を次の画面に表示します。

以降の手順では、いくつかのサブウィザードを詳しく確認していきます。

4. 最初のサブウィザードはRAC Framework Resource Groupサブウィザードです。このサブウィザードは最初に実行する必要があり、ゾーン・クラスタごとに1回実行されます。RAC Framework Resource Groupサブウィザードを使用して、Oracle RACデータベースをホストするゾーン・クラスタ・ノードを選択し、Oracle RACのサポートにSKGXNまたはOracle UDLMのどちらを使用するかを選択します。このサブウィザードはOracle RACフレームワーク・リソース・グループと、このリソース・グループに含まれるリソースを構成します。

   図13に示すRAC Framework Resource Groupサブウィザードのスクリーンショットでは、このサブウィザードの各手順が一覧表示されています。

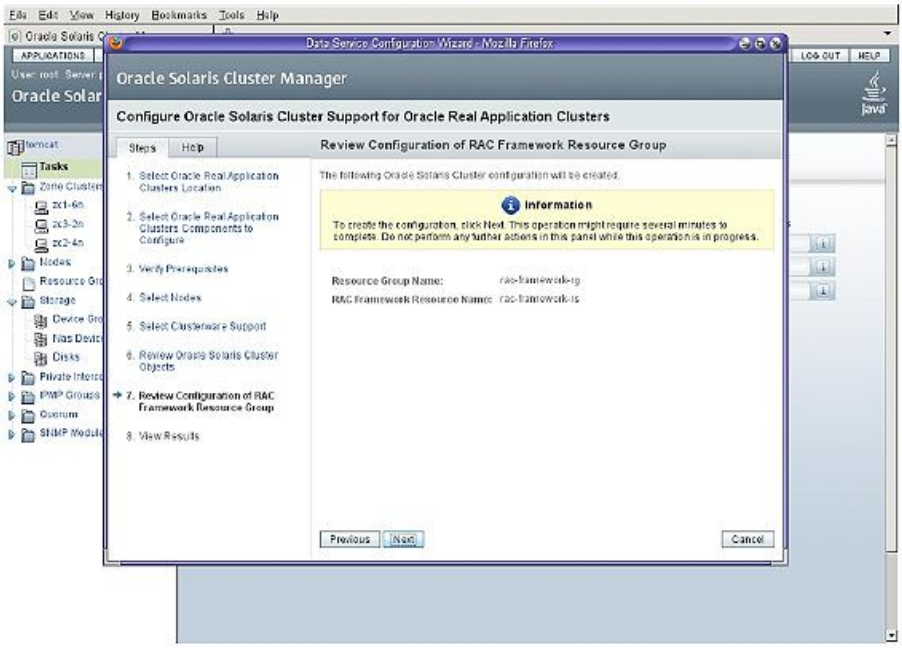

図13.スクリーンショット - RAC Framework Resource Groupサブウィザード

5. 次に、Multi-owner Volume-Manager Framework Resource Groupサブウィザードが実行されます。このサブウィザードはRAC Framework Resource Groupサブウィザードの前後どちらでも実行できますが、Storage Resources for Oracle Filesサブウィザードの前に実行する必要があります。このサブウィザードは選択したボリューム・マネージャのサポートに必要なリソースを作成します。図14にこのサブウィザードのスクリーンショットを示します。

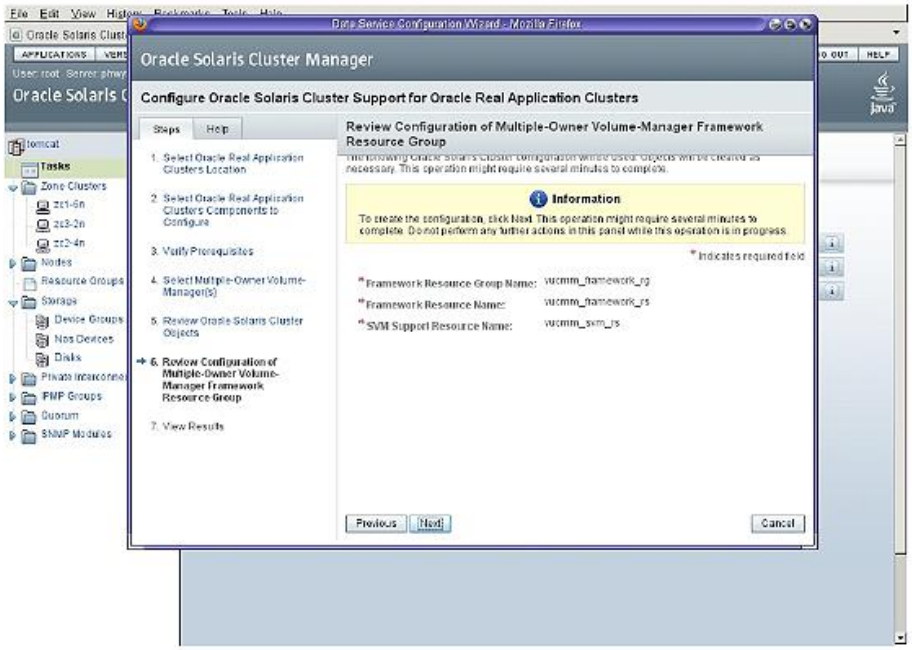

図14.スクリーンショット - Multi-owner Volume-Manager Framework Resource Groupサブウィザード

6. 次に、Storage Resources for Oracle Filesサブウィザードが実行されます。このサブウィザードは、Multi-owner Volume-Manager Framework Resource Groupサブウィザードの完了後にのみ実行できます。管理者はこのサブウィザードを使用してストレージ・トポロジを選択します。選択されたストレージ・トポロジに関する情報はウィザードが実行されたノード上のファイルに書き込まれます。
   オプションは次のとおりです。

   - クラスタ機能を備えたOracle Solaris Volume Manager上のQFS共有ファイル・システム
   - ハードウェアRAID上のQFS共有ファイル・システム
   - クラスタ機能を備えたOracle Solaris Volume Manager
   - ボリューム・マネージャのないハードウェアRAID（このオプションは現在、Oracle ASMを使用したOracle RACのサポート時に使用されています）

   Storage Resources for Oracle Filesサブウィザードでは、以下のファイルに対するストレージを指定します。

   - Clusterware実行可能ファイル
   - データベース実行可能ファイル
   - Clusterwareデータ
   - データベース・データ

1つのクラスタ内に複数のデータベースを配置できます。ここでは、サブウィザードを使用して複数のストレージ・ロケーションを指定しています。

図15に示すスクリーンショットでは、このサブウィザードで実行された手順の一覧が表示されています。同じスクリーンショットのウィンドウには、サブウィザードの処理中に管理者が下した決定に応じて、システムが実行したすべてのコマンドが表示されています。

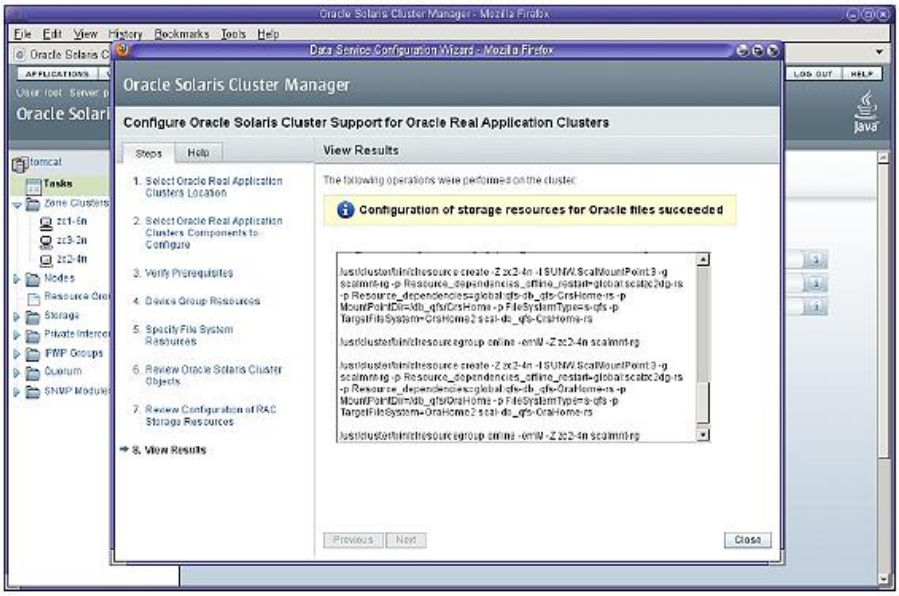

図15.スクリーンショット – Storage Resources for Oracle Filesサブウィザード

7. ここで、管理者は構成ウィザードからいったん離れる必要があります。
8. ゾーン・クラスタ・ノードにログインし、Oracle RACソフトウェアをインストールします。Oracle RACから見ると、ゾーン・クラスタはあたかも従来型のクラスタのように見えます。グローバル・クラスタ内のインストールと同じ方法で、Oracleのインストールおよび構成手順を実行します。Oracleソフトウェアのインストールは通常、Oracle Database製品に付属しているOracle Universal Installerを起動して1つのノードから実行します。Oracle Universal Installerは選択されたすべてのクラスタ・ノードにClusterwareとOracle RDBMSソフトウェアをインストールします。

   図16に、Oracle RACインストール・ソフトウェアの実行中のスクリーンショットを示します。この例では、管理者がクラスタ名を指定しています。

30

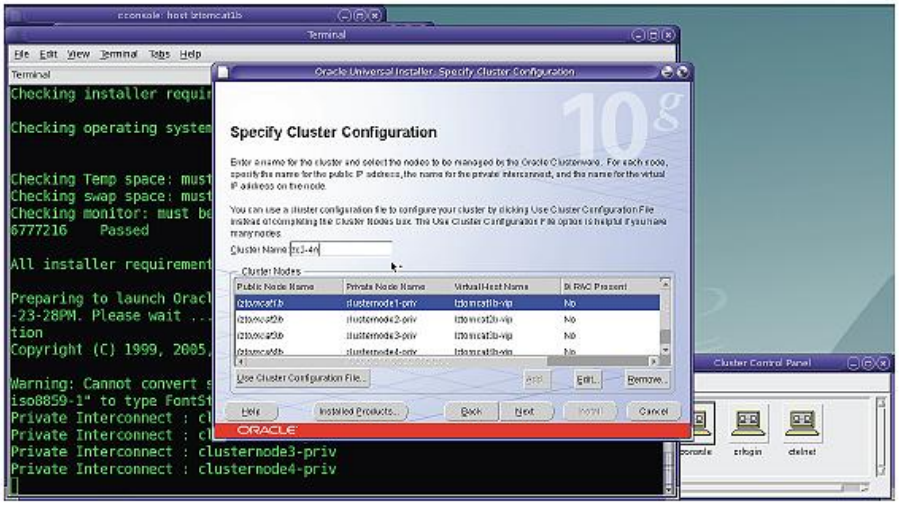

図16.スクリーンショット – クラスタ構成の指定

9. データベース作成の一部でDatabase Configuration Assistant（dbca）が使用され、このプロセスの一部でユーザーがOracle ASMを構成します。この例で、ゾーン・クラスタzc2-4nはOracle ASMを使用してデータを管理します。次の手順では、図17に示すように、Oracle Solaris Volume Managerボリュームを使用してOracle ASMディスク・グループを作成します。

