

Oracle Maximum
Availability Architecture

Oracleホワイト・ペーパー
2011年12月

Sun ZFS Storage ApplianceとOracle Data Guardを使用したデータベース・ クローニング

概要	2
このドキュメントについて	2
対象読者	3
Sun ZFS Storage Applianceについて	4
アーキテクチャの概要	4
プラットフォーム	5
概念と用語	5
データ・サービス	6
Oracle Databaseのクローニング・アーキテクチャと手順	8
手法の概要	8
ホストとストレージの構成	10
データベース・クローニング用のアーキテクチャ	13
フィジカル・スタンバイ・データベースの構成	15
スタンバイ・データベースのクローニング	18
配置シナリオの例	24
配置シナリオ1：すべてのスタンバイ・データベースおよびクローンを1つのホストに配置	26
配置シナリオ2：クローン・データベースを異なるホストに配置	27
配置シナリオ3：スタンバイおよびクローンを新規ホストに配置	28
Sun ZFS Storage Applianceを使用したデータベース・クローニングの利点	29
結論	33
付録	34
サンプル・スクリプト	34
用語集	37
リソース	39

概要

データベース管理者は、アプリケーション開発とその後のコード・テストに対する絶え間ない要求に対応するため、ミッション・クリティカルな大規模データベースを効率的に複製しなければならないという課題に直面しています。また、大規模な本番システムに関連する多数の開発作業とテスト作業に対応するために、それぞれの本番データベースに対して複数のクローンが必要とされることも多いため、この問題はさらに複雑になります。さらに、クローニング・プロセスが本番データベースに影響を与えないことも重要になります。クローンを作成するための基本的な方法は、最新のバックアップから別のデータベース・サーバーへデータベースをリストアする方法ですが、このような方法がいかに非効率で時間を要するかはすぐに明らかになります。効率的な本番データベースのクローニング・ソリューションなしでは、ストレージ消費量とコストが増加するだけでなく、企業が大きな管理負担を抱え込むことで、よりタイム・クリティカルなサポート機能から注意がそらされる結果になります。

ミッション・クリティカルなOracle Databaseに高可用性と障害時リカバリ機能を提供するため、すでに多数のオラクル顧客がOracle Data Guardを配置しています。Sun ZFS Storage Appliance（このドキュメントではアプライアンスと呼びます）がオラクルのソリューション・ポートフォリオに加わったことで、クローニング要件を満たすためのまたとない機会がData Guardユーザーにもたらされました。アプライアンスは既存のData Guard構成に対して容易に追加でき、企業のスナップショット要件とクローン要件に対して、低コストで時間効率とスペース効率に優れた業界最高のソリューションを提供します。

このドキュメントについて

このドキュメントでは、Sun ZFS Storage Applianceのスナップショットおよびクローニング機能と組み合わせてOracle Data Guard機能を配置することで、簡単かつ効率的にスタンバイ・データベースのデータベース・クローニングを実行する方法について説明します。また、オラクルによって検証されたベスト・プラクティスとデータベース・クローニング処理の自動化スクリプトを提供します。

このドキュメントで説明するソリューションは、次のオペレーティング・システムとデータベースのバージョンをサポートしています。

表1: サポートされるバージョン

パラメータ	バージョン
オペレーティング・システム	Oracle Solaris 10オペレーティング・システム (SPARC、X86) RedHat Linux 4.x、5.x (およびそれ以降) Oracle Enterprise Linux 4.x、5.x (およびそれ以降)
Oracleデータベース	10.2.0.1 (およびそれ以降) 単一インスタンスのスタンバイ・データベース Oracle Data Guard、Active Data Guard Oracle Exadata Hybrid Columnar Compression (HCC) を使用する場合は、Oracle 11.2.0.3以降
ストレージ	Oracle Sun ZFS Storage Appliance
プロトコル	NFSv3、NFSv4
スタンバイ・データベースとクローン・データベース	同じホストを共有 別のスタンバイ・ホストと同じホストを共有 スタンバイ・インスタンスとクローン・インスタンスを別々のホストに配置

