Oracle Database Applianceによる Oracle Data Guardのデプロイメント

Oracle Database Applianceベア・メタルおよび
仮想化プラットフォーム構成への適用

Oracle ホワイト・ペーパー | 2016 年 1 月

ORACLE

はじめに

Oracle Database Appliance は事前に構築、チューニングが行われ、すぐに使用できる、クラスタ化されたデータベース・システムです。構成が最適化されたサーバー、ストレージ、ネットワーク、ソフトウェアが含まれているため、デプロイメント、操作、管理を容易に行えます。Oracle Database Appliance はあらゆる規模の実装に理想的なデータベース・プラットフォームであり、世界トップの Oracle Database、人気の Oracle Real Application Clusters（Oracle RAC）データベース・オプション、Oracle Clusterware、Oracle Automatic Storage Management などの強力で実績のある Oracle テクノロジーが組み込まれています。Oracle Database Appliance ではハードウェアとソフトウェアが統合されているため、手作業で組み立てる非統合のソリューションに伴う複雑さを解消し、数週間から数か月かかっていたデプロイメントを数時間に短縮しています。また、構成や設定の誤りにより、最適とは言えない、管理が困難なデータベース環境となることを防ぎます。

スタンバイ・データベース環境が必要な理由

Oracle Database Appliance はそれ自体が可用性の高いシステムですが、スタンバイ・データベース環境を配置すると、計画停止時間および計画外停止時間に対する保護と同時に、プライマリ・データベース環境が利用できなくなった場合のデータ損失に対する保護を提供できます。また、適切なテクノロジーを使用して、スタンバイ・データベースとプライマリ・データベースの同期を維持できるため、ユーザー・エラーからシステム障害、そして災害に至るまでの問題に対するシステム・パッチの適用などの計画的なメンテナンスの場合でも、データベースの運用をほぼ透過的に継続できます。このように、スタンバイ・データベースは、重要な本番システムの高可用性と保護を確保する上で常に鍵となってきました。

オラクルでは、プライマリの Oracle Database Appliance システムで動作するミッション・クリティカルな本番システムに対応するスタンバイ・システムを、独立した専用の Oracle Database Appliance システムを使用してホストすることを推奨しています。

図1：2つのOracle Database Applianceを使用したOracle Data Guardの設定

Oracle Data Guardを使用する理由

Oracle Data Guard がディザスタ・リカバリ・ソリューションとして推奨される理由は、Oracle Database Appliance 上に配置されたデータベースを、データベースの"異常"または停止につながるデータベースまたはクラスタの障害、データ破損、災害、ユーザー・エラーから保護できるためです。Oracle Data Guard と Oracle Database の緊密な統合は、他のソリューションでは達成できない、これまでにないレベルのデータ保護を実現します。Oracle Data Guard は、Oracle Database Enterprise Edition の一部として提供されています。Oracle Data Guard はデプロイメントが簡単で、1 つ以上の同期コピー（スタンバイ・データベース）を作成して維持するための管理、監視、および自動化を目的としたソフトウェアを提供します。Oracle Data Guard を使用すると、何らかの理由で本番データベース・システムが使用できなくなった場合に、データベースの可用性を容易に維持できます。また、アプリケーションのワークロードをスタンバイ環境に移行し、計画的な保守作業で生じる停止時間を最

小限に抑えることができます。

Oracle Data Guardの使用によるさまざまな利点

Oracle Data Guard を使用すれば、スタンバイ・データベース環境がアイドル状態の無駄な容量となることがありません。むしろ、スタンバイ・データベースは多くの有益な目的に非常に役立ちます。多くの目的に使用できるため、労力と投資の見返りとして、全体的に大きな利点が得られます。

Oracle Database Appliance への移行：既存のデータベースを Oracle Database Appliance に移行する場合、Oracle Data Guard を使用すれば、データベースを Oracle Database Appliance に容易に移行できます。Oracle Database Appliance でフィジカル・スタンバイ・データベースを設定するだけで、古い環境から新しい環境に運用をスイッチオーバーできます。これには、特定のプラットフォーム間での移行も含まれます。たとえば、現在 Windows で稼働しているデータベースを Linux プラットフォームの Oracle Database Appliance に移行するには、2 つの環境の間に Oracle Data Guard を設定してスイッチオーバーを実行します。このプラットフォーム移行のアプローチでは、テスト後に何らかの理由でスイッチバックが必要となった場合に、柔軟にスイッチバックを行えます。Oracle Data Guard を使用したプラットフォームの移行の詳細については、My Oracle Support（MOS）Note 413484.1、『Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration』を参照してください。

注：Oracle Data Guard では、一時ロジカル・スタンバイ・データベースを使用してバージョンの異なるデータベース間で移行を行うこともできます。

ディザスタ・リカバリ：Oracle Data Guard のフィジカル・スタンバイ・データベースは、災害対策として理想的なソリューションです。災害のシナリオは、水道管や排気管の破裂、火災、ハリケーン、破壊行為、地震、洪水、テロ行為に至るまでさまざまです。Oracle Data Guard のフィジカル・スタンバイ・データベースは、本番データベースのコピーをブロック単位で保持します。何らかの理由でプライマリ環境が利用できなくなった場合は、スタンバイ環境をすばやくアクティブ化し、アプリケーションに対してデータベースの可用性を引き続き維持できます。

高可用性：スタンバイ・データベースは、計画停止時間や計画外停止時間中の可用性を維持する上でも役立ちます。このようなイベントには、構成の変更、ハードウェアの交換などの他、データの破損、人的エラーによる障害、その他の予期しないシステム・コンポーネントの障害やシステム全体の障害などが含まれます。