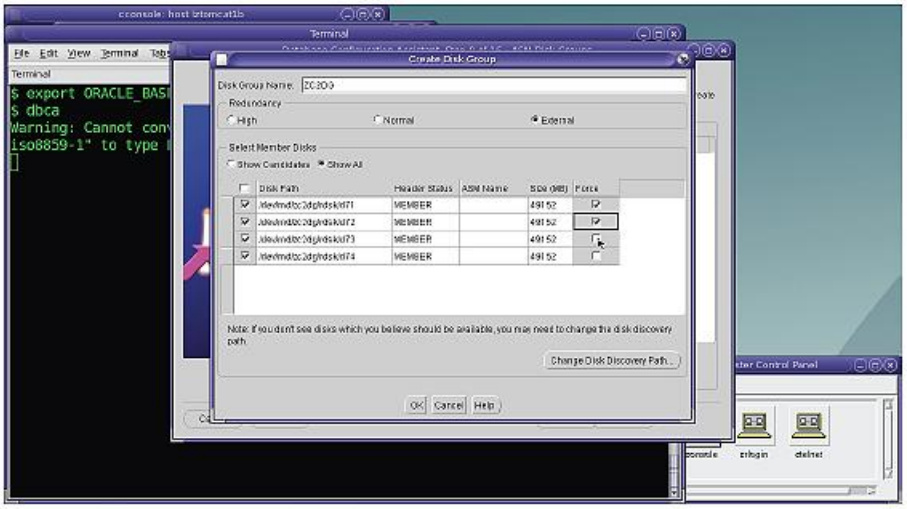

図17.スクリーンショット – Oracle Solaris Volume Managerデバイスを使用したOracle ASMディスク・グループの作成

10. 次に、Oracle RACデータベースを1つまたは複数作成します。図18にデータベース作成プロセスの一部のスクリーンショットを示します。

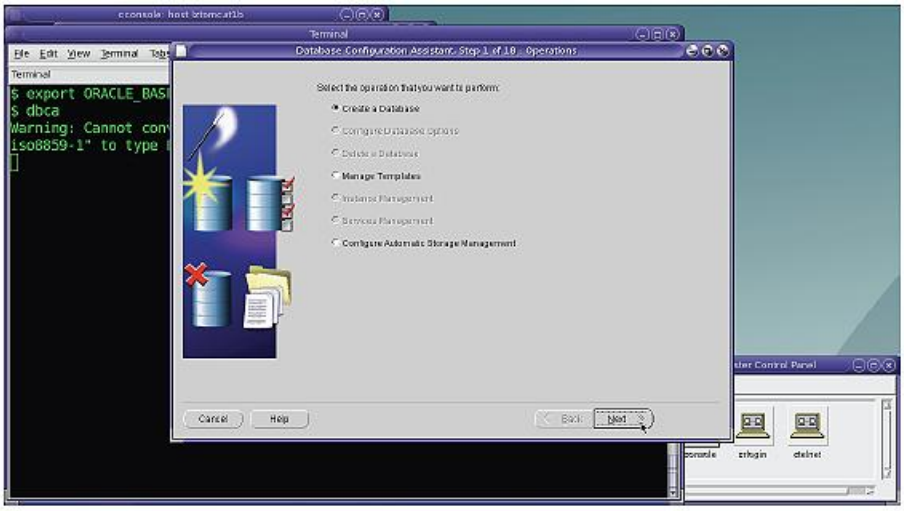

図18.スクリーンショット – データベース作成

11. 管理者はここでゾーン・クラスタから離れて、グローバル・クラスタにログインします。Oracle Solaris Cluster構成ウィザードはブラウザ・ベースのユーザー・インタフェースです。ブラウザ・ウィンドウを1つ開いたままにして、もう1つのウィンドウを使用してゾーン・クラスタにログインできます。重要なのは、大域ゾーンから作業を行うことです。
12. Oracle RAC用Oracle Solaris Cluster構成ウィザードを開始して、Oracle Clusterware Framework Resourceサブウィザードを選びます。このサブウィザードはClusterwareのインストール・ディレクトリのプロパティを指定するために使用します。
13. Oracle RAC用Oracle Solaris Cluster構成ウィザードをもう一度開始して、Automatic Storage Managementサブウィザードを選びます。図19に、サブウィザードの使用中に取得したスクリーンショットを示します。このサブウィザードは以下を含む各種ASMプロパティを指定するために使用します。
    - Oracle ASMインスタンスのOracleホーム・ディレクトリ
    - Oracle ASMインスタンス番号

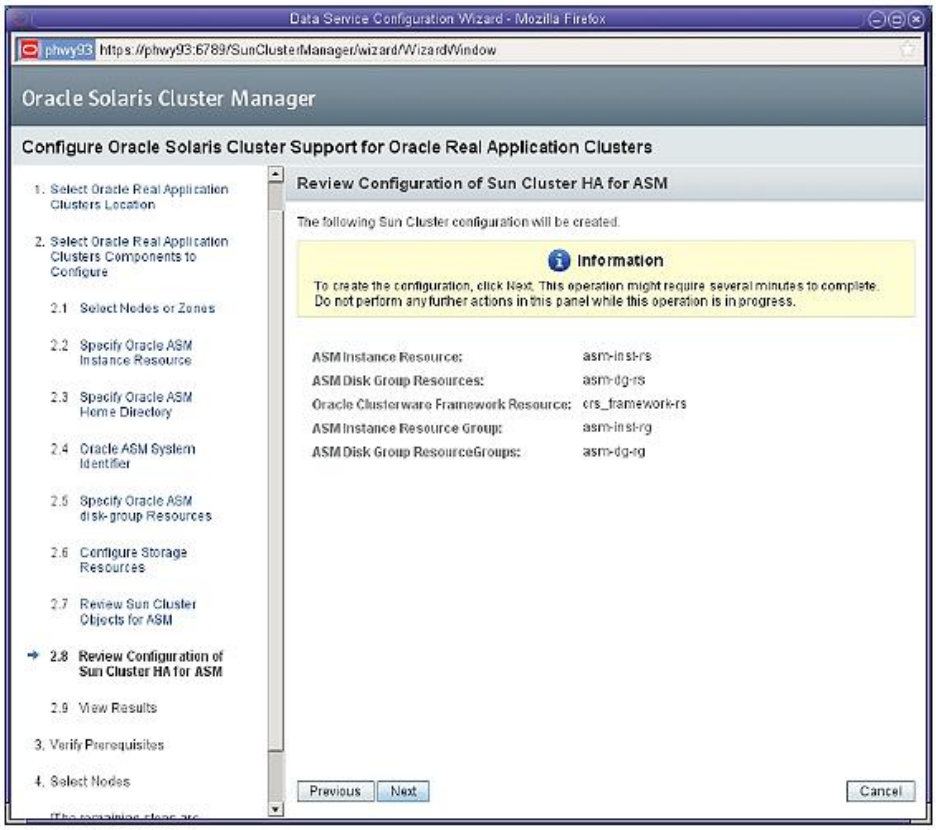

図19.スクリーンショット – Automatic Storage Managementサブウィザード

14. 最後にもう一度Oracle RAC用Oracle Solaris Cluster構成ウィザードを開始して、Resources for Oracle Real Application Clusters Database Instancesサブウィザードを選びます。このサブウィザードは以下を含む各種のOracleプロパティを指定するために使用します。
    - Oracle RACコンポーネント（Oracle9i RACのサーバーとリスナーのみ）
    - Oracleホーム・ディレクトリ
    - Clusterwareホーム・ディレクトリ（Oracle RAC 10g/11gのみ）
    - Oracleシステム識別子
    - 論理ホスト名（Oracle9i RACのみ）
    - ストレージ
    - データベース名

    このプロセスにはリソース・グループやリソース、リソース・グループ・アフィニティ、リソース依存性の作成が含まれます。

    詳しくは、『*Oracle Solaris Cluster Data Service for Oracle RAC Guide for Oracle Solaris OS*』を参照してください。
15. Oracle RACデータベースを使用する準備が整いました。次に、Oracle RACデータベースにOrder Entry

スキーマをロードします。

16. クライアント・マシンに移動して、ゾーン・クラスタzc2-4n上のデータベースでSwingbench Order Entryワークロードを起動します。

    図20にこのワークロードのスクリーンショットを示します。

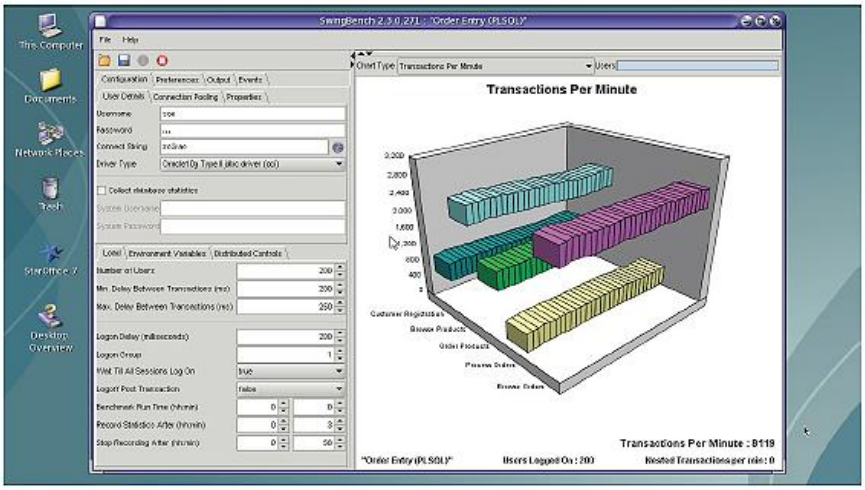

図20.スクリーンショット – ワークロード

17. 図21に示すとおり、ゾーン・クラスタzc2-4n上のOracle Enterprise Managerを使用して、データベースに関する情報を監視できます。

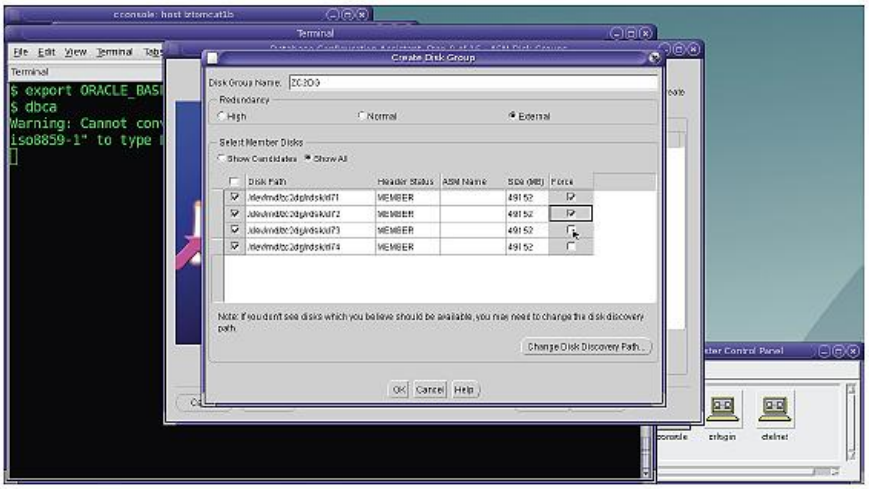

図21.スクリーンショット – データベース関連情報の表示

18. Oracle Solaris Cluster Manager GUIを使用すると、ゾーン・クラスタzc2-4nに関する各種のステータス情報を監視できます。図22に、ゾーン・クラスタzc2-4n上のRGMリソースに関するステータス情報を示します。

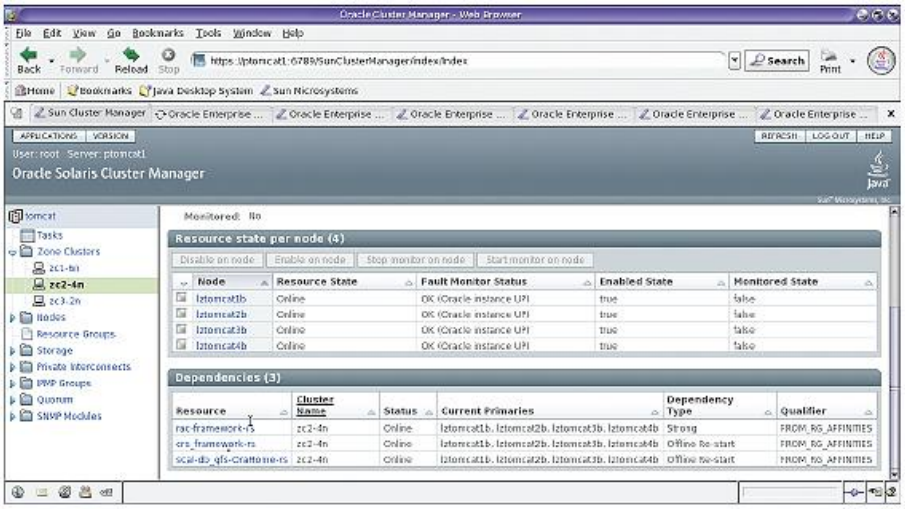

図22.スクリーンショット – ゾーン・クラスタzc2-4nのステータス

19. 以上で、ゾーン・クラスタzc2-4nを使用する準備が整いました。

Oracle RAC構成

このセクションでは、ゾーン・クラスタがサポートする各種のOracle RAC構成について説明します。構成はデータ・ストレージの種類によって異なり、以下を含みます。