NFSプロトコルを使用して、スナップショットおよびクローン処理のサポート専用新しいスタンバイ・データベースをアプライアンス上に作成します (1つのData Guard構成で、最大30のスタンバイ・データベースをサポートします)。スタンバイ・データベースを管理するOracleインスタンスは、既存のスタンバイ・データベースと同じサーバー上に配置することも、異なるサーバー上に配置することもできます。既存のプライマリ・データベースおよびスタンバイ・データベースで使用されているストレージは、どのベンダーのものでも構いません。この方式では、既存のOracle Data Guard環境を中断することなく、Sun ZFS Storage Applianceを素早く配置できます。

アプライアンス上にスタンバイ・サイトを構築すると、スタンバイ・データベースのファイル・システムに対してスナップショットが実行され、その時点でのデータのコピーが作成されます。次に、スナップショット・データがクローニングされ、再度、ファイル・システムに対する読取りと書込みが可能になります。クローニングされたスタンバイ・データベースはアクティブ化されてから別のデータベースに変換され、テストや開発、およびQA目的で使用されます。

対象読者

本書は、Oracleデータベース管理者、ストレージ/システム管理者、および技術営業担当者を対象としています。また、読者がOracle Data Guardの機能に精通していることを前提としています。

Sun ZFS Storage Applianceについて

アーキテクチャの概要

ここでは、Sun ZFS Storage Applianceの概要について説明します。特徴、機能、および各種プラットフォーム製品について、詳しくは次のWebサイトを参照してください。

<http://www.oracle.com/jp/products/servers-storage/storage/nas/overview/index.html>

Sun ZFS Storage Applianceは、管理の簡易性、マルチプロトコル接続、およびビジネス継続性向けデータ・サービスを1つのアプライアンスに統合したものです。このアプライアンスは、NFS、Common Internet File System (CIFS)、Internet Small Computer System Interface (iSCSI)、InfiniBand (IB)、およびファイバ・チャネル (FC) をサポートしています。シングル・ヘッドとクラスタ・ヘッド (高可用性のため) のいずれとしても使用できるこのアプライアンスは、バックアップおよびリストアのためにNetwork Data Management Protocol (NDMP) をサポートしています。Oracle Solaris ZFSを使用したOracle Solarisオペレーティング・システムがこのアプライアンスの中核をなし、ZFS ファイル・システムによってすべてのデータ・ストレージ・サービス、管理サービス、およびデータ・サービスが機能します。これらのサービスに対しては、直感的なユーザー・インタフェースまたはCLIモードを使用してアクセスできます。

このストレージ・アーキテクチャには、DRAM、フラッシュ、物理ディスクがシームレスに統合されたハイブリッド・ストレージ・プール (HSP) が採用されているため、データが効率的に配置されます。ユーザーのI/O要求やパターンに基づいて、これら各層間のデータ移動がアプライアンスによって自動的に処理されます。また、ストレージには*Analytics (解析)* という強力なパフォーマンス監視ツールが搭載されています。このツールは、ネットワーク、ストレージ、ファイル・システム、クライアント・アクセスなどの各種コンポーネントのパフォーマンスについての詳細情報を提供します。多彩なドリルダウン・オプションも用意されており、たとえば、Analyticsでは、どのクライアントがどのファイル・システムのどのファイルにアクセスしているか、待機時間や転送サイズはどのくらいであるかなどを表示できます。

さらに、このストレージには、アプリケーションの容量要件、保護要件、パフォーマンス要件のバランスを調整する各種RAID保護機能が搭載されています。

プラットフォーム

Sun ZFS Storage Applianceは、価格、パフォーマンス、容量、保護機能に対する顧客ニーズに合わせて3つのモデルで提供されています。ミッドレンジからハイエンドのプラットフォームには最大2TBの読み取りキャッシュが搭載されているため、アプライアンスの応答時間は多くの場合、数ミリ秒になります。また、4つすべてのプラットフォームの書き込みフラッシュで、同期書き込み時の応答時間が1ミリ秒未満に短縮されています。

新しいSun ZFS Storage Applianceプラットフォームは、ミッション・クリティカルなアプリケーションのストレージ要件を満たす高速CPU、大容量フラッシュ・キャッシュ、大容量ストレージ、高スループット、広帯域幅を提供します。