データベース・ローリング・アップグレード：スタンバイ・データベースは、プライマリ Oracle Database Appliance に対してパッチ・バンドルを適用する場合や変更を加える場合の停止時間を最小限に抑えます。パッチやその他の保守作業は、まずスタンバイ・データベースに適用して検証を行ってから、本番のワークロードをプライマリからスタンバイ・システムに切り替えます。データベースの停止時間は、プライマリとスタンバイの間でロールを切り替えるのに要する短い時間のみです。詳細については、My Oracle Support（MOS）Note 1265700.1、『Oracle Patch Assurance - Data Guard Standby-First Patch Apply』を参照してください。

ワークロードとアクティビティのオフロード：スタンバイという名前が付いてはいますが、スタンバイ環境をアイドル状態にしておく必要はありません。全体的な投資収益率を最大限に高めるよう、積極的に利用できます。フィジカル・スタンバイ・データベースを配置すると、いくつかの重要なアクティビティをスタンバイ環境にオフロードできます。次にその例を挙げます。

- **読取り専用ワークロード：**Oracle Active Data Guard オプションを使用すると、スタンバイ・モードであっても、スタンバイ・データベースは読取り専用の問合せワークロードを受け付けることができ、プライマリ・データベースからの REDO ログの更新を受け付けることができます。多くの場合、読取り専用ワークロードをスタンバイ・データベースにオフロードすると本番ワークロードを大幅に削減できるため、本番システムの使用可能な容量が全体的に増加します。
- **バックアップ：**Oracle Data Guard のフィジカル・スタンバイ・データベースにプライマリ・データベースのコピーがブロック単位で保持されるため、データベースのバックアップをスタンバイ環境に完全にオフロードできます。また、障害が発生した場合やデータベースが失われた場合に、これらのバックアップを透過的に使用して、プライマリ・データベースのリストアとリカバリを行うことができます。Oracle Active Data Guard オプションのライセンスがある場合は、スタンバイ・データベースで高速増分バックアップを実行できるため、スタンバイ・データベースへのバックアップ・オフロードがさらに魅力的なものとなります。
- **ブロック修復：**フィジカル・スタンバイ・データベースのその他の利点として、ブロックが破損した場合に自動リカバリが行える点が挙げられます。プライマリ/スタンバイ構成では、破損したブロックを自動的に修復でき、この操作をエンドユーザーおよびデータベース管理者に対して完全に透過的に実行できます。ブロック修復機能も Oracle Active Data Guard オプションの一部です。

スナップショット・スタンバイ：スナップショット・スタンバイ・データベースは、プライマリ・データベースの完全なデータ保護を実現する更新可能なスタンバイ・データベースです。このデータベースは、プライマリからの REDO データを継続的に受信しますが、スタンバイ・データベースがテスト用の読取り/書込み操作を受け付けている間は、適用プロセスが停止します。テストが完了すると、1 つのコマンドで、スタンバイ・データベースを元の状態に戻し、読取り/書込みモードで受け付けている間に行われた変更を破棄し、蓄積された REDO ログを適用してプライマリ・データベースの現在の状態と同期させます。

設定のベスト・プラクティス

このセクションでは、Oracle Database Appliance 上で Oracle Data Guard を設定する際の重要なベスト・プラクティスについていくつか説明します。Oracle Data Guard の一般的なベスト・プラクティス（Oracle Database Appliance 環境にも適用可能）の一覧については、『Oracle® Database High Availability Best Practices Guide 11g Release 2 (11.2)』の第 8 章「Configuring Oracle Data Guard」を参照してください。

- **プライマリ・データベースとスタンバイ・データベースの構成を一致させる：**一貫したサービス・レベルを維持し、プライマリ・データベースとスタンバイ・データベースを透過的に使用するには、プライマリ・システムとスタンバイ・システムのリソース、設定、構成をできるだけ一致させることが重要です。プライマリ・データベースとスタンバイ・データベースの構成が大きく異なっていると、ロール移行の発生時に最善のパフォーマンスが得られなかったり、想定外の動作が発生したりすることがあります。特に、以下の推奨事項を検討する必要があります。
    - ***プライマリ・データベースとスタンバイ・データベースを別々の Oracle Database Appliance***

上で実行：プライマリ・データベースとスタンバイ・データベースは、できれば地理的に離れた場所に設置された、別々の Oracle Database Appliance 装置上で実行させることを推奨します。

- ***プライマリ・データベースとスタンバイ・データベースを同じ構成で実行：***Oracle Database Appliance では、Oracle RAC データベース、Oracle RAC One、および Single-Instance Enterprise Edition データベースの 3 つのタイプのデータベース構成がサポートされています。また、スタンバイ・データベースはプライマリ・データベースと同じタイプの構成にする必要があります。したがって、プライマリ・データベースを Oracle RAC データベースとして構成する場合は、スタンバイ・データベースも Oracle RAC データベースとして構成する必要があります。
- ***プライマリ・インスタンスとスタンバイ・インスタンスで互いに類似した構成を使用：***プライマリ・データベースとスタンバイ・データベースのインスタンスは、メモリ、CPU、ネットワーク、ストレージなどのデータベース・パラメータ設定について、相互に類似した構成にする必要があります。これにより、データベースのロールの切替え時に予測不可能な動作の発生を防止するのに役立ちます。また、オペレーティング・システムの構成のカスタマイズを 2 つの環境でミラー化する必要があります。
- ***ロール移行に備えてプライマリ・データベースとスタンバイ・データベースを事前に構成：***プライマリ・データベースとスタンバイ・データベースは、プライマリからスタンバイ、またスタンバイからプライマリへのロール移行中に、要求される変更や必要な変更が最小限で済むように構成する必要があります。そのため、プライマリ・データベースに実装されたすべてのデータベース機能とカスタマイズは、事前にスタンバイ・データベースでも構成しておく必要があります。