- Oracle Solaris Volume Managerを使用したQFS共有ファイル・システム上のOracle RAC
- ハードウェアRAIDを使用したQFS共有ファイル・システム上のOracle RAC
- Oracle Solaris Cluster向けOracle Solaris Volume Manager上のOracle RAC
- Automatic Storage Management上のOracle RAC - Oracle RAC 10gおよび11gをサポート、Oracle9i RACはサポート対象外（Oracle ASMの導入はOracle RAC 10gリリースから）
- NAS上のOracle RAC
- ハードウェアRAID上のOracle RAC

注 - 同じ物理クラスタ上にあっても、ゾーン・クラスタごとに異なるOracle RAC構成をサポートできます。

Oracle RACソフトウェアはどのゾーン・クラスタ・ノードからでも、データ用に使用されるストレージ・サブシステムに常に直接アクセスできます。場合によって、ストレージ・サブシステムにはファイル・システムやボリューム・マネージャ・ボリュームなどの複数の階層が含まれます。ストレージ・サブシステムの最上位層は常にゾーン・クラスタ内に存在します。たとえば、ファイル・システムはゾーン・クラスタ内に存在しますが、ボリューム・マネージャ・ボリュームはグローバル・クラスタ内にあります。多くのストレージ構成において、Oracle Solaris Clusterはストレージ・サブシステム・コンポーネント（ファイル・システム、ストレージ・デバイスなど）の可用性を管理します。リソースとリソース関係はすべてのノードで同じストレージ構成になります。簡略化のため、以降のセクションで示す図には1つのノード上のリソースとリソース関係のみを表示します。

ゾーン・クラスタはOracle9i、10g、11gのOracle RACをサポートしており、ストレージ構成はほとんど同一です。Oracle RAC 10gおよび11gはどのようなストレージ構成でも同じ方法でサポートされます。したがって、10gおよび11gリリースは各ストレージ構成の共通セクションに集約されています。Oracle9i RACはOracle RAC 10gおよび11gとは異なるため、Oracle9i RAC構成は個別のサブセクションに記載しています。

一部のストレージ・トポロジは実行可能ファイルに対して使用できないか、または一般に使用されていません。このような場合は、サブセクションに実行可能ファイルのサポート方法を記載していません。管理者は、別のストレージ・ソリューションを使用して実行可能ファイルをホストできます。

ここからは、それぞれの構成に適したサブセクションを参照してください。

- "Oracle Solaris Volume Manager上のQFSを使用したOracle RAC 10g/11g"
- "RAID上のQFSを使用したOracle RAC 10g/11g"
- "Oracle Solaris Volume Manager上のOracle RAC 10g/11g"
- "Oracle ASM上のOracle RAC 10g/11g"
- "NAS上のOracle RAC 10g/11g"
- "Oracle Solaris Volume Manager上のQFSを使用したOracle9i RAC"

- "RAID上のQFSを使用したOracle9i RAC"
- "Oracle Solaris Volume Manager上のOracle9i RAC"
- "NAS上のOracle9i RAC"
- "ハードウェアRAID上のOracle9i RAC、Oracle RAC 10g/11g"

Oracle Solaris Volume Manager上のQFSを使用したOracle RAC 10g/11g

ゾーン・クラスタは、Oracle Solaris Cluster共有ディスク・セット向けのOracle Solaris Volume Manager上に構成されたQFS共有ファイル・システムを使用したOracle RAC 10g/11gをサポートしています。

図23の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。11g Release 2より前のリリースでは、Oracle Clusterwareインストールに対して共有QFSを使用できますが、11g Release 2でOracle Grid Infrastructureインストールに共有QFSを使用することはできません。ただし、Oracle RAC RDBMSインストールに対しては、すべてのリリースで共有QFSを使用できます。

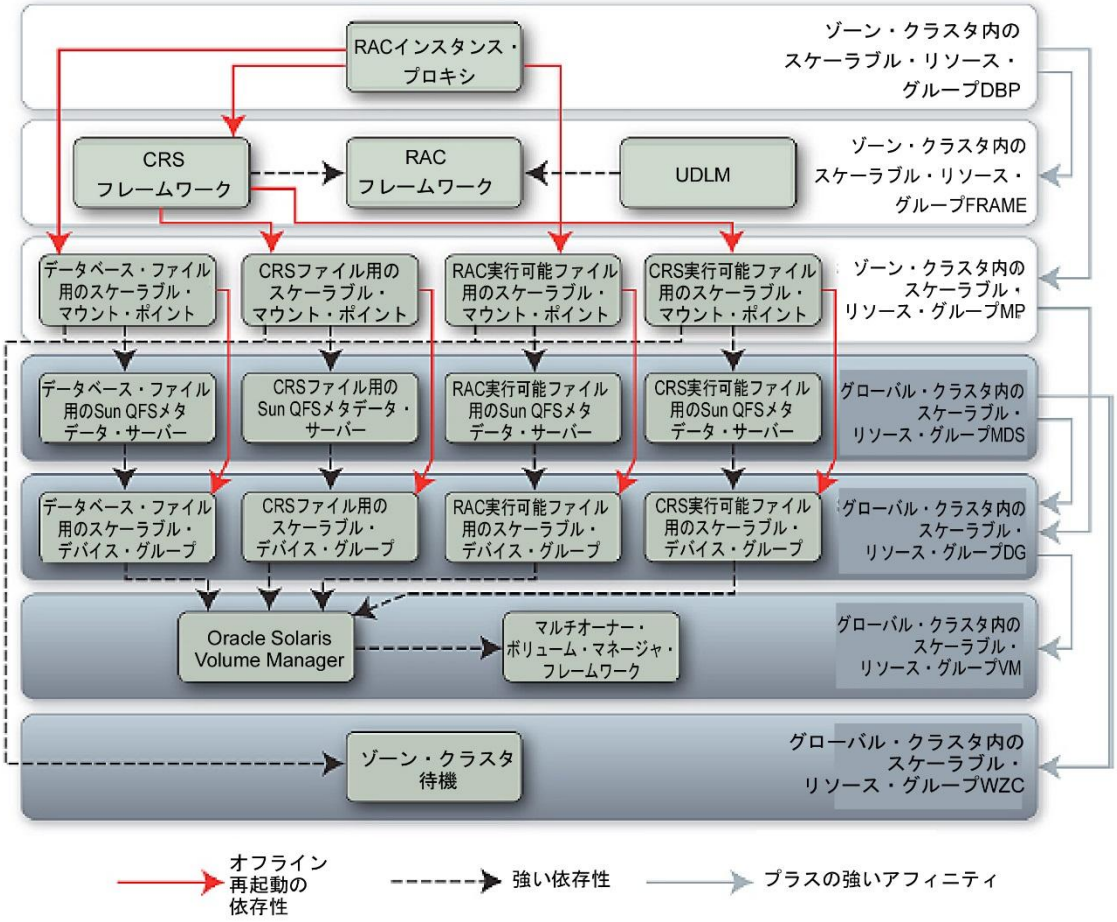

図23. Oracle Solaris Volume Managerを使用した共有QFS上のOracle RAC 10g/11gに対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図23を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。
2. 大域ゾーン内にスケーラブル・リソース・グループVMを作成します。このスケーラブル・リソース・グループ内にVUCMMフレームワーク用とOracle Solaris Volume Manager用のリソースを作成します。VMリソース・グループをオンライン状態にします。
3. 大域ゾーンで構成を実行している場合とまったく同じ方法でOracle Solaris Volume Manager共有ディスク・セットを作成します。図23に示すとおりにリソース・グループVM内のリソース依存性を設定します。
4. 大域ゾーン内にスケーラブル・リソース・グループDGを作成します。図23に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内でスケーラブル・デバイス・グループ用のリソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図23に示すとおりにこのリソース・グループ内のリソース依存性を指定します。DGリソース・グループをオンライン状態にします。
5. 大域ゾーンで構成を実行している場合とまったく同じ方法でQFS共有ファイル・システムを作成します。ファイル・システムをマウントしないでください。
6. ゾーン・クラスタを作成してQFS共有ファイル・システムに対するアクセス権を付与します。ゾーン・クラスタをインストールして起動します。
7. ファイル・システムのマウント・ポイントを作成し、QFS共有ファイル・システムをマウントします。
8. 大域ゾーン内にスケーラブル・リソース・グループWZCを作成し、新しく作成したゾーン・クラスタに対してゾーン・クラスタ待機（SUNW.wait_zc_boot）リソースを作成します。WZCリソース・グループをオンライン状態にします。図23に示すとおりにリソース・グループ・アフィニティを更新します。
9. 大域ゾーン内にフェイルオーバー・リソース・グループMDSを作成します。このフェイルオーバー・リソース・グループ内にQFSメタデータ・サーバー・リソースを作成します。図23に示すとおりに、このリソース・グループに対するリソース依存性を設定します。MDSリソース・グループをオンライン状態にします。
10. ゾーン・クラスタに入ります。
11. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
12. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図23に示すとおりのリソース依存性を設定します（CRSフレームワーク・リソースは後で作成します）。リソース・グループFRAMEをオンライン状態にします。
13. この手順ではスケーラブル・リソース・グループMPを取り上げます。システムはスケーラブル・マウント・ポイントのリソース依存性を作成時に検証します。この依存性はクラスタ間の依存性です。クラスタ間依存性はグローバル・クラスタの大域ゾーンからのみ指定できます。

    ゾーン・クラスタ内にスケーラブル・リソース・グループMPを作成します。次に、大域ゾーンに入ります。図23に示すとおりにリソース・グループMPに対してリソース・グループ・アフィニティを設定します。このゾーン・クラスタに属するスケーラブル・リソース・グループMP内に、スケーラブル・マウント・ポイント・リソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図23に示すとおりにこのリソース・グループ内のリソース依存性を指定します。リソース・グループMPをオンライン状態にします。ゾーン・クラスタに戻ります。
14. Oracle Clusterwareをインストールしてから、Oracle RACをインストールします。QFS共有ファイル・システム上では、11g Release 2より前のリリースに対してインストールできるのはOracle Clusterwareバイナリのみです。
15. スケーラブル・リソース・グループFRAME内にCRSフレームワーク・リソースを作成し、図23に示すとおりのリソース依存性を設定します。
16. データベースを作成します。
17. Oracle Clusterware内のファイル・システムに対してプロキシ・リソースを作成します。次に、11g Release 2より前の構成に対して、データベース・インスタンスに対するClusterwareリソースの依存性を更新して、上記プロキシ・リソースをノードごとに追加します。
18. ゾーン・クラスタ内にスケーラブル・リソース・グループDBPを作成し、図23に示すとおりのリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACイ

ンスタンス・プロキシ・リソースを作成し、図23に示すとおりのリソース依存性を設定します。リソース・グループDBPをオンライン状態にします。

RAID上のQFSを使用したOracle RAC 10g/11g

ゾーン・クラスタは、ハードウェアRAID上のQFS共有ファイル・システムを使用したOracle RAC 10g/11gをサポートしています。

注 – 現時点では、ハードウェアRAIDを表すリソースは存在しません。

図24の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。11g Release 2より前のリリースでは、Oracle Clusterwareインストールに対して共有QFSを使用できますが、11g Release 2でOracle Grid Infrastructureインストールに共有QFSを使用することはできません。ただし、Oracle RAC RDBMSインストールに対しては、すべてのリリースで共有QFSを使用できます。