表2に、各モデルの詳しい構成を示します。

表2: Sun ZFS Storage Applianceのモデル

プラットフォーム	ストレージ容量	プロセッサ	メモリ (DRAM)	書き込み用に最適化されたSSD	読み取り用に最適化されたSSD	クラスターオプション
Sun ZFS Storage 7120	最大60 x 2TBのSAS Disk [120TB]	1xクアドコア・インテル Westmere-EP E5620 @ 2.4GHz	最大36GB	最大96GB	N/A	なし
Sun ZFS Storage 7320 (詳細はコントローラ当たり)	最大96 x 2TBのSAS Disk [192TB]	2xクアドコア・インテル Westmere-EP E5620 @ 2.4GHz	最大72GB	最大16x18GB	最大4x512GB	あり
Sun ZFS Storage 7420 (詳細はコントローラ当たり)	最大576 x 2TBのSAS Disk [1.1PB]	4x6CインテルNehalem-EX E7530 @ 1.86GHz [または] 4x8C インテルNehalem-EX X7550 @ 2GHz	最大512GB	最大96x18GB	最大4x512GB	あり

概念と用語

ストレージ・プール

ストレージ・プール（ボリューム・グループとほぼ同義）は、一連の物理ディスク上に作成されません。その後、ストレージ・プール上にファイル・システムが作成されます。ユーザーは使用可能な物理ディスク上に1つ以上のストレージ・プールを作成し、ストレージ・プールの構成時に、プールに対してフラッシュ・ドライブを割り当てることができます。

プロジェクト

すべてのファイル・システムとLUNは、プロジェクトとしてグループ化されます。また、プロジェクトはコンシステンシ・グループとしてもみなされます。プロジェクトには、シェアを管理する共通管理コントロール・ポイントが定義されます。プロジェクト内のすべてのシェアは共通設定を共有でき、割当てはシェア・レベルの他、プロジェクト・レベルで実施できます。プロジェクトはまた、論理的に関連したシェアをグループ化する目的でのみ使用されるため、共通属性（蓄積領域など）へのシングル・ポイント・アクセスが可能になります。

シェア

シェアとは、サポートされているデータ・プロトコルを使用してアプライアンスのクライアントにエクスポートされるファイル・システムやLUNです。ファイル・システムはファイルベースの階層をエクスポートし、CIFS、NFS、HTTP/WebDav、FTP、およびInfiniBandを使用してアクセスされます。LUNはブロックベースのボリュームをエクスポートし、iSCSIおよびファイバ・チャネル・プロトコルを使用してアクセスされます。

プロジェクト/シェアは、プール内のシェアに対する一意の識別子です。複数のプロジェクトに同一名を持つ複数のシェアを格納することはできますが、1つのプロジェクトに同一名を持つ複数のシェアを格納することはできません。1つのプロジェクトにファイル・システムとLUNの両方を格納して、同じ名前空間を共有できます。

データ・サービス

このアプライアンスには多数のデータ・サービスがバンドルされております。データベース・クローニング手順に関連したおもなデータ・サービスは、次のとおりです。

スナップショット

Sun ZFS Storage Applianceは、無制限のスナップショット機能をサポートしています。スナップショットは、ファイル・システムに対する読取り専用のポイント・イン・タイム・コピーです。作成はただちに実行されますが、当初は領域が割り当てられません。ブロック割当ては、ベース・ファイル・システムが変更されると実施されます（Copy-On-Write）。スナップショットは、手動で開始するか、または特定の間隔で自動実行されるようスケジュール設定できます。このスナップショット・データへは、任意のバックアップ目的で直接アクセスできます。

スナップショット・ブロックの読取りは、ベース・ファイル・システムのブロックで処理されます。ベース・ファイル・システムが変更されると、古いブロックがスナップショットによって参照されるようになり、新しく変更されたブロックはファイル・システムによって参照されます。

プロジェクト・スナップショットは、プロジェクト内のすべてのシェアに対してスナップショットを実行する処理に相当します。

クローン

このアプライアンスは、無制限の数のクローンをサポートしています。クローンは、即時作成される読み取り/書き込み可能なスナップショットのコピーです。1つのスナップショットから、1つまたは複数のクローンが作成されます。これらのクローンは、通常のファイル・システムとしてユーザーに表示されます。クローンからのスナップショットの取得を含めたすべての通常操作をクローンに対して実行できます。クローンは通常、テスト、開発、QA、およびバックアップ環境で使用されます。