- **プライマリ・データベースとスタンバイ・データベースの両方でフラッシュバック・データベースを構成：**フラッシュバック・データベース機能は、ロール移行をすばやく行い、移行後にデータベースのロールを再設定するために必要な作業を軽減します。ベスト・プラクティスとして、プライマリ・データベースとスタンバイ・データベースの両方で、フラッシュバック・データベースを構成する必要があります。

  注：フラッシュバックを Data Guard 構成を再指定する目的でのみ使用する場合のベスト・プラクティスは、フラッシュバックの保存ターゲットをデフォルトの 24 時間から 2 時間に短縮することです。

- **スタンバイ・トラフィックに専用ネットワークを使用：**Oracle Database Appliance は複数の冗長ネットワーク・インタフェースが事前に組み込まれた状態で出荷されます。必要に応じて、スタンバイ・データベースのトラフィック用に独立したネットワーク・パスを構成して、ユーザーやアプリケーションに関連するワークロードのパフォーマンス低下を最小限に抑えることができます。Oracle Data Guard で必要なのは、プライマリ・データベースに対して行われた変更のみをプライマリ・データベースからスタンバイ・データベースに転送することなので、ネットワークに必要以上の要件が課されることはありません。そのため、多くの場合、Oracle Data Guard のデプロイメントでは、プライマリとスタンバイ間での REDO ログの転送に独立したネットワーク・パスを確立する必要はありません。ただし、一部の大容量アプリケーションを使用する場合や、組織のベスト・プラクティスや基準によっては、REDO ログの転送用に独立したネットワーク・パスが必要になることもあります。このために Oracle Database Appliance では、各サーバー・ノードに対して追加のネットワーク・インタフェースを提供しています。Oracle Database Appliance でのディザスタ・リカバリ用の専用ネットワークの構成についての詳細は、MOS Note 1422563.1 を参照してください。

- **特定のワークロードをスタンバイ・データベースにオフロードすることを検討：**Oracle Recovery

Manager（Oracle RMAN）はプライマリ・データベースとスタンバイ・データベースの間で透過的に機能します。そのため、フィジカル・スタンバイ・データベースで作成されたデータベース・バックアップを本番データベースのリカバリに使用できます。スタンバイ・データベースは、プライマリ・データベース環境からのバックアップのオフロードに利用する必要があります。Oracle Active Data Guard 構成を使用すると、問合せのワークロードをスタンバイ環境にオフロードできます。また、フィジカル・スタンバイ・データベースにより、ブロックの破損の修復を透過的に行えるようになります。

- **Oracle Active Data Guard の利用を検討：**Oracle Active Data Guard は、管理リカバリ（プライマリ・データベースからの REDO 転送とスタンバイ・データベースへの適用）を同時にアクティブにしている間、スタンバイ・データベースが読取り専用操作を受け付けることを可能にします。これによってプライマリ環境のワークロードをスタンバイ・データベースに分散させ、スタンバイ・データベースの投資対効果を高めることができます。Oracle Active Data Guard により、スタンバイ・データベースで高速増分バックアップを使用することもできます。高速増分バックアップを使用すると、バックアップ時間が数時間から数分に短縮される場合もあります。

Oracle Database の Oracle Maximum Availability Architecture（Oracle MAA）ベスト・プラクティスを確認：Data Guard 環境のデプロイメントと使用状況によって異なりますが、Oracle Data Guard 用の以下のベスト・プラクティスが役立つ場合があります。

Data Guard 12c でのクライアント・フェイルオーバーのベスト・プラクティス

http://www.oracle.com/technetwork/jp/database/availability/client-failover-2280805-ja.pdf

Active Data Guard 12c での REDO 転送構成のベスト・プラクティス

http://www.oracle.com/technetwork/jp/database/availability/broker-12c-transport-config-2082184-ja.pdf

ロール移行のベスト・プラクティス

http://www.oracle.com/technetwork/jp/database/availability/maa-roletransitionbp-2621582-ja.pdf

ブロック破損の防止、検出、および修復 - Oracle Database 12c

http://www.oracle.com/technetwork/jp/database/availability/corruption-bestpractices-12c-2141348-ja.pdf

Oracle Database 12c の Maximum Availability Architecture（MAA）のベスト・プラクティスは以下で入手できます。

http://www.oracle.com/technetwork/jp/database/features/availability/maa-098142-ja.html

Oracle Database Applianceベア・メタルおよび仮想化プラットフォーム構成

Oracle Database Appliance はベア・メタル（非仮想化）プラットフォームまたは仮想化プラットフォームとして構成できます。このホワイト・ペーパーに記載した Oracle Data Guard のフィジカル・スタンバイ設定プロセスは両方の Oracle Database Appliance 構成で使用できます。Oracle Database Appliance の仮想化プラットフォームでは、構成手順は ODA_BASE ドメイン内で実行されます。また、Oracle Database Appliance の仮想化プラットフォームで仮想 LAN を使用して、ディザスタ・リカバリ用に論理的に独立したネットワークを構成できます。