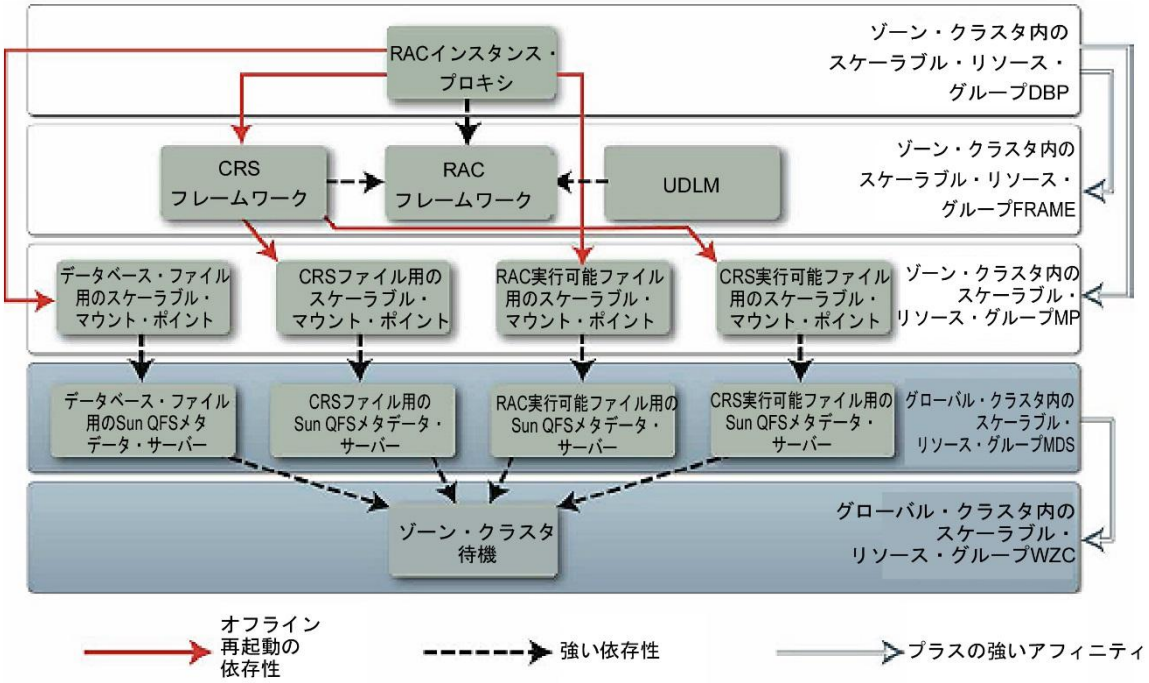

図24. ハードウェアRAIDを使用した共有QFS上のOracle RAC 10g/11gに対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図24を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。
2. 大域ゾーンで構成を実行している場合とまったく同じ方法でQFS共有ファイル・システムを作成します。ファイル・システムをマウントしないでください。
3. ゾーン・クラスタを作成してQFS共有ファイル・システムに対するアクセス権を付与します。ゾーン・クラスタをインストールして起動します。
4. ファイル・システムのマウント・ポイントを作成し、QFS共有ファイル・システムをマウントします。
5. 大域ゾーン内にスケーラブル・リソース・グループWZCを作成し、新しく作成したゾーン・クラスタに

対してゾーン・クラスタ待機（SUNW.wait_zc_boot）リソースを作成します。WZCリソース・グループをオンライン状態にします。

6. 大域ゾーン内にフェイルオーバー・リソース・グループMDSを作成します。図24に示すとおりにリソース・グループMPに対してリソース・グループ・アフィニティを設定します。このフェイルオーバー・リソース・グループ内にQFSメタデータ・サーバー・リソースを作成します。図24に示すとおりに、このリソース・グループに対するリソース依存性を設定します。MDSリソース・グループをオンライン状態にします。
7. ゾーン・クラスタに入ります。

8. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
9. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図24に示すとおりのリソース依存性を設定します（CRSフレームワーク・リソースは後で作成します）。リソース・グループFRAMEをオンライン状態にします。
10. この手順ではスケーラブル・リソース・グループMPを取り上げます。システムはスケーラブル・マウント・ポイントのリソース依存性を作成時に検証します。この依存性はクラスタ間の依存性です。クラスタ間依存性はグローバル・クラスタの大域ゾーンからのみ指定できます。
11. ゾーン・クラスタ内にスケーラブル・リソース・グループMPを作成します。次に、大域ゾーンに入ります。図24に示すとおりにリソース・グループMPに対してリソース・グループ・アフィニティを設定します。このゾーン・クラスタに属するスケーラブル・リソース・グループMP内に、スケーラブル・マウント・ポイント・リソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図24に示すとおりにこのリソース・グループ内のリソース依存性を指定します。リソース・グループMPをオンライン状態にします。ゾーン・クラスタに戻ります。
12. Oracle Clusterwareをインストールしてから、Oracle RACをインストールします。QFS共有ファイル・システム上では、11g Release 2より前のリリースに対してインストールできるのはOracle Clusterwareバイナリのみです。
13. スケーラブル・リソース・グループFRAME内にCRSフレームワーク・リソースを作成し、図24に示すとおりのリソース依存性を設定します。
14. データベースを作成します。
15. Oracle Clusterware内のファイル・システムに対してプロキシ・リソースを作成します。次に、11g Release 2より前の構成に対して、データベース・インスタンスに対するClusterwareリソースの依存性を更新して、上記プロキシ・リソースをノードごとに追加します。
16. ゾーン・クラスタ内にスケーラブル・リソース・グループDBPを作成し、図24に示すとおりのリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACインスタンス・プロキシ・リソースを作成し、図24に示すとおりのリソース依存性を設定します。リソース・グループDBPをオンライン状態にします。

41

Oracle Solaris Volume Manager上のOracle RAC 10g/11g

ゾーン・クラスタは、Oracle Solaris Cluster向けのOracle Solaris Volume Managerを使用したOracle RAC 10g/11gをサポートしています。この構成ではゾーンからディスク・セットに直接アクセスできます。この方法はサポートされていますが、ゾーン内のプログラムからデバイスに対して直接ioctl関数コールを実行できることを管理者が認識しておく必要があります。

図25の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。

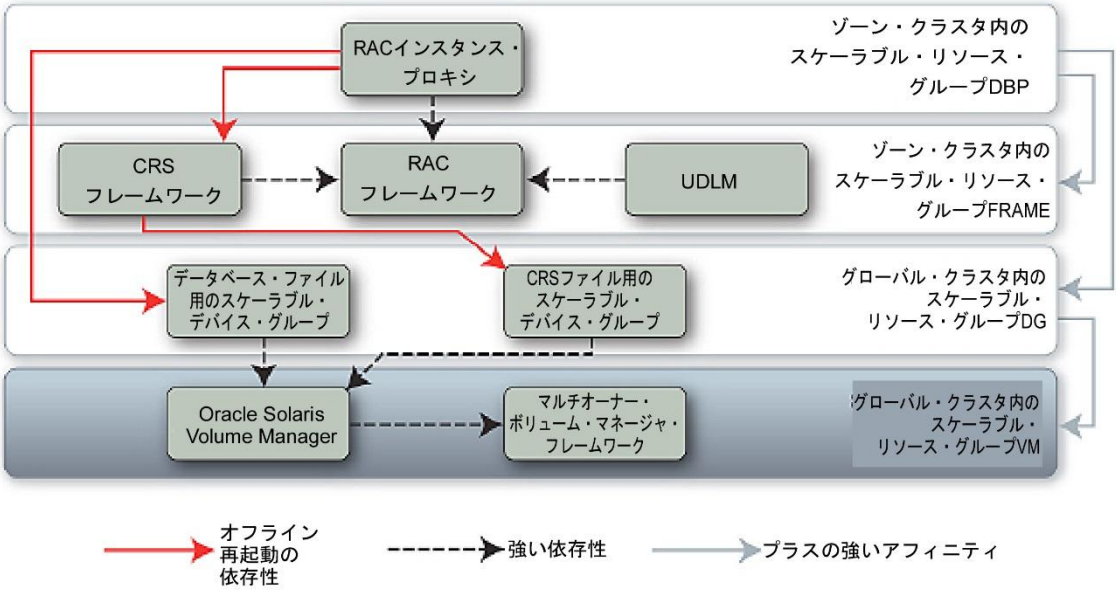

図25.Oracle Solaris Volume ManagerのOracle RAC 10g/11gに対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図25を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。
2. 大域ゾーン内にスケーラブル・リソース・グループVMを作成します。このスケーラブル・リソース・グループ内にVUCMMフレームワーク用とOracle Solaris Volume Manager用のリソースを作成します。VMリソース・グループをオンライン状態にします。
3. 大域ゾーンで構成を実行している場合とまったく同じ方法でOracle Solaris Volume Manager共有ディスク・セットを作成します。
4. ゾーン・クラスタを作成し、Oracle Solaris Volume Managerディスク・セットに対するアクセス権を付与します。ゾーン・クラスタをインストールして起動します。
5. ゾーン・クラスタに入ります。
6. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
7. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図25に示すとおりのリソース依存性を設定します（CRSフレームワーク・リソースは後で作成します）。リソース・グループFRAMEをオンライン状態にします。

8. ゾーン・クラスタ内にスケーラブル・リソース・グループDGを作成します。図25に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内でスケーラブル・デバイス・グループ用のリソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図25に示すとおりにこのリソース・グループ内のリソース依存性を指定します。DGリソース・グループをオンライン状態にします。
9. Oracle Clusterwareをインストールしてから、Oracle RACをインストールします。
10. スケーラブル・リソース・グループFRAME内にCRSフレームワーク・リソースを作成し、図25に示すとおりのリソース依存性を設定します。
11. データベースを作成します。
12. Oracle Clusterware内のデバイスに対してプロキシ・リソースを作成します。次に、11g Release 2より前の構成に対して、データベース・インスタンスに対するClusterwareリソースの依存性を更新して、上記プロキシ・リソースをノードごとに追加します。
13. ゾーン・クラスタ内にスケーラブル・リソース・グループDBPを作成し、図25に示すとおりのリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACインスタンス・プロキシ・リソースを作成し、図25に示すとおりのリソース依存性を設定します。リソース・グループDBPをオンライン状態にします。

ハードウェアRAIDを使用したOracle ASM上のOracle RAC 10g/11g

ゾーン・クラスタは、ハードウェアRAID上にデプロイされたOracle Automatic Storage Managementを使用したOracle RAC 10g/11gをサポートしています。

この構成ではゾーンからディスクに直接アクセスできます。Oracle ASMはDIDディスクを直接操作します。この方法はサポートされていますが、ゾーン内のプログラムからデバイスに対して直接ioctl関数コールを実行できることを管理者が認識しておく必要があります。