スナップショットと同様に、クローンの作成時には、領域の割当ては行われません。クローンの読み取りは、ベース・ファイル・システムのブロックで処理されます。クローン内のブロックが変更された場合にのみ、変更されたブロックが割り当てられます。スナップショットとクローン間で領域が共有され、1つのスナップショットに複数のクローンが格納されるため、スナップショットを破棄するとアクティブなクローンも破棄されます。

クライアントからは、クローンのファイル・システムが独立したファイル・システムであるかのように見えます。クローンにアクセスするための特別な要件はありません。

スナップショットのロールバック

スナップショットのロールバックとは、スナップショットが取得された時点でベース・ファイル・システムに戻すプロセスです。ロールバック・プロセスは、スナップショットの取得時からロールバック時まで発生したベース・ファイル・システムでのすべての変更を破棄します。これによって、データのリストア・プロセスは不要になります。

Sun ZFS Storage Applianceで利用できるデータ・サービスの一覧については、次のドキュメントURLを参照してください。

<http://wikis.oracle.com/display/FishWorks/Documentation>

Oracle Databaseのクローニング・アーキテクチャと手順

手法の概要

スタンバイ・データベースのクローニングは、Sun ZFS Storage Applianceのスナップショットおよびクローン機能を使用して実行されます。おおまかな手順について、次に説明します。

Sun ZFS Storage Applianceでの新しいスタンバイ・データベースのセットアップ

注：これは1回限りのセットアップ・プロセスです。

- アプライアンスから、スタンバイ・データベースを格納および処理するためのプロジェクトとファイル・システムをセットアップします。
- データベース・サーバーにファイル・システムをマウントします。
- Oracle Recovery Manager (Oracle RMAN) を使用して、プライマリの本番データベースをスタンバイ・データベースに複製します。
- Oracle Data GuardまたはActive Data Guardをセットアップし、プライマリ・サイトと新しいスタンバイ・サイト間にリアルタイム・レプリケーションを確立します。
- 管理リカバリ手順を有効化します。Oracleのバージョン11g Release 2（またはそれ以降）を使用している場合、カスケード・スタンバイを構築することもできます。

データベース・クローニングの手順

- 管理リカバリを停止し、スタンバイ・データベースを一貫性のある状態にします。
- Sun ZFS Storage Applianceのスナップショット機能を使用して、スタンバイ・データベースが格納されたプロジェクトのスナップショットを取得します。
- スタンバイ・サイトの管理リカバリを再開します。
- ステップ2で取得したスナップショットからSun ZFS Storage Applianceに含まれるファイル・システムをクローニングします。
- NFSを使用してクローニングしたファイル・システムを、データベース・サーバーにマウントします。
- クローニングしたスタンバイ・データベースを、テスト、開発、およびQA向けの読取り/書込み可能なデータベースに変換します。

図1にファイル・システム、スナップショット、およびクローン間の関係を示します。ファイル・システムとスナップショット間およびスナップショットとクローン間の関係は1対多の関係になります。