結論

Oracle Data Guard を使用すると、Oracle Database Appliance でのデプロイメント初期段階から効果的なディザスタ・リカバリ保護計画をデプロイできます。このホワイト・ペーパーで説明したフィジカル・スタンバイの構成および設定プロセスはシンプルで時間がかからず、プライマリ・データベースの停止時間を発生させることなく完了できます。スタンバイの作成手順の大部分は、Oracle Appliance Manager、Oracle Database Configuration Assistance（Oracle DBCA）、Oracle RMAN、Oracle Data Guard などのツールを使用することで自動化され、Oracle Database Appliance でスタンバイ・データベースを設定する際に使用できます。このホワイト・ペーパーの付録 A には、参考のために完全なプロセスを示しています。

付録A：Oracle Database Applianceでの設定例

サンプル環境

このセクションでは、後述する Oracle Database Appliance を使用した Data Guard の設定例で使用される、プライマリ・データベース環境およびスタンバイ・データベース環境のトポロジについて説明します。

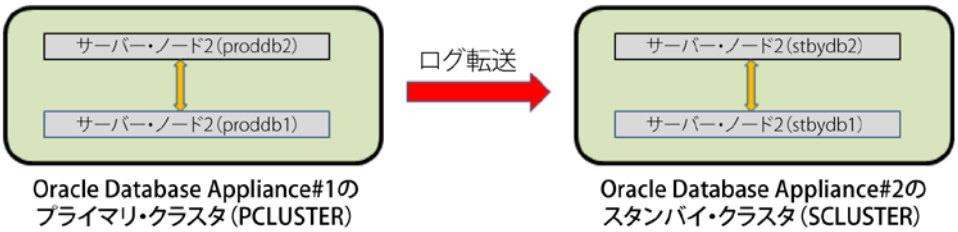

図 2：Oracle Database Appliance での Oracle RAC の構成トポロジ

	プライマリ Oracle Database Appliance		スタンバイ Oracle Database Appliance	
アプライアンス名	appliance#1		appliance#2	
ホスト名	pnode1	pnode2	snode1	snode2
クラスタ名	PCLUSTER		SCLUSTER	
データベース名	proddb		stbydb	
インスタンス名	proddb1	proddb2	stbydb1	stbydb2
SCAN 名と IP	primary-scan（10.1.27.2、10.1.27.3）		standby-scan（10.1.27.4、10.1.27.5）	
Oracle Grid Infrastructure ソフトウェアのインストール場所	/u01/app/12.1.0.2/grid		/u01/app/12.1.0.2/grid	
Oracle Database ソフトウェアのインストール場所	/u01/app/oracle/product/12.1.0.2/db_home1		/u01/app/oracle/product/12.1.0.2/db_home1	
ARCHIVELOG モード	有効		有効	
FORCE LOGGING モード	有効		有効	

表 1：Oracle Database のネーミング規則の例

プライマリ環境の構成

1. スタンバイREDOログの作成

スタンバイ REDO ログには、プライマリ・データベースから受け取った REDO データが記録されます。オラクルは、プライマリ・スタンバイの設定前に、プライマリ・データベース上にもスタンバイ REDO ログを作成しておくことを推奨しています。これによって、スタンバイ・ロールにスイッチオーバーした後すぐに REDO データを受け取れる状態になります。

プライマリ・データベース上にスタンバイ REDO ログ（SRL）を作成します。スタンバイ REDO ログの各スレッドには、オンライン REDO ログの対応するスレッドよりも少なくとも 1 つ多い REDO ログ・グループを割り当てる必要があります。次に例を示します。

```
$ sqlplus / as sysdba
SQL> alter database add standby logfile thread 1 group 9 size 1024M, group 10 size 1024M, group 11 size 1024M, group 12 size 1024M, group 13 size 1024M;
SQL> alter database add standby logfile thread 2 group 14 size 1024M, group 15 size 1024M, group 16 size 1024M, group 17 size 1024M, group 18 size 1024M;
```

オンライン REDO ログの数とそのサイズを確認するには、次の問合せを使用します。

```
SQL> select group#, thread#, bytes from v$log;
```

スタンバイ REDO ログのサイズが REDO ログのサイズと一致している必要があることに注意してください。スタンバイ REDO ログは、ソリッド・ステート・ディスク上にある REDO ディスク・グループに作成する必要があります。

スタンバイ REDO ログ内の各ログ・ファイルのサイズとログ・グループの数を確認するには、次の問合せを使用します。

```
SQL> select group#, thread#, bytes from v$standby_log;
```

2. プライマリ・データベースでのARCHIVELOGモードの有効化

アーカイブとは、アクティブ・データベースの REDO ログが循環式に上書きされる前に、アーカイブ・ファイル形式で REDO 情報を保存し、保護するプロセスのことです。Oracle Database Appliance で作成されたデータベースは、デフォルトでアーカイブがオンになっています。ただし、データベースを ARCHIVELOG モードで実行することは必須ではありません。

プライマリ・データベースが ARCHIVELOG モードで実行されていることを確認します。

```
SQL> archive log list
```

プライマリ/データベースが ARCHIVELOG モードで実行されていない場合は、次のように ARCHIVELOG モードを有効にします。Oracle Database Appliance で両方のインスタンスを停止します。

```
$ srvctl stop database –d proddb
```

1 つのインスタンスを排他モードで起動し、マウントします。

```
SQL> startup mount exclusive;
```

アーカイブをオンにします。

```
SQL> alter database archivelog;
```

インスタンスを停止します。

SQL> shutdown immediate;

データベースを再起動します。

$ srvctl start database –d proddb

3. **FORCE LOGGINGモードの有効化**

FORCE LOGGING を使用すると、NOLOGGING 属性で実行されたデータベース操作を取得できます。これにより、スタンバイ・データベースの整合性が確保されます。プライマリ・データベースで FORCE LOGGING が有効化されていることを確認します。