図26の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。

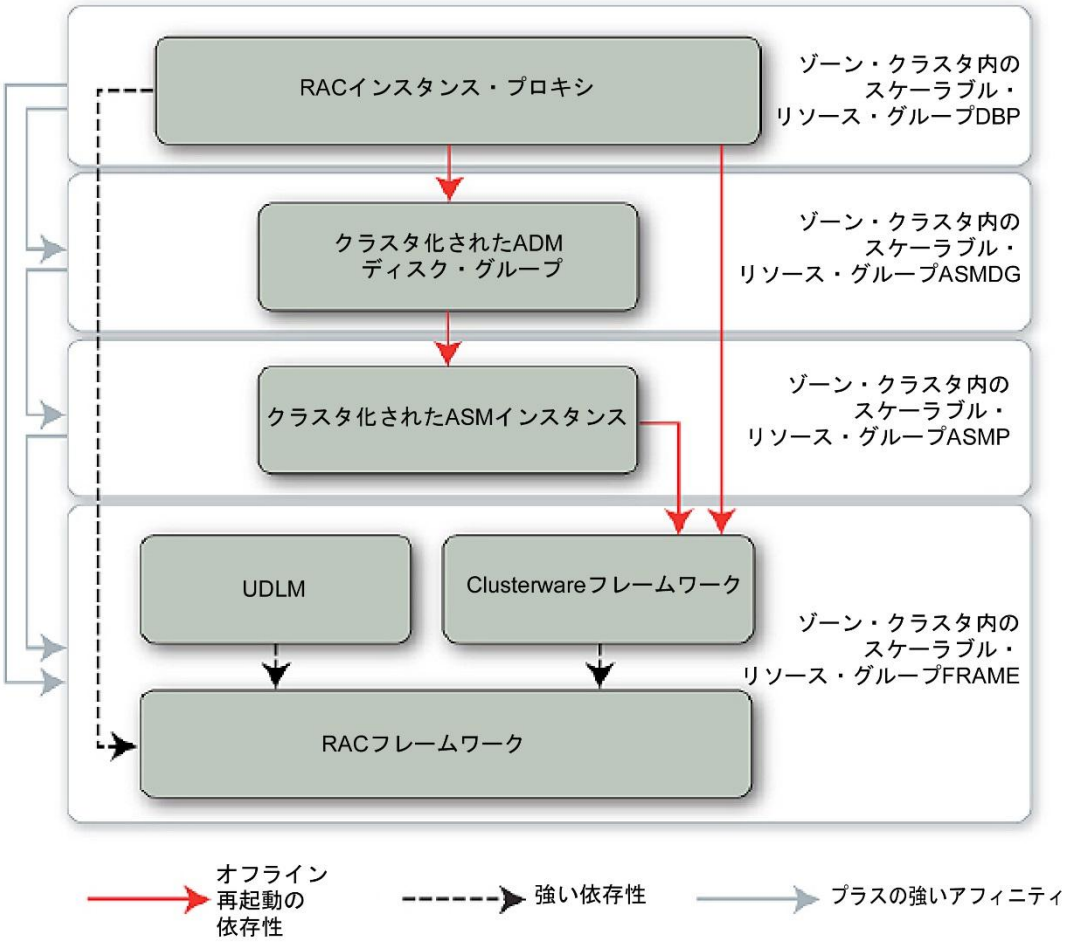

図26.ハードウェアRAIDを使用したOracle ASM上のOracle RAC 10g/11gに対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図26を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。
2. ゾーン・クラスタを作成してディスクに対するアクセス権を付与します。ゾーン・クラスタをインストールして起動します。
3. ゾーン・クラスタに入ります。
4. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
5. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図26に示すとおりのリソース依存性を設定しますリソース・グループFRAMEをオンライン状態にします。
6. Oracle Clusterwareをインストールします。11gを使用している場合、この時点でゾーン・クラスタ内に構成されているストレージ・デバイスをOracle ASMが操作できるように設定します。
7. スケーラブル・リソース・グループFRAME内にCRSフレームワーク・リソースを作成し、図26に示すとおりのリソース依存性を設定します。
8. Oracle RACをインストールします。10gを使用している場合、この時点でゾーン・クラスタ内に構成されているストレージ・デバイスをOracle ASMが操作できるように設定します。

9. ゾーン・クラスタ内にスケーラブル・リソース・グループASMPを作成します。図26に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にクラスタ化されたOracle ASMインスタンス・リソースを作成し、図26に示すとおりのリソース依存性を設定します。
10. ゾーン・クラスタ内にスケーラブル・リソース・グループASMDGを作成します。図26に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にクラスタ化されたOracle ASMディスク・グループ・リソースを作成し、図26に示すとおりのリソース依存性を設定します。11g Release 2を使用している場合は、SUNW.asm_diskgroupリソースの代わりにSUNW.scalable_asm_diskgroup_proxyリソースを構成します。
11. データベースを作成します。
12. ゾーン・クラスタ内にスケーラブル・リソース・グループDBPを作成します。図26に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACインスタンス・プロキシ・リソースを作成し、図26に示すとおりのリソース依存性を設定します。リソース・グループDBPをオンライン状態にします。

Oracle Solaris Volume Managerを使用したOracle ASM上のOracle RAC 10g/11g

ゾーン・クラスタは、Oracle Solaris Clusterボリューム向けのOracle Solaris Volume Manager上にデプロイされたOracle Automatic Storage Managementを使用したOracle RAC 10g/11gをサポートしています。

この構成ではゾーンからディスクに直接アクセスできます。Oracle ASMはOracle Solaris Clusterボリューム向けのOracle Solaris Volume Managerを直接操作します。この方法はサポートされていますが、ゾーン内のプログラムからデバイスに対して直接ioctl関数コールを実行できることを管理者が認識しておく必要があります。

図27の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。

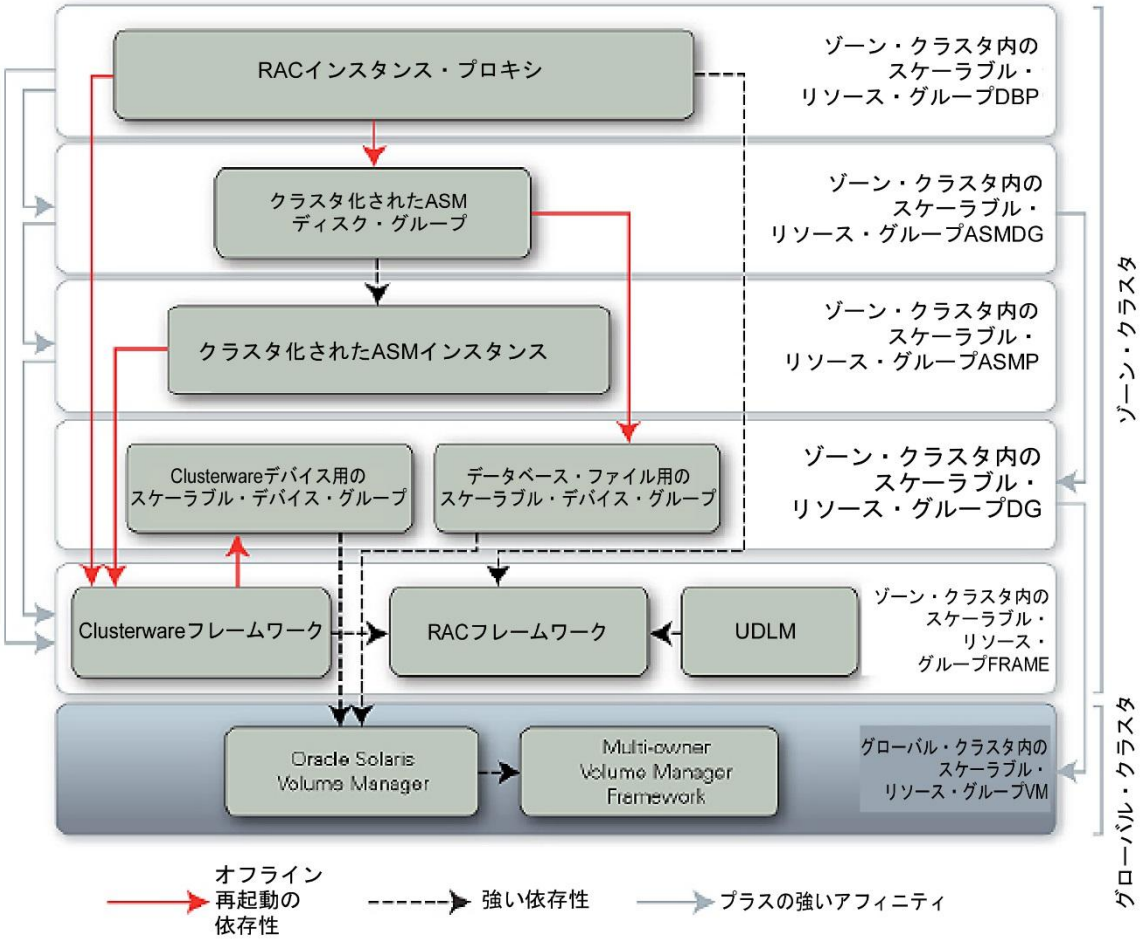

図27.Oracle Solaris Volume Managerを使用したOracle ASM上のOracle RAC 10g/11gに対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図27を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。
2. 大域ゾーン内にスケーラブル・リソース・グループVMを作成します。このスケーラブル・リソース・グループ内にVUCMMフレームワーク用とOracle Solaris Volume Manager用のリソースを作成します。VMリソース・グループをオンライン状態にします。
3. 大域ゾーンで構成を実行している場合とまったく同じ方法でOracle Solaris Volume Manager共有ディスク・セットを作成します。
4. ゾーン・クラスタを作成し、Oracle Solaris Volume Managerディスク・セットに対するアクセス権を付与します。ゾーン・クラスタをインストールして起動します。
5. ゾーン・クラスタに入ります。
6. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
7. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図27に示すとおりのリソース依存性を設定します（CRSフレームワーク・リソースは後で作成します）。リソース・グルー

プFRAMEをオンライン状態にします。

8. ゾーン・クラスタ内にスケーラブル・リソース・グループDGを作成します。図27に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内でスケーラブル・デバイス・グループ用のリソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図27に示すとおりにこのリソース・グループ内のリソース依存性を指定します。11gを使用しており、投票ディスクとOracle Cluster RegistryファイルがOracle ASMに格納されている場合、Oracle ASM用のデバイス・グループ・リソースのみが必要になります。DGリソース・グループをオンライン状態にします。
9. Oracle Clusterwareをインストールします。11gを使用している場合、この時点でゾーン・クラスタ内に構成されているストレージ・デバイスをOracle ASMが操作できるように設定します。
10. スケーラブル・リソース・グループFRAME内にCRSフレームワーク・リソースを作成し、図27に示すとおりのリソース依存性を設定します。
11. Oracle RACをインストールします。10gを使用している場合、この時点でゾーン・クラスタ内に構成されているストレージ・デバイスをOracle ASMが操作できるように設定します。
12. ゾーン・クラスタ内にスケーラブル・リソース・グループASMPを作成します。図27に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にクラスタ化されたOracle ASMインスタンス・リソースを作成し、図27に示すとおりのリソース依存性を設定します。
13. ゾーン・クラスタ内にスケーラブル・リソース・グループASMDGを作成します。図27に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にクラスタ化されたOracle ASMディスク・グループ・リソースを作成し、図27に示すとおりのリソース依存性を設定します。11g Release 2を使用している場合は、SUNW.asm_diskgroupリソースの代わりにSUNW.scalable_asm_diskgroup_proxyリソースを構成します。
14. データベースを作成します。
15. ゾーン・クラスタ内にスケーラブル・リソース・グループDBPを作成します。図27に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACインスタンス・プロキシ・リソースを作成し、図27に示すとおりのリソース依存性を設定します。リソース・グループDBPをオンライン状態にします。

NAS上のOracle RAC 10g/11g

ゾーン・クラスタは、Network Attached Storage（NAS）デバイス上のNFSマウント・ファイル・システムを使用したOracle RAC 10g/11gをサポートしています。NFSマウントはゾーン・クラスタの内部で実行されます。

スケーラブル・マウント・ポイント・リソース（SUNW.ScalMountPoint）はNASデバイス上のNFSファイル・システムのステータスを監視し、ゾーン・クラスタ内に存在します。

図28の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。

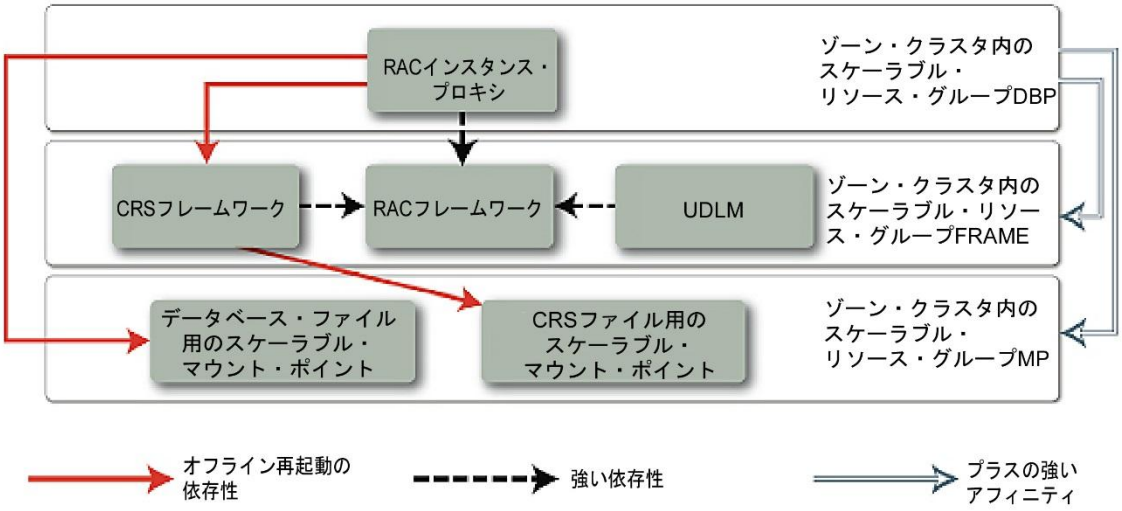

図28.NAS上のOracle RAC 10g/11gに対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図28を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。
2. ゾーン・クラスタを作成します。ゾーン・クラスタをインストールして起動します。
3. ゾーン・クラスタに入ります。
4. NASデバイス上にNFSファイル・システムに対するNFSクライアント・マウントを作成します。
5. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
6. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図28に示すとおりのリソース依存性を設定します（Clusterwareフレームワーク・リソースは後で作成します）。リソース・グループFRAMEをオンライン状態にします。
7. ゾーン・クラスタ内にスケーラブル・リソース・グループMPを作成します。このスケーラブル・リソース・グループ内にスケーラブル・マウント・ポイント・リソースを作成します。図28に示すとおりに、大域ゾーンからリソース依存性とリソース・グループ・アフィニティを設定します。リソース・グループMPをオンライン状態にします。
8. Oracle Clusterwareをインストールしてから、Oracle RACをインストールします。
9. スケーラブル・リソース・グループFRAME内にClusterwareフレームワーク・リソースを作成し、図28に示すとおりのリソース依存性を設定します。
10. データベースを作成します。
11. Oracle Clusterware内のファイル・システムに対してプロキシ・リソースを作成します。次に、11g Release 2より前の構成に対して、データベース・インスタンスに対するClusterwareリソースの依存性を更新して、上記プロキシ・リソースをノードごとに追加します。
12. ゾーン・クラスタ内にスケーラブル・リソース・グループDBPを作成します。図28に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACインスタンス・プロキシ・リソースを作成し、図28に示すとおりのリソース依存性を設定します。