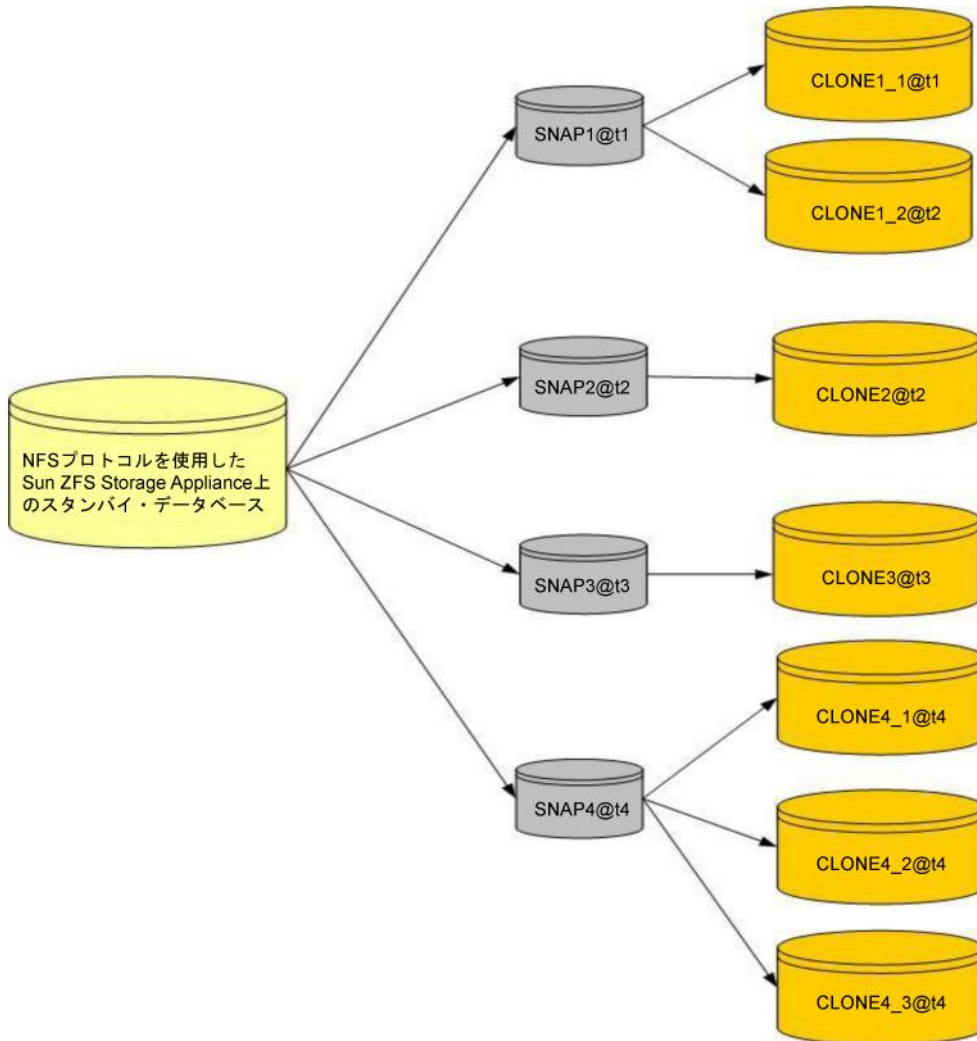


図1：ベース・ファイル・システムに対するスナップショットとクローンの関係

ホストとストレージの構成

ここでは、データベースのクローニング手順をテストするためにラボで使用された構成について、詳しく説明します。

表3: プライマリ・サイトの構成

パラメータ	値
ホスト名	aie-6300b
システム構成	Sun Blade T6300/UltraSPARC-T1、32コアCPU/16GB
オペレーティング・システム	Solaris 10 U 8 (SPARC)
ストレージ接続	FC – 500GB LUNS
Oracleデータベース	11.2.0.1、単一インスタンス、Oracle ASM
Oracleインスタンス名	PRODDB
initPRODDB.ora (新規スタンバイ用に変更する部分のみ)	log_archive_dest_3= 'service=PRODDB_DR async db_unique_name=PRODDB_DR valid_for=(primary_role,online_logfile) log_archive_config= 'dg_config=(PRODDB,PRODDB_SBY,PRODDB_DR) log_archive_dest_state_2= ENABLE
Oracle ASMディスク・グループ	+PRODDG
データファイル、REDOファイル、制御ファイル	+PRODDG/data/* +PRODDG/logs/*

表4: NFSスタンバイ・データベース向けのスタンバイ・ホスト構成

パラメータ	値										
ホスト名	aie-6300c										
システム構成	Sun Blade T6300/UltraSPARC-T1、32コアCPU/16GB										
オペレーティング・システム	Solaris 10 U 8 (SPARC)										
ストレージ接続	10Gbイーサネット										
Oracleデータベース	11.2.0.1、単一インスタンス、NFS										
マウント・ポイント	<table border="1"> <thead> <tr> <th>クライアント</th> <th>Sun ZFS Storage Appliance</th> </tr> </thead> <tbody> <tr> <td>/oradata/stby/data</td> <td>/export/standby_db/data</td> </tr> <tr> <td>/oradata/stby/logs</td> <td>/export/standby_db/logs</td> </tr> <tr> <td>/oradata/stby/archive</td> <td>/export/standby_db/archive</td> </tr> <tr> <td>/oradata/stby/alerts</td> <td>/export/standby_db/alerts</td> </tr> </tbody> </table>	クライアント	Sun ZFS Storage Appliance	/oradata/stby/data	/export/standby_db/data	/oradata/stby/logs	/export/standby_db/logs	/oradata/stby/archive	/export/standby_db/archive	/oradata/stby/alerts	/export/standby_db/alerts
クライアント	Sun ZFS Storage Appliance										
/oradata/stby/data	/export/standby_db/data										
/oradata/stby/logs	/export/standby_db/logs										
/oradata/stby/archive	/export/standby_db/archive										
/oradata/stby/alerts	/export/standby_db/alerts										
NFSマウント・オプション	rw,bg,hard,nointr,rsize=32768,wsz=32768,proto=tcp,vers=3,actimeo=0										
ORACLE_SID	PRODDB_DR										
ORACLE_HOME	/oracle/products/11.2.0/db (ホストに対してローカル)										
initPRODDB_DR.ora	*.control_files= '/oradata/stby/logs/control0.ctl', '/oradata/stby/logs/control1.ctl' *.db_block_size=8192 *.db_cache_size=4096M *.db_file_multiblock_read_count=128 *.db_file_name_convert='+PRODDG/DATA', '/oradata/stby/data' *.db_files=200 *.db_keep_cache_size=256M *.db_name='PRODDB' *.db_recycle_cache_size=256M *.db_unique_name='PRODDB_DR' *.diagnostic_dest= /oradata/stby/alerts										