SQL> select force_logging from v$database;

FORCE LOGGING が有効化されていない場合は、次のコマンドで有効化します。

SQL> alter database force logging;

4. **フラッシュバック・データベース機能の構成**

Oracle フラッシュバック・データベース機能は、未完了のデータベース・リカバリをすばやく行うための代替方法を提供します。フラッシュバック・データベース機能の使用は任意ですが、フェイルオーバーの後で、古いプライマリ・データベースの復元をすばやく行うためには非常に有益です。そのため、スタンバイ・データベースへのフェイルオーバーを実行すると、古いプライマリ・データベースの修復が可能となり、古いプライマリ・データベースをスタンバイ・データベースとして再構築することなく、フラッシュバックするだけで、それ以降 Oracle Data Guard が再同期するようになります。

プライマリ・データベースでフラッシュバック・データベース機能が有効化されているかどうかを確認し、必要に応じて有効化します。

SQL> select flashback_on from v$database;
SQL> alter database flashback on;

フラッシュバック・データベースを有効化すると、ファスト・リカバリ領域（RECO ディスク・グループ）の消費量が増加することに注意してください。フラッシュバック・ログによって使用される領域を制御するには、パラメータ DB_FLASHBACK_RETENTION_TARGET に望ましい値を設定します。この値は分単位です。次に例を示します。

SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120 scope=both sid='*';

5. **スタンバイ・ファイル管理の有効化**

プライマリ・データベースでデータファイルの追加または切離しが行われた場合、スタンバイ・データベースでもそれに対応するアクションが自動的に実行される必要があります。この操作を有効化するには、自動スタンバイ・ファイル管理を使用します。

SQL> alter system set STANDBY_FILE_MANAGEMENT=AUTO scope=both sid='*';

6. リモート特権ログインの有効化

プライマリ・データベースの各インスタンスがリモート・ログイン・パスワード・ファイルを用いて構成されていることを確認します。Oracle Database Appliance では、この設定を使用してデータベースがデプロイされることに注意してください。初期化パラメータ REMOTE_LOGIN_PASSWORDFILE は、exclusive に設定する必要があります。実際の環境でこのパラメータがリセットされており、次のように変更する必要がある場合は、変更を有効にするためにデータベースを再起動する必要があります。

```
$ sqlplus / as sysdba
SQL> show parameter remote_login_passwordfile
SQL> alter system set remote_login_passwordfile='exclusive' scope=spfile sid='*';
```

7. TNSエントリの設定

Oracle Net サービス名は、データベース間で REDO 転送ができるように構成する必要があります。プライマリ・データベースとスタンバイ・データベースの両方について、TNS 別名を含めるように tnsnames.ora ファイルを更新します。Oracle Database Appliance では、tnsnames.ora ファイルは Oracle Database ホームの network/admin ディレクトリにあります。

```
PRODDB =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST = primary-scan) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = proddb)))
 STBYDB =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST = stanby-scan) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = stbydb)))
```

8. REDO転送サービスの設定

Oracle Data Guard の REDO 転送メカニズムでは、Oracle Net 接続を利用してデータベース間で REDO が送信されます。REDO 転送を有効化するには、LOG_ARCHIVE_DEST_n パラメータを設定します。たとえば、次の設定により、ログ送信が有効化され、LGWR ベースの転送が非同期モードで使用されます。

```
SQL> alter system set log_archive_dest_2='SERVICE=stbydb LGWR ASYNC REGISTER
VALID_FOR=(online_logfile,primary_role)  REOPEN=60  DB_UNIQUE_NAME=stbydb'
scope=both sid='*';
```

REDO ログ転送オプションについての詳細については、『Oracle Data Guard Concepts and Administration』ガイドを参照してください。

9. フェッチ・アーカイブ・ログ・サーバーの設定

データベースがスタンバイ・ロールになっており、欠落したログ・ファイルをプライマリ・データベースが送信できない場合、スタンバイ・データベースは FAL_SERVER 設定を使用してこれらの欠落したログ・ファイルを取得できます。FAL_SERVER パラメータでは Oracle Net サービス名が使用されます。

```
SQL> alter system set FAL_SERVER=stbydb scope=both sid='*';
```

スタンバイ環境の構成

このセクションでは、スタンバイ・データベースで実行する必要のある手順について説明します。スタンバイ環境に Oracle Database Appliance システムが設定されていることを前提としています。ベア・メタルまたは仮想化プラットフォーム構成での Oracle Database Appliance の設定については、『Oracle Database Appliance Setup Poster』を参照してください。URL は次のとおりです。
http://docs.oracle.com/cd/E22693_01/doc.12/e55694.pdf

10. TNSエントリの設定

Oracle Net サービス名は、データベース間で REDO 転送ができるように構成する必要があります。プライマリ・データベースとスタンバイ・データベースの両方について、TNS 別名を含めるように tnsnames.ora ファイルを更新します。Oracle Database Appliance では、tnsnames.ora ファイルは Oracle Database ホームの network/admin ディレクトリにあります。

```
PRODDB =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST = primary-scan) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = proddb)))

STBYDB = (DESCRIPTION = (UT=A)
(ADDRESS = (PROTOCOL = TCP) (HOST = stanby-scan) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = stbydb)))
```

11. 静的リスナー構成の作成

インスタンス化実行中に grid ユーザーとして、スタンバイ・データベース上に Oracle Recovery Manager（Oracle RMAN）接続用の静的リスナー・サービスを作成します。リスナー・ホームは、Grid Infrastructure のホーム（/u01/app/12.1.0.2/grid/network/admin）にあることに注意してください。

```
SID_LIST_LISTENER =
  (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = stbydb)
  (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)(SID_NAME = stbydb)))
```