リソース・グループDBPをオンライン状態にします。

Oracle Solaris Volume Manager上のQFSを使用したOracle9i RAC

ゾーン・クラスタは、Oracle Solaris Cluster共有ディスク・セット向けのOracle Solaris Volume Manager上のQFS共有ファイル・システムを使用したOracle9i RACをサポートしています。

図29の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。この例では、QFS共有ファイル・システム内にOracle RAC実行可能ファイルも配置されています。

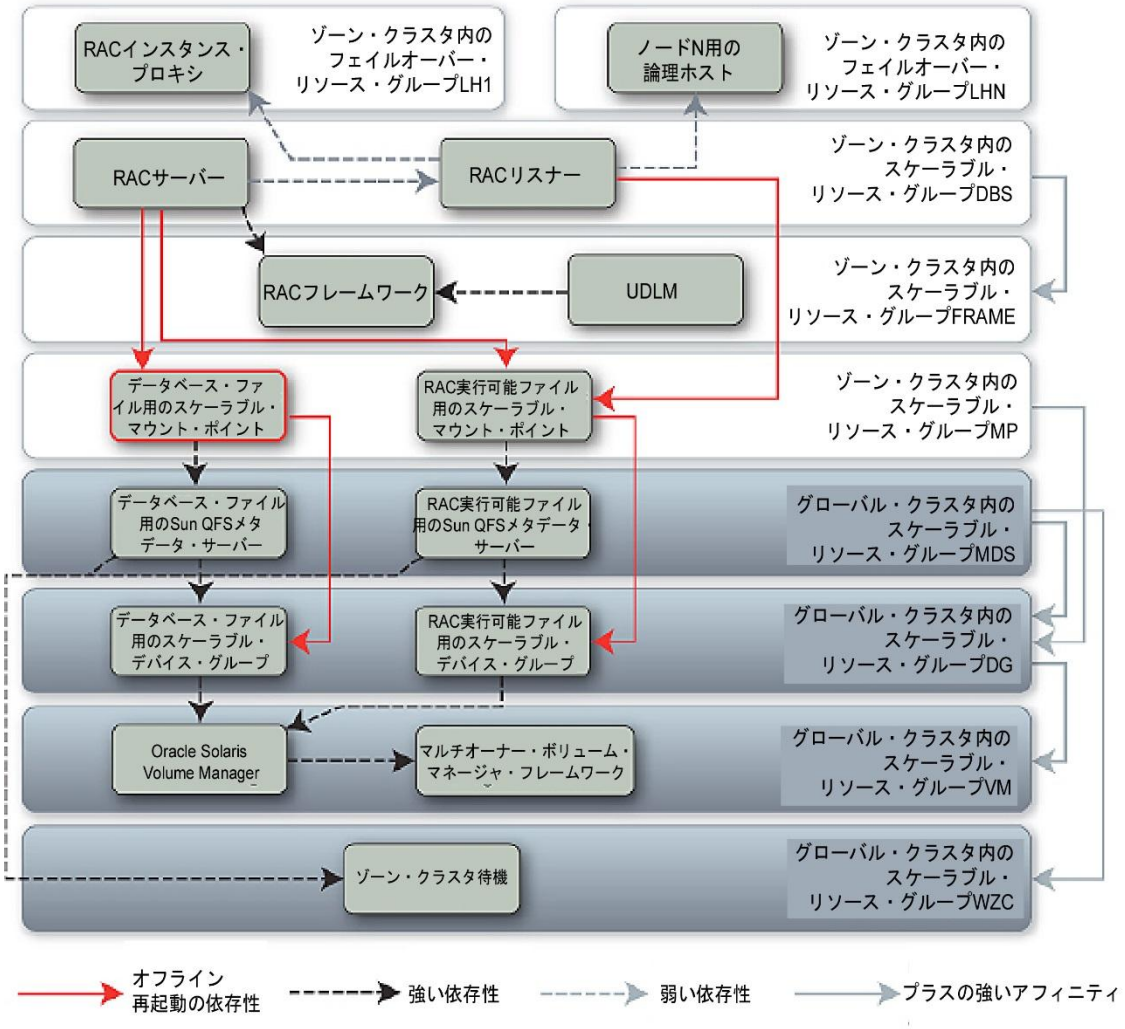

図29. Oracle Solaris Volume Managerを使用した共有QFS上のOracle9i RAC対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図29を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。

2. 大域ゾーン内にスケーラブル・リソース・グループVMを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワーク用とOracle Solaris Volume Manager用のリソースを作成します。VMリソース・グループをオンライン状態にします。
3. 大域ゾーンで構成を実行している場合とまったく同じ方法でOracle Solaris Volume Manager共有ディスク・セットを作成します。図29に示すとおりにこのリソース・グループ内のリソース依存性を設定します。
4. 大域ゾーン内にスケーラブル・リソース・グループDGを作成します。図29に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内でスケーラブル・デバイス・グループ用のリソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図29に示すとおりにこのリソース・グループ内のリソース依存性を指定します。DGリソース・グループをオンライン状態にします。
5. 大域ゾーンで構成を実行している場合とまったく同じ方法でQFS共有ファイル・システムを作成します。ファイル・システムをマウントしないでください。
6. ゾーン・クラスタを作成してQFS共有ファイル・システムと論理ホストのネットワーク・リソースに対するアクセス権を付与します。ゾーン・クラスタをインストールして起動します。
7. ファイル・システムのマウント・ポイントを作成し、QFS共有ファイル・システムをマウントします。
8. 大域ゾーン内にスケーラブル・リソース・グループWZCを作成し、新しく作成したゾーン・クラスタに対してゾーン・クラスタ待機（SUNW.wait_zc_boot）リソースを作成します。WZCリソース・グループをオンライン状態にします。
9. 大域ゾーン内にフェイルオーバー・リソース・グループMDSを作成します。図29に示すとおりにリソース・グループ・アフィニティを設定します。このフェイルオーバー・リソース・グループ内にQFSメタデータ・サーバー・リソースを作成します。図29に示すとおりに、このリソース・グループに対するリソース依存性を設定します。MDSリソース・グループをオンライン状態にします。
10. ゾーン・クラスタに入ります。
11. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
12. ノードごとに1つのフェイルオーバー・リソース・グループ（LH1、LH2・・・）を作成し、そのフェイルオーバー・リソース・グループ内に論理ホスト・リソースを作成します。作成したリソース・グループをすべてオンライン状態にします。
13. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図29に示すとおりのリソース依存性を設定しますリソース・グループFRAMEをオンライン状態にします。
14. この手順ではスケーラブル・リソース・グループMPを取り上げます。システムはスケーラブル・マウント・ポイントのリソース依存性を作成時に検証します。この依存性はクラスタ間の依存性です。クラスタ間依存性はグローバル・クラスタの大域ゾーンからのみ指定できます。

    ゾーン・クラスタ内にスケーラブル・リソース・グループMPを作成します。次に、大域ゾーンに入ります。図29に示すとおりにリソース・グループMPに対してリソース・グループ・アフィニティを設定します。このゾーン・クラスタに属するスケーラブル・リソース・グループMP内に、スケーラブル・マウント・ポイント・リソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図29に示すとおりにこのリソース・グループ内のリソース依存性を指定します。リソース・グループMPをオンライン状態にします。ゾーン・クラスタに戻ります。

15. Oracle RACをインストールします。
16. データベースを作成します。
17. ゾーン・クラスタ内にスケーラブル・リソース・グループDBSを作成します。図29に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACサーバーとOracle RACリスナーのリソースを作成し、図29に示すとおりのリソース依存性を設定します。リソース・グループDBSをオンライン状態にします。

RAID上のQFSを使用したOracle9i RAC

ゾーン・クラスタは、ハードウェアRAID上のQFS共有ファイル・システムを使用したOracle9i RACをサポートしています。

注 – 現時点では、ハードウェアRAIDを表すリソースは存在しません。

図30の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。この例では、QFS共有ファイル・システム内にOracle RAC実行可能ファイルも配置されています。

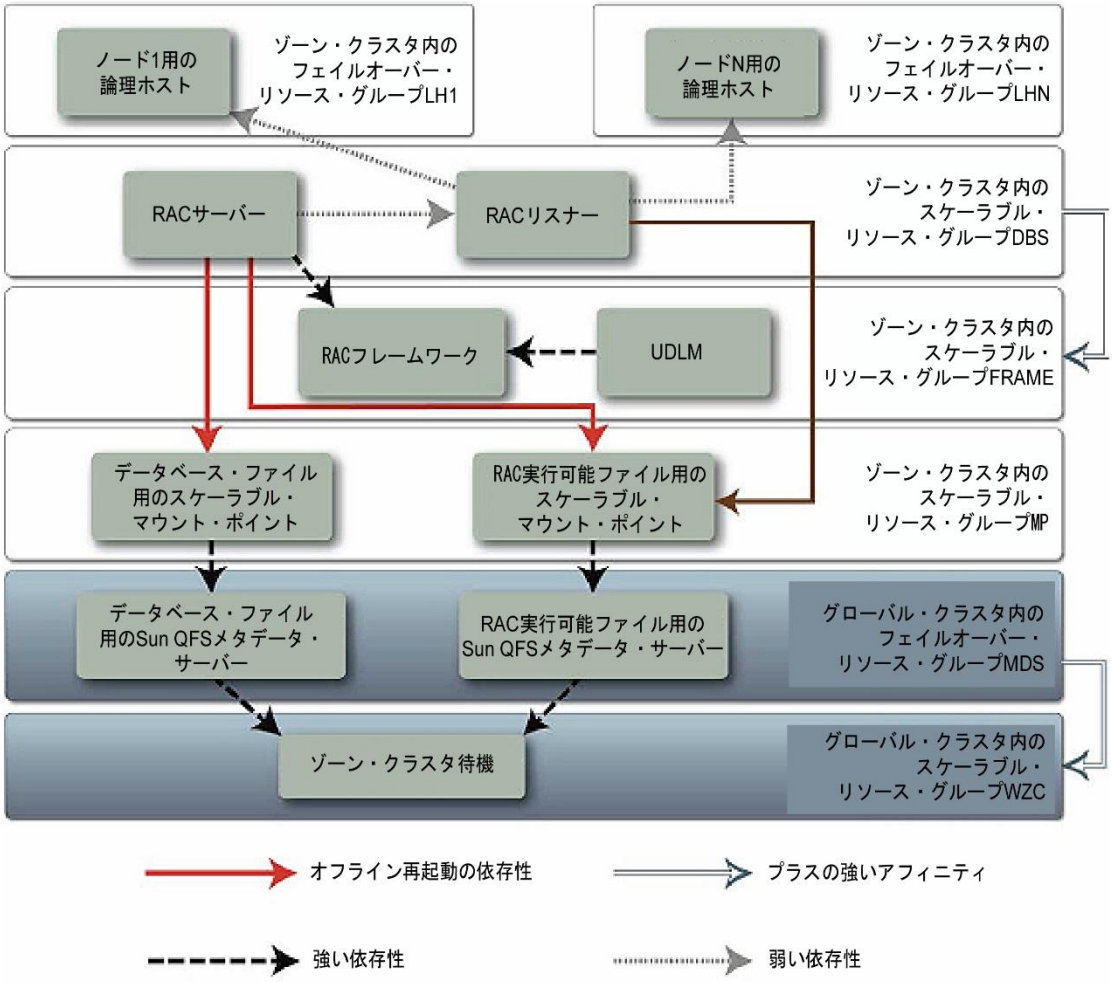

図30. ハードウェアRAIDを使用したQFS共有ファイル・システム上のOracle9i RACに対するリソース関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図30を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。
2. 大域ゾーンで構成を実行している場合とまったく同じ方法でQFS共有ファイル・システムを作成します。ファイル・システムをマウントしないでください。
3. ゾーン・クラスタを作成してQFS共有ファイル・システムに対するアクセス権を付与します。ゾーン・