パラメータ	値
	<pre> *.fal_client=PRODDB_DR' *.fal_server=PRODDB' *.file_systemio_options='SETALL' *.instance_name=PRODDB_DR' *.java_pool_size=16M *.large_pool_size=16M *.log_archive_dest_2='service=PRODDB async db_unique_name=PRODDB valid_for=(primary_role,online_logfile)' *.log_archive_dest_state_2=DEFER *.log_buffer=104857600 *.log_checkpoint_interval=10000 *.log_checkpoint_timeout=1800 *.log_file_name_convert='+PRODDG/LOGS',/oradata/stby/logs' *.nls_date_format=MM/DD/YYYY HH24:MI:SS' *.processes=400 *.service_names='PRODDB_DR' *.shared_pool_size=256M *.log_archive_dest_1='location=/oradata/stby/archive_1' *.undo_management='auto' PRODDB_DR.undo_tablespace='UNDOTBS1' *.log_archive_config='dg_config=(PRODDB,PRODDB_DR)' </pre>
Listener.ora	<pre> SID_LIST_LISTENER = (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = PRODDB_DR) (ORACLE_HOME = /oracle/products/11.2.0/db) (SID_NAME = PRODDB_DR))) </pre>
Tnsnames.ora	<pre> PRODDB_DR = (DESCRIPTION = (AADDRESS_LIST = (AADDRESS = (PROTOCOL = TCP)(HOST = aie-6300c)(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = PRODDB_DR) (ORACLE_SID=PRODDB_DR)))) PRODDB = (DESCRIPTION = (AADDRESS = (PROTOCOL = TCP)(HOST = aie-6300b)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (ORACLE_SID = PRODDB) (SERVICE_NAME = PRODDB)))) </pre>

表5: クローン・データベースのホスト構成

パラメータ	値
ホスト名	aie-6300c (スタンバイ・データベースと同じホストを使用する場合) aie-2000d (スタンバイ・データベースとは異なるホストを使用する場合)
システム構成	8GB/CPU
オペレーティング・システム	Solaris 10 U 8 (SPARC)
ストレージ接続	10Gbおよび1Gbイーサネット