12. リスナーの再起動

リスナーに変更を加えたら、リスナーを再起動する必要があります。

```
grid $> lsnrctl reload listener
```

13. 初期スタンバイ・パラメータ・ファイルの作成

スタンバイ・データベースのパラメータ・ファイル（$ORACLE_HOME/dbs/initstbydb.ora）を作成します。ORACLE_HOME の環境変数が正しいホームを指していることを確認します。次に例を示します。

```
grid $> echo 'DB_NAME=stbydb' > $ORACLE_HOME/dbs/initstbydb.ora
```

14. パスワード・ファイルの作成

Oracle RMAN の複製プロセス中に、リモート認証で補助インスタンスにアクセスする必要があり、この認証のパスワード・ファイルを作成する必要があります。

```
oracle $> orapwd file=/u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/orapwstbydb
password=<primary sysdba passwd>
```

15. 監査ディレクトリの作成

スタンバイ・データベースに監査ファイルの保存先ディレクトリを作成します。

```
$ mkdir –p /u01/app/oracle/admin/stbydb/adump
```

16. スタンバイOracle Database ApplianceシステムでのACFSボリュームの作成（任意）

データベース環境を作成せずにスタンバイ・データベース・アプライアンスをデプロイする場合、"oakcli create dbstorage"コマンドを使用してデータベース・ストレージを各自で作成できます。詳細については、以下の Web サイトを参照してください。

http://docs.oracle.com/cd/E22693_01/doc.12/e55580/oakcli.htm#CHDBHHDG

17. スタンバイ・インスタンスの起動

NOMOUNT 状態である最初のスタンバイ・ホスト上でスタンバイ・データベース・インスタンスを起動して、インスタンス化の準備を行います。

```
$> export ORACLE_SID=stbydb
$> sqlplus / as sysdba
SQL> startup nomount
```

18. ネットワーク接続の確認

この時点で、Oracle Net はプライマリ環境とスタンバイ環境の両方の TNS 別名をスタンバイ環境から解決できる必要があります。

```
$ tnsping proddb
$ tnsping stbydb
$ sqlplus sys/<password>@//pnode1:1521/proddb as sysdba
```

スタンバイ・データベースのインスタンス化

このセクションでは、プライマリ環境とスタンバイ環境の設定が完了した後のスタンバイ・データベースのインスタンス化について説明します。

19. データベースの複製

Oracle Recovery Manager（Oracle RMAN）を使用すると、DUPLICATE DATABASE コマンドでスタンバイ・データベースを作成できます。複製プロセスの一部として、パラメータ・ファイル、パスワード・ファイル、制御ファイル、データベース・ファイルがプライマリ環境からスタンバイ環境にコピーされます。

スタンバイ操作のパラメータ設定に必要な変更は、Oracle RMAN の DUPLICATE DATABASE コマンドでも指定する必要があります。Oracle RMAN によりプライマリのパラメータ・ファイルがコピーされると、それに応じて、DUPLICATE DATABASE コマンドで指定したパラメータが変更されます。

パスワード・ファイルもコピーされるため、スタンバイ・データベースのパスワードはプライマリ・データベースと同じになります。

```
$rman
connect target sys/welcome1@//pnode1:1521/proddb
connect auxiliary sys/welcome1@//snode1:1521/stbydb
run {
allocate channel p1 type disk;
allocate channel p2 type disk;
allocate channel p3 type disk;
allocate channel p4 type disk;
allocate auxiliary channel s1 type disk;
allocate auxiliary channel s2 type disk;
allocate auxiliary channel s3 type disk;
```

```
allocate auxiliary channel s4 type disk;
duplicate target database for standby from active database
dorecover
spfile
parameter_value_convert='/proddb','/stbydb'
set db_unique_name = 'stbydb'
set cluster_database = 'false'
set audit_file_dest = '/u01/app/oracle/admin/stbydb/adump'
set db_create_file_dest = '/u02/app/oracle/oradata/datastore'
set log_archive_config="DG_CONFIG=(proddb,stbydb)"
set diagnostic_dest="/u01/app/oracle"
set db_recovery_file_dest="/u01/app/oracle/fast_recovery_area/datastore"
set db_create_online_log_dest_1="/u01/app/oracle/oradata/datastore"
set control_files="/u01/app/oracle/oradata/datastore/stbydb/control01.ctl"
set log_archive_dest_2 = 'service=proddb lgwr async register valid_for=(online_logfiles,
primary_role)'
db_unique_name=proddb
set remote_listener = 'standby-scan:1521'
set fal_server='proddb'
nofilenamecheck;
}
```

20. フラッシュバック・データベースの有効化

スタンバイ・データベースでフラッシュバック・データベースを有効化し、必要に応じて保存期間を調整します。次に例を示します。

```
SQL> alter database flashback on;
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120;
```

21. スタンバイのspfileをASMに移動

スタンバイの spfile をローカル・ファイル・システムからスタンバイ・データベースの Automatic Storage Management（ASM）に移動します。

```
$ export ORACLE_SID=stbydb
RMAN> connect target /
RMAN> backup spfile;
RMAN> restore spfile to '/u02/app/oracle/oradata/datastore/stbydb/spfilestbydb.ora';
```

22. 管理リカバリ・モードの開始

次のように、スタンバイ・データベースでの管理リカバリをリアルタイム・モードで開始します。

```
SQL> alter database recover managed standby database disconnect;
```