クラスタをインストールして起動します。

4. ファイル・システムのマウント・ポイントを作成し、QFS共有ファイル・システムをマウントします。
5. 大域ゾーン内にスケーラブル・リソース・グループWZCを作成し、新しく作成したゾーン・クラスタに対してゾーン・クラスタ待機（SUNW.wait_zc_boot）リソースを作成します。WZCリソース・グループをオンライン状態にします。
6. 大域ゾーン内にフェイルオーバー・リソース・グループMDSを作成します。図30に示すとおりにリソース・グループ・アフィニティを設定します。このフェイルオーバー・リソース・グループ内にQFSメタデータ・サーバー・リソースを作成します。図30に示すとおりに、このリソース・グループに対するリソース依存性を設定します。MDSリソース・グループをオンライン状態にします。
7. ゾーン・クラスタに入ります。
8. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
9. ノードごとに1つのフェイルオーバー・リソース・グループ（LH1、LH2・・・）を作成し、そのフェイルオーバー・リソース・グループ内に論理ホスト・リソースを作成します。作成したリソース・グループをすべてオンライン状態にします。
10. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図30に示すとおりのリソース依存性を設定します。リソース・グループFRAMEをオンライン状態にします。
11. この手順ではスケーラブル・リソース・グループMPを取り上げます。システムはスケーラブル・マウント・ポイントのリソース依存性を作成時に検証します。この依存性はクラスタ間の依存性です。クラスタ間依存性はグローバル・クラスタの大域ゾーンからのみ指定できます。
12. ゾーン・クラスタ内にスケーラブル・リソース・グループMPを作成します。次に、大域ゾーンに入ります。図30に示すとおりにリソース・グループMPに対してリソース・グループ・アフィニティを設定します。このゾーン・クラスタに属するスケーラブル・リソース・グループMP内に、スケーラブル・マウント・ポイント・リソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図30に示すとおりにこのリソース・グループ内のリソース依存性を指定します。リソース・グループMPをオンライン状態にします。ゾーン・クラスタに戻ります。
13. Oracle RACをインストールします。
14. データベースを作成します。
15. ゾーン・クラスタ内にスケーラブル・リソース・グループDBSを作成します。図30に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACサーバーとOracle RACリスナーのリソースを作成し、図30に示すとおりのリソース依存性を設定します。リソース・グループDBSをオンライン状態にします。

Oracle Solaris Volume Manager上のOracle9i RAC

ゾーン・クラスタは、Oracle Solaris Cluster向けのOracle Solaris Volume Managerを使用したOracle9i RACをサポートしています。

この構成ではゾーンからディスク・セットに直接アクセスできます。この方法はサポートされていますが、ゾーン内のプログラムからデバイスに対して直接ioctl処理を実行できることを管理者が認識しておく必要があります。

図31の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。

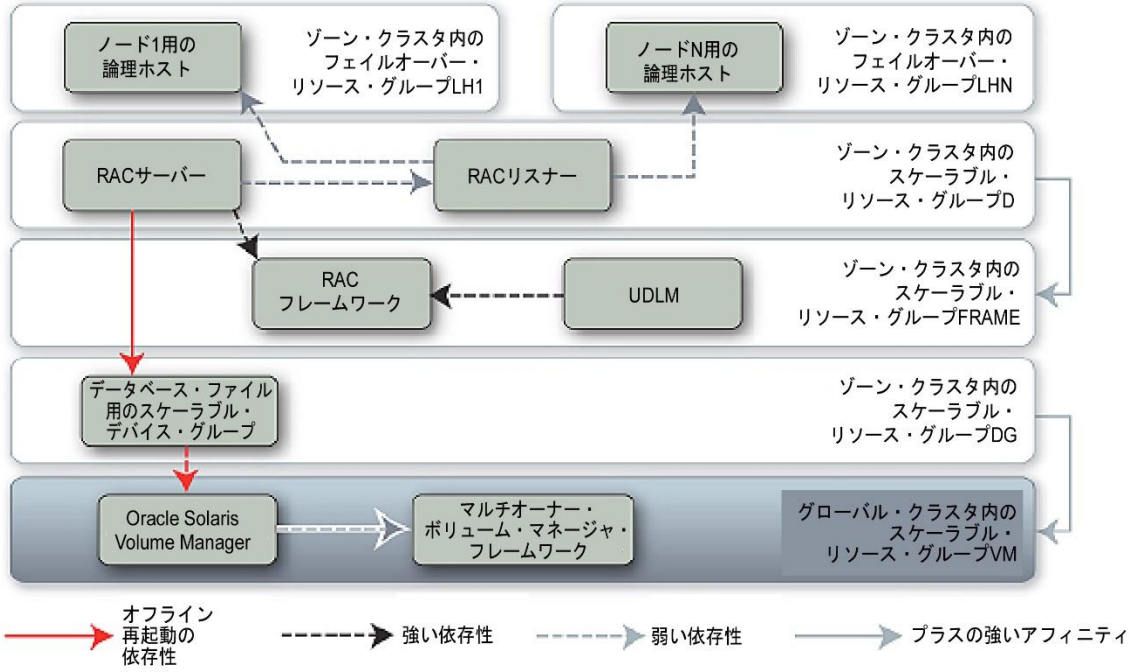

図31.Oracle Solaris Volume ManagerのOracle9i RACに対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図31を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。
2. 大域ゾーン内にスケーラブル・リソース・グループVMを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワーク用とOracle Solaris Volume Manager用のリソースを作成します。VMリソース・グループをオンライン状態にします。
3. 大域ゾーンで構成を実行している場合とまったく同じ方法でOracle Solaris Volume Manager共有ディスク・セットを作成します。
4. ゾーン・クラスタを作成し、Oracle Solaris Volume Managerディスク・セットに対するアクセス権を付与します。ゾーン・クラスタをインストールして起動します。
5. ゾーン・クラスタに入ります。
6. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
7. ノードごとに1つのフェイルオーバー・リソース・グループ（LH1、LH2・・・）を作成し、そのフェイルオーバー・リソース・グループ内に論理ホスト・リソースを作成します。作成したリソース・グループをすべてオンライン状態にします。
8. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMのリソースを作成し、図に示すとおりのリソース依存性を設定します。リソース・グループFRAMEをオンライン状態にします。
9. ゾーン・クラスタ内にスケーラブル・リソース・グループDGを作成します。図31に示すとおりにリソー

ス・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内でスケーラブル・デバイス・グループ用のリソースを作成します。リソースを作成するコマンドで、このリソースに対するリソース依存性も指定します。図31に示すとおりにこのリソース・グループ内のリソース依存性を指定します。DGリソース・グループをオンライン状態にします。

10. Oracle RACをインストールします。
11. データベースを作成します。
12. ゾーン・クラスタ内にスケーラブル・リソース・グループDBSを作成します。図31に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACサーバーとOracle RACリスナーのリソースを作成し、図31に示すとおりのリソース依存性を設定します。リソース・グループDBSをオンライン状態にします。

NAS上のOracle9i RAC

ゾーン・クラスタは、Network Attached Storage（NAS）デバイス上のNFSマウント・ファイル・システムを使用したOracle9i RACをサポートする予定です。NFSマウントはゾーン・クラスタの内部で実行されます。

スケーラブル・マウント・ポイント・リソース（SUNW.ScalMountPoint）はNASデバイス上のNFSファイル・システムのステータスを監視し、ゾーン・クラスタ内に存在します。

図32の構成例では、ゾーンのロケーションとリソース・グループ、リソース、リソース依存性、さらにリソース・グループのマシン・ロケーションを制御するリソース・グループ・アフィニティの関係を示します。

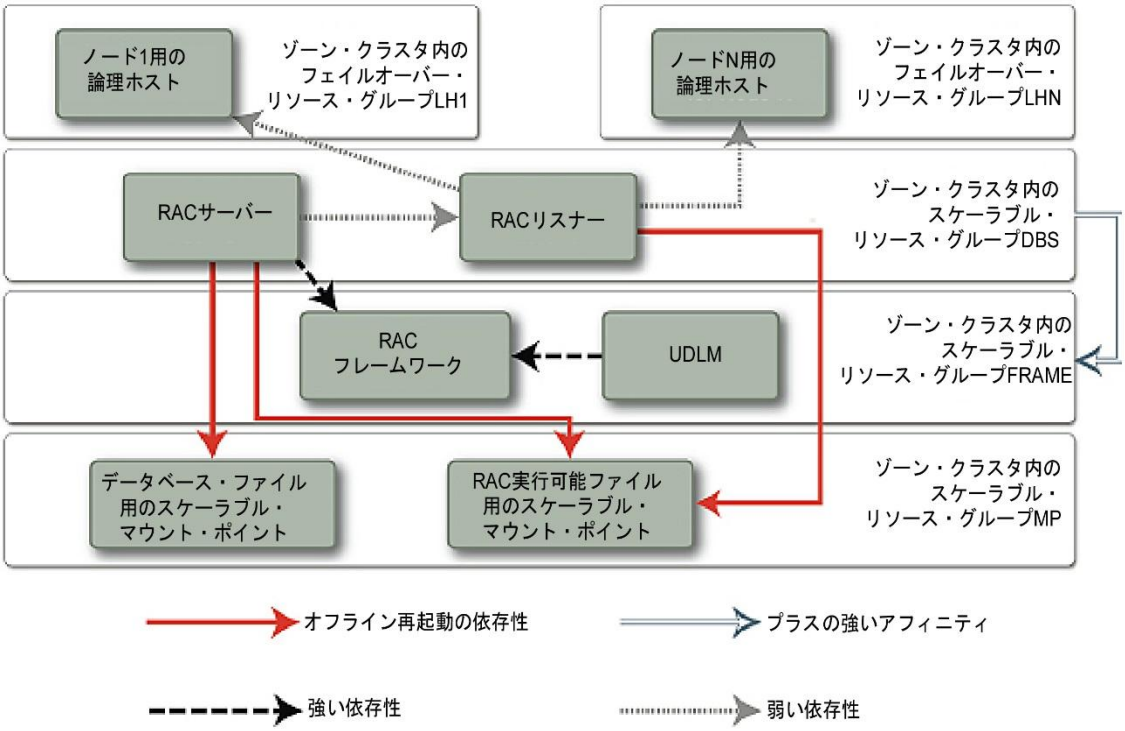

図32. NAS上のOracle9i RACに対するリソースの関係

システム管理者は次の手順に従ってこの構成を作成します。すべての手順で図32を参照してください。

1. まずは大域ゾーンから開始します（ゾーン・クラスタはまだ存在していません）。

2. ゾーン・クラスタを作成します。ゾーン・クラスタをインストールして起動します。
3. ゾーン・クラスタに入ります。
4. NASデバイス上にNFSファイル・システムに対するNFSクライアント・マウントを作成します。
5. ゾーン・クラスタ内にOracle UDLMパッケージをインストールします。
6. ノードごとに1つのフェイルオーバー・リソース・グループ（LH1、LH2・・・）を作成し、そのフェイルオーバー・リソース・グループ内に論理ホスト・リソースを作成します。作成したリソース・グループをすべてオンライン状態にします。
7. ゾーン・クラスタ内にスケーラブル・リソース・グループFRAMEを作成します。このスケーラブル・リソース・グループ内にOracle RACフレームワークとOracle UDLMリソースを作成し、図32に示すとおりのリソース依存性を設定します。リソース・グループFRAMEをオンライン状態にします。
8. ゾーン・クラスタ内にスケーラブル・リソース・グループMPを作成します。このスケーラブル・リソース・グループ内にスケーラブル・マウント・ポイント・リソースを作成し、図32に示すとおりのリソース依存性を設定します。リソース・グループMPをオンライン状態にします。
9. Oracle RACをインストールします。
10. データベースを作成します。
11. ゾーン・クラスタ内にスケーラブル・リソース・グループDBSを作成します。図32に示すとおりにリソース・グループ・アフィニティを設定します。このスケーラブル・リソース・グループ内にOracle RACサーバーとOracle RACリスナーのリソースを作成し、図32に示すとおりのリソース依存性を設定します。リソース・グループDBSをオンライン状態にします。