注：'USING CURRENT CONTROLFILE'句は Oracle Release 12c では非推奨となっています。リアルタイム適用が Oracle Release 12c のデフォルト動作です。

リアルタイム適用では REDO 適用サービスは、現在のスタンバイ REDO ログがアーカイブされるのを待たずに、受信したらすぐに REDO ログをスタンバイ・データベースに適用できます。そのため、スタンバイ REDO ログ・ファイルはフェイルオーバーまたはスイッチオーバーが発生した時点ですでに適用されているため、フェイルオーバーとスイッチオーバーの時間が短縮されます。

インスタンス化後の手順

以下の手順は、スタンバイ・データベースのインスタンス化の完了後に実行します。

23. Oracle Clusterwareによるスタンバイ・データベースの登録

ORACLE_HOME 環境変数が正しく設定されていることを確認します。Oracle Clusterware を使用して、クラスタの 1 つのノードから実行されるシングル・インスタンスとしてスタンバイ・データベースを登録します。

Active Data Guard 以外の構成の場合：

```
$ srvctl add database –d stbydb –o $ORACLE_HOME –p
"/u02/app/oracle/oradata/datastore/stbydb/spfilestbydb.ora" –r physical_standby –s
mount -c SINGLE -x snode1
```

Active Data Guard 構成の場合：

```
$ srvctl add database –d stbydb –o $ORACLE_HOME –p
"/u02/app/oracle/oradata/datastore/stbydb/spfilestbydb.ora" –r physical_standby –s
'read only' -c SINGLE -x snode1
```

1. Oracle RACへのスタンバイ・データベースの変換

この手順は任意です。この段階では、スタンバイ・データベースはシングル・インスタンス・データベースとして構成されています。プライマリ・データベースが Oracle RAC データベースであった場合、スタンバイ・データベースも Oracle RAC スタンバイに変換できます。rconfig ツールを使用してシングル・インスタンス・データベースを Oracle RAC スタンバイ・データベースに変換する方法については、付録 B を参照してください。

2. Data Guard Broker構成の設定

この手順は任意です。Data Guard Broker 構成を作成すると、Data Guard 環境全体を単一のエンティティとして容易に管理できます。この構成はローカルとリモートの両方で使用可能な管理機能、保守機能、監視機能を提供します。Data Guard Broker 構成の設定についての詳細は、付録 C を参照してください。

3. 専用のDRネットワークの設定

この手順は任意です。専用のネットワークを使用するように REDO 転送サービスを構成できます。専用のネットワーク・チャネルは、REDO 転送のパフォーマンスの向上に役立ちます。特に、アプリケーションのネットワーク・トラフィックによってパブリック・ネットワークで使用可能な帯域幅の大部分が消費されている場合に効果的です。Data Guard の REDO 転送用に専用のネットワーク・チャネルを設定する方法については、MOS Note 1451810.1、『Configuring Dedicated Disaster Recovery Network on Oracle Database Appliance』を参照してください。

4. **構成と設定の確認**

スタンバイ・データベースで内部データ・ディクショナリ・ビューを使用して、スタンバイ・データベースの動作を確認できます。

```
$ srvctl config database -d stbydb
SQL> select database_role, switchover_status from v$database;
SQL> select thread#, sequence#, applied from v$archived_log order by sequence#;
```

付録B：シングル・インスタンス・データベースからOracle RACへの変換

rconfig コマンドライン・ユーティリティを使用して、シングル・インスタンス・データベースをOracle RAC データベース、または Oracle RAC One Node に変換できます。

この機能を使用するには、次の手順を実行します。

構成XMLファイルの作成

convert.xml として保存する構成 XML ファイルの例を以下に示します。このファイルは、システムでの必要に応じて変更できます。

サンプル XML ファイルは$ORACLE_HOME/assistants/rconfig/sampleXML ディレクトリにあります。

注：変換が正しく行えることを確認するため、最初に変換オプション `Convert verify="ONLY"`を設定してテスト変換を実行してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<n:RConfig xmlns:n="http://www.oracle.com/rconfig"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://www.oracle.com/rconfig rconfig.xsd">
 <n:ConvertToRAC>
    <n:Convert verify="YES">
       <n:SourceDBHome>/u01/app/oracle/product/12.1.0.2/dbhome_1</n:SourceDBHome>
       <n:TargetDBHome>/u01/app/oracle/product/12.1.0.2/dbhome_1</n:TargetDBHome>
       <n:SourceDBInfo SID="stbydb">
         <n:Credentials>
           <n:User>sys</n:User>
           <n:Password>welcome1</n:Password>
           <n:Role>sysdba</n:Role>
         </n:Credentials>
       </n:SourceDBInfo>
       <n:NodeList>
         <n:Node name="snode1"/>
         <n:Node name="snode2"/>
       </n:NodeList>
       <n:InstancePrefix>stbydb</n:InstancePrefix>
         <n:SharedStorage type="CFS">
            <n:TargetDatabaseArea></n:TargetDatabaseArea>
            <n:TargetFlashRecoveryArea></n:TargetFlashRecoveryArea>
         </n:SharedStorage>
   </n:Convert>
 </n:ConvertToRAC>
```

```
</n:RConfig>
```

rconfigツールの実行

変更が完了したら、ファイルを保存します。スタンバイ・データベースで次のコマンドを実行します。convert.xml は上記で構成した XML 入力ファイルの名前です。

```
$ rconfig convert.xml
```

Cluster Ready Servicesリソースの更新

変換後のデータベースに対応するように、Oracle Cluster Ready Services（Oracle CRS）リソースを更新する必要があります。