ハードウェアRAID上のOracle9i RAC、Oracle RAC 10g/11g

ゾーン・クラスタはハードウェアRAID上でOracle RACを実行する機能をサポートしていますが、この構成は一般的ではありません。この構成はファイル・システムまたはOracle ASMのいずれも使用しません。

ハードウェアRAID上のOracle RAC 10g/11g向け構成は図26（"Oracle ASM上のOracle RAC 10g/11g"）と同じです。Oracle9i RAC構成に関する上記セクションで示したように、Oracle9i RAC向けの構成ではOracle RACインスタンス・プロキシを除外して、Oracle RACサーバー、Oracle RACリスナー、論理ホストのリソースを追加します。

単一システム上の複数構成

ゾーン・クラスタのおもなメリットの1つは、物理的に1つのクラスタで複数の構成をサポートできる点です。ここまでに記載した構成の任意の組合せを単一の物理クラスタ上に配置できます。ハードウェア・リソースが十分であれば、サポートされるすべての構成を単一クラスタでサポートできます。当然、同じ種類の構成を使用した複数のゾーン・クラスタを1つの物理クラスタ上に構成できます。

複数ゾーン・クラスタ構成に関する重要な点を説明するために、1つの物理クラスタ上に配置された2つの異なるゾーン・クラスタの例を図33に示します。ゾーン・クラスタAはOracle Solaris Cluster向けのOracle Solaris Volume Manager上のOracle RAC 10gをサポートしており、ゾーン・クラスタBはOracle Solaris Cluster向けのOracle Solaris Volume Manager上のOracle9i RACをサポートしています。この例が示すように、ゾーン・クラスタごとに異なるOracle RACバージョンをサポートできます。

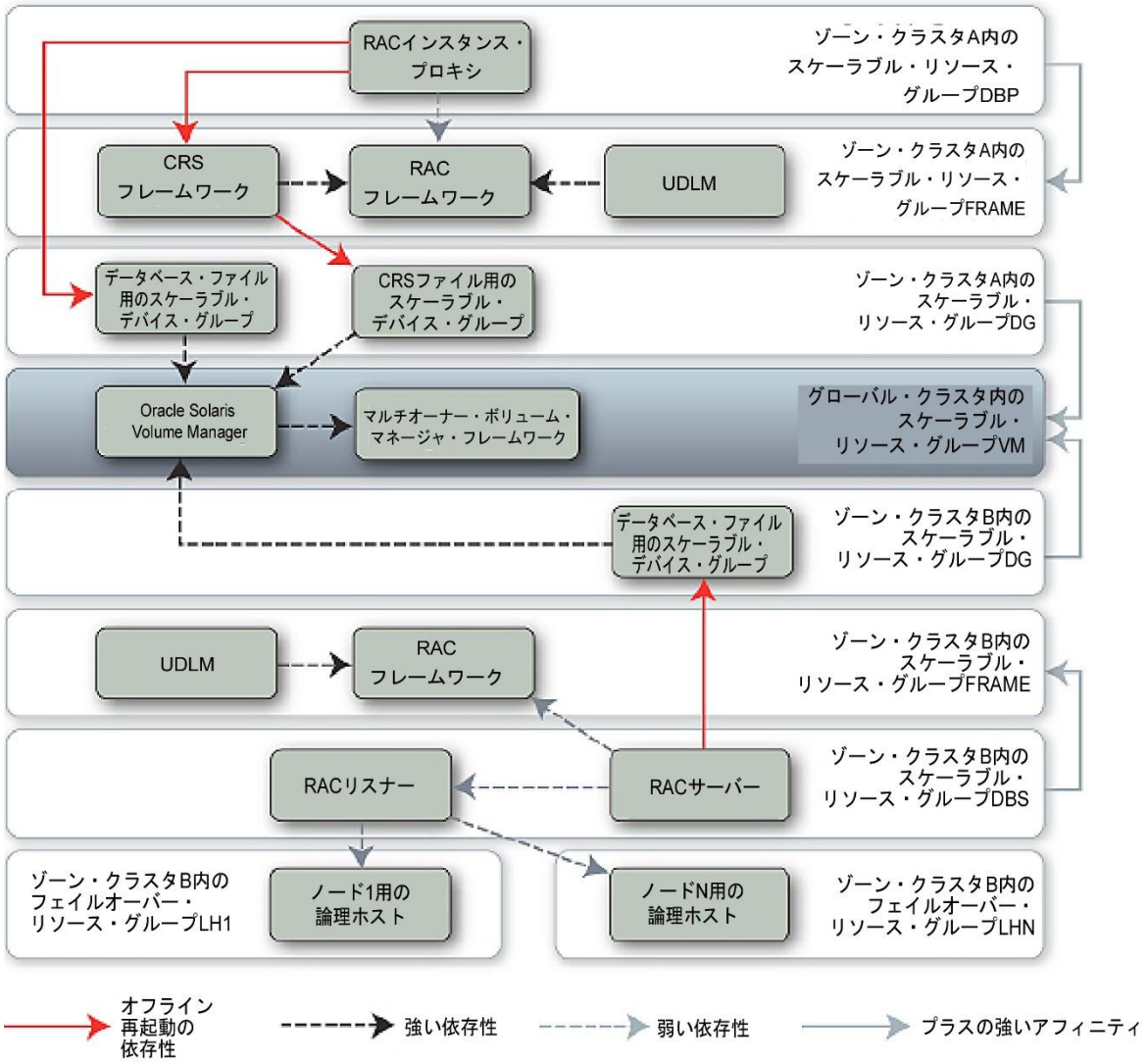

図33.異なるゾーン・クラスタでのOracle Solaris Volume Manager上のOracle RAC 10gとOracle Solaris Volume Manager上のOracle9i RACに対するリソースの関係

特定のゾーン・クラスタ内のアプリケーションとリソースは、大域ゾーンの管理者が依存性またはアフィニティという形で明確に関係を設定しない限り、その他すべてのゾーン・クラスタ内のアプリケーションおよびリソースとは無関係です。異なるゾーン・クラスタ間では名前も独立しています。図33では、ゾーン・クラスタAとゾーン・クラスタBの両方にFRAMEという名前のリソース・グループが含まれています。各リソース・グループが異なる名前空間に配置されているため、名前の競合は発生しません。それぞれのゾーン・クラスタが独自の名前空間を持ちます。

一部の構成には大域ゾーンのサポートが必要です。現時点でボリューム・マネージャは大域ゾーン内にのみ存在するため、ボリューム・マネージャにはこれが当てはまります。1つのゾーン内に配置できるOracle RACフレームワーク・リソースは1つだけであり、この制限は大域ゾーンにも当てはまります。Oracle RACフレームワーク・リソースは、Oracle Solaris Volume ManagerまたはVeritas Volume Managerクラスタ機能（大域ゾーンでのみサポート）のいずれに管理されているかに関係なく、任意の数のボリュームを同時にサポートできます。Oracle RACフレームワーク・リソースはさらに、このゾーン内で実行されている任意のOracle RACソフトウェアを同時にサポートします。このため、既存のゾーン・クラスタ向けのOracle RACフレームワーク・リソースが大域ゾーン内にすでに存在する場合、同じOracle RACフレームワーク・リソースが、同様のサポートを必要とする次のゾーン・クラスタをサポートします。図33は、1つのOracle RACフレームワーク・リソースが複数のゾーン・クラスタをサポートする例を示しています。

著者について

Gia-Khanh NguyenはOracle Solaris Clusterデータベース・ソリューション担当のアーキテクトです。Nguyenはゾーン・クラスタ・プロジェクトでのOracle RACサポートに対する製品要件と設計仕様に貢献しており、各機能の実証を行いました。NguyenはEngineering School for Informatics, Electronics and Automation（フランス）でコンピュータ科学と電気工学の理学修士を取得しています。

Tim ReadはOracle Solaris Clusterグループのソフトウェア開発者です。Readのおもな職務はOracle Solaris Geographic Edition製品の開発です。Readは高可用性とディザスタ・リカバリに関する多数のホワイト・ペーパーと書籍を発表しており、2010年に出版された『Oracle Solaris Cluster Essentials』の著者でもあります。Readはバーミンガム大学（イギリス）で天体物理学の物理学士を取得しています。

当初のホワイト・ペーパーの共著者はEllard Roush博士です。

謝辞

上記機能の開発はプロジェクト・チームによって実施されました。さまざまな方法で貢献してくれたすべてのチーム・メンバーに感謝の意を表します。チーム・メンバーは、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、and Burt Clouseです。

参考資料

表4：参考資料

Webサイト	
Oracle Solaris Cluster	http://www.oracle.com/jp/products/servers-storage/solaris/cluster/overview/index.html http://www.oracle.com/technetwork/jp/server-storage/solaris-cluster/overview/index.html
Oracle Partitioning Rules and Oracle Solaris 10 Containers	http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf
Oracle Database Support for Server Virtualization	http://www.oracle.com/technetwork/database/enterprise-edition/db-virtualization-support-133757.pdf http://www.oracle.com/technetwork/database/clustering/tech-generic-unix-new-166583.html
オラクルの技術記事とホワイト・ペーパー	
ゾーン・クラスタ-仮想クラスタの配備方法と配備する理由	http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/322070_ja.pdf
Oracle Solaris Cluster Essentials	Prentice Hall ISBN-13 978-0-13-248622-4
ドキュメント	
Oracle Solaris Cluster 3.3 Documentation Center	http://download.oracle.com/docs/cd/E19680-01/html/821-1261/fxiwa.html
Oracle Solaris Cluster Concepts Guide for Oracle Solaris OS	http://download.oracle.com/docs/cd/E19680-01/html/821-1254/index.html
Oracle Solaris Clusterシステム管理	http://download.oracle.com/docs/cd/E19680-01/html/821-1257/index.html
System Administration Guide:Oracle Solaris Containers− Resource Management and Oracle Solaris Zones	http://download.oracle.com/docs/cd/E18752_01/html/817-1592/index.html
Oracle Solaris Cluster Software Installation Guide for Oracle Solaris OS, "Configuring a Zone Cluster"	http://download.oracle.com/docs/cd/E19680-01/html/821-1255/ggzen.html
Oracle Solaris Cluster Data Service for Oracle RAC Guide for Oracle Solaris OS	http://download.oracle.com/docs/cd/E19680-01/html/821-1262
clzonecluster(1CL)マニュアル・ページ	http://download.oracle.com/docs/cd/E19680-01/html/821-1263/clzonecluster-1cl.html
sysidcfg(4)マニュアル・ページ	http://download.oracle.com/docs/cd/E19253-01/816-5174/sysidcfg-4/index.html
zoneadm(1M)マニュアル・ページ	http://download.oracle.com/docs/cd/E19253-01/816-5166/zoneadm-1m/index.html
zonecfg(1M)マニュアル・ページ	http://download.oracle.com/docs/cd/E19253-01/816-5166/zonecfg-1m/index.html

59

Oracle Solaris ZonesへのOracle Real Application Clustersのデプロイ
2011年7月

著者：Gia-Khanh Nguyen、Tim Read

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 © 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、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および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。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

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

AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。UNIXはX/Open Company, Ltd.によってライセンス提供された登録商標です。0310

Hardware and Software, Engineered to Work Together