```
$ srvctl modify database -d stby -r physical_standby -s mount
```

構成の確認

スタンバイ・データベースの構成を確認します。

```
$ srvctl config database -d stby
```

付録C：Data Guard Broker構成の作成

このセクションでは、Oracle Data Guard Broker 構成のプロセスを説明します。

静的登録を行うためのリスナーの構成

プライマリ・データベースとスタンバイ・データベースのすべてのインスタンスを静的に登録するためのリスナーを構成します。Oracle Database Appliance では、リスナーは Grid Infrastructure ホームから実行されます。Oracle RAC のプライマリ/スタンバイ構成の静的登録の例を以下に示します。

ノード pnode1：

```
SID_LIST_LISTENER =
    (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = proddb_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
    (SID_NAME = proddb1)))
```

ノード pnode2：

```
SID_LIST_LISTENER =
    (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = proddb_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
    (SID_NAME = proddb2)))
```

ノード snode1：

```
SID_LIST_LISTENER =
    (SID_LIST =
    (SID_DESC =
    (GLOBAL_DBNAME = stbydb_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
    (SID_NAME = stbydb1)))
```

ノード snode2：

```
SID_LIST_LISTENER =
    (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = stbydb_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
    (SID_NAME = stbydb2)))
```

Broker構成ファイルの設定

プライマリ・データベースとスタンバイ・データベースの両方で Broker 構成ファイルの場所を設定します。

ノード pnode1：

```
SQL> ALTER SYSTEM SET
DG_BROKER_CONFIG_FILE1='/u02/app/oracle/oradata/datastore/stbydb/broker/dr1.dat'
SCOPE=BOTH SID='*';
SQL> ALTER SYSTEM SET
DG_BROKER_CONFIG_FILE2='/u02/app/oracle/oradata/datastore/stbydb/broker/dr2.dat '
SCOPE=BOTH SID='*';
```

ノード snode1：

```
SQL> ALTER SYSTEM SET
DG_BROKER_CONFIG_FILE1='/u02/app/oracle/oradata/datastore/stbydb/broker/dr1.dat'
SCOPE=BOTH SID='*';
SQL> ALTER SYSTEM SET
DG_BROKER_CONFIG_FILE2='/u02/app/oracle/oradata/datastore/stbydb/broker/dr1.dat'
SCOPE=BOTH SID='*';
```

Data Guard Brokerの有効化

プライマリ・データベースとスタンバイ・データベースの両方で Data Guard Broker を有効化します。

ノード pnode1：

```
SQL> ALTER SYSTEM SET DG_BROKER_START='TRUE' SCOPE=BOTH SID='*';
```

ノード snode1：

```
SQL> ALTER SYSTEM SET DG_BROKER_START='TRUE' SCOPE=BOTH SID='*';
```

Broker構成の作成

プライマリ・データベースの DB_UNIQUE_NAME とそれに対応する TNS 別名を使用して、プライマリで Broker 構成を作成します。

```
DGMGRL> connect sys/welcome1;
DGMGRL> CREATE CONFIGURATION 'ODADGConfig' AS
> PRIMARY DATABASE IS 'PRODDB'
> CONNECT IDENTIFIER is PRODDB;
```

Data Guard Broker構成へのスタンバイ・データベースの追加

スタンバイ・データベースの DB_UNIQUE_NAME を使用して構成にスタンバイ・データベースを追加します。

```
DGMGRL> ADD DATABASE 'STBYDB' AS CONNECT IDENTIFIER IS STBYDB;
```

構成の有効化

次のように Data Guard Broker 構成を有効化します。

```
DGMGRL> ENABLE CONFIGURATION;
```

構成の確認

次のコマンドを実行して、作成された構成を確認します。

```
DGMGRL> show configuration;
DGMGRL> show database verbose prim;
DGMGRL> show instance verbose prim1;
DGMGRL> show instance verbose prim2;
DGMGRL> show database verbose stby;
DGMGRL> show instance verbose stby1;
DGMGRL> show instance verbose stby2;
```

参考資料

OTN の Oracle Database Appliance の Web サイト

http://www.oracle.com/technetwork/jp/database/database-appliance/overview/index.html

OTN の Oracle Real Application Clusters の Web サイト

http://www.oracle.com/technetwork/jp/database/options/clustering/overview/index.html

OTN の Oracle Clusterware の Web サイト

http://www.oracle.com/technetwork/jp/database/clusterware/overview/index.html

OTN の Oracle Data Guard の Web サイト

http://www.oracle.com/technetwork/jp/database/options/active-data-guard/overview/index.html

Oracle Maximum Availability Architecture

http://www.oracle.com/technetwork/jp/database/features/availability/maa-098142-ja.html

Oracle Data Guard 概要および管理 12c リリース 1

http://docs.oracle.com/database/121/SBYDB/toc.htm

My Oracle Support（MOS）Note 1075908.1

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1075908.1

OTN の Oracle Database 高可用性の Web サイト

http://www.oracle.com/technetwork/jp/database/features/availability/index.html

Oracle® Database 高可用性ベスト・プラクティス 12c リリース 1

http://docs.oracle.com/database/121/HABPT/toc.htm

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200

CONNECT WITH US

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

Hardware and Software, Engineered to Work Together

Copyright © 2015, 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、記載内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

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

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。1215

ホワイト・ペーパー： Oracle Database Appliance による Oracle Data Guard のデプロイメント
2016 年 1 月
著者： Ramasubramanian Athmanathan、Ravi Sharma、Ravinder Ransi
共著者： Oracle RAC Pack