パラメータ	値		
Oracleデータベース	11.2.0.1、単一インスタンス、NFS		
クローン名	CLONE1		
スナップショットに基づくクローン	SNAP1		
マウント・ポイント	<table border="1"> <tr> <td>クライアント /oradata/CLONE1/data /oradata/CLONE1/logs /oradata/CLONE1/archive /oradata/CLONE1/alerts</td> <td>Sun ZFS Storage Appliance /export/CLONE1/data /export/CLONE1/logs /export/CLONE1/archive /export/CLONE1/alerts</td> </tr> </table>	クライアント /oradata/CLONE1/data /oradata/CLONE1/logs /oradata/CLONE1/archive /oradata/CLONE1/alerts	Sun ZFS Storage Appliance /export/CLONE1/data /export/CLONE1/logs /export/CLONE1/archive /export/CLONE1/alerts
クライアント /oradata/CLONE1/data /oradata/CLONE1/logs /oradata/CLONE1/archive /oradata/CLONE1/alerts	Sun ZFS Storage Appliance /export/CLONE1/data /export/CLONE1/logs /export/CLONE1/archive /export/CLONE1/alerts		
スタンバイ・データベースのSRLIに対する参照	/oradata/stby/logs -スタンバイと同じホストを使用する場合、上記がマウント済み -スタンバイと異なるホストを使用する場合、/oradata/stby/logsのソフト・リンクは/oradata/CLONE1/logsを指す		
NFSマウント・オプション	rw,bg,hard,nointr,rsize=32768,wsiz=32768,proto=tcp,vers=3,actimeo=0		
ORACLE_SID	CLONE1		
ORACLE_HOME	/oracle/products/11.2.0/db (ホストに対してローカル)		
initCLONE1.ora	control_files= /oradata/CLONE1/logs/control0.ctl',/oradata/CLONE1/logs/control1.ctl' db_block_size=8192 db_cache_size=4096M db_file_multiblock_read_count=128 log_file_name_convert= '+PRODDG',/oradata/CLONE1',/stby',/CLONE1' db_file_name_convert= '+PRODDG',/oradata/CLONE1',/stby',/CLONE1' db_files=200 db_keep_cache_size=256M db_name= 'PRODDB' db_recycle_cache_size=256M db_unique_name= 'CLONE1' diagnostic_dest=/'oradata/CLONE1/alerts' file_systemio_options='SETALL' instance_name= 'CLONE1' java_pool_size=16M large_pool_size=16M log_archive_dest_state_2=DEFER log_buffer=104857600 log_checkpoint_interval=10000 log_checkpoint_timeout=1800 processes=400 service_names='CLONE1' shared_pool_size=256M log_archive_dest_1='location=/oradata/CLONE1/archive' undo_management='auto' CLONE1.undo_tablespace='UNDOTBS1'		
Listner.ora	SID_LIST_LISTENER = (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = CLONE1) (ORACLE_HOME = /oracle/products/11.2.0/db) (SID_NAME = CLONE1))))		
Tnsnames.ora	CLONE1 = (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = aie-6300c)(PORT = 1521)))) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = CLONE1) (ORACLE_SID=CLONE1)))		

表6: Sun ZFS Storage Applianceの構成

パラメータ	値	
ホスト名	aie-7420b	
システム構成	Sun ZFS Storage 7420、128GBメモリ	
接続性	10Gbイーサネット（データ用） 1Gbイーサネット（管理用）	
スタンバイ・データベースのプロジェクト	プロジェクト名: PRODDB_DR /export/standby_db/data /export/standby_db/logs /export/standby_db/archive /export/standby_db/alerts	レコード・サイズ 8KB 128KB 128KB 128KB
クローン・データベースのプロジェクト	プロジェクト名: CLONE1 /export/CLONE1/data /export/CLONE1/logs /export/CLONE1/archive /export/CLONE1/alerts	

表7: スイッチ構成

パラメータ	値
ネットワーク・スイッチ（ストレージ、管理用）	Extreme X350-48t、48ポート Cisco WS-X6708-10G-3C、8ポート
ストレージ・スイッチ（FC用）	Brocade 4100 32ポートのフル・ファブリック 4GB SWL SFP ファブリックOS : v6.1.0

データベース・クローニング用のアーキテクチャ

前項で説明したハードウェア構成を使用したOracleデータベースのクローニング手順について、図2にその構成と接続性を示します。

- ・ プライマリ本番データベース・インスタンス（PRODDB）はSun Storage 6780のファイバ・チャネル（FC）アレイ・ストレージを使用して、Sun Blade 6000ディスク・モジュール上で実行されます。
- ・ FCベースのスタンバイ・データベース・インスタンス（PRODDB_SBY）はStorageTek 6140アレイに格納されたデータを使用して、別のSun Blade 6000上で実行されます。

- NFSベースのスタンバイ・データベース・インスタンス (PRODDB_DR) はSun ZFS Storage Applianceに格納されたデータを使用して、同じSun Blade 6000ディスク・モジュール上で実行されます。
- スタンバイ・データベース・ファイル@ t1のスナップショットSNAP1はCLONE1にクローニングされ、SNAP2@ t2はCLONE2にクローニングされます。
- CLONE1データベース・インスタンスは、スタンバイ・データベースと同じSun Blade 6000ディスク・モジュールからアクセスされます。
- CLONE2データベース・インスタンスは、Sun Fire T2000サーバーからアクセスされます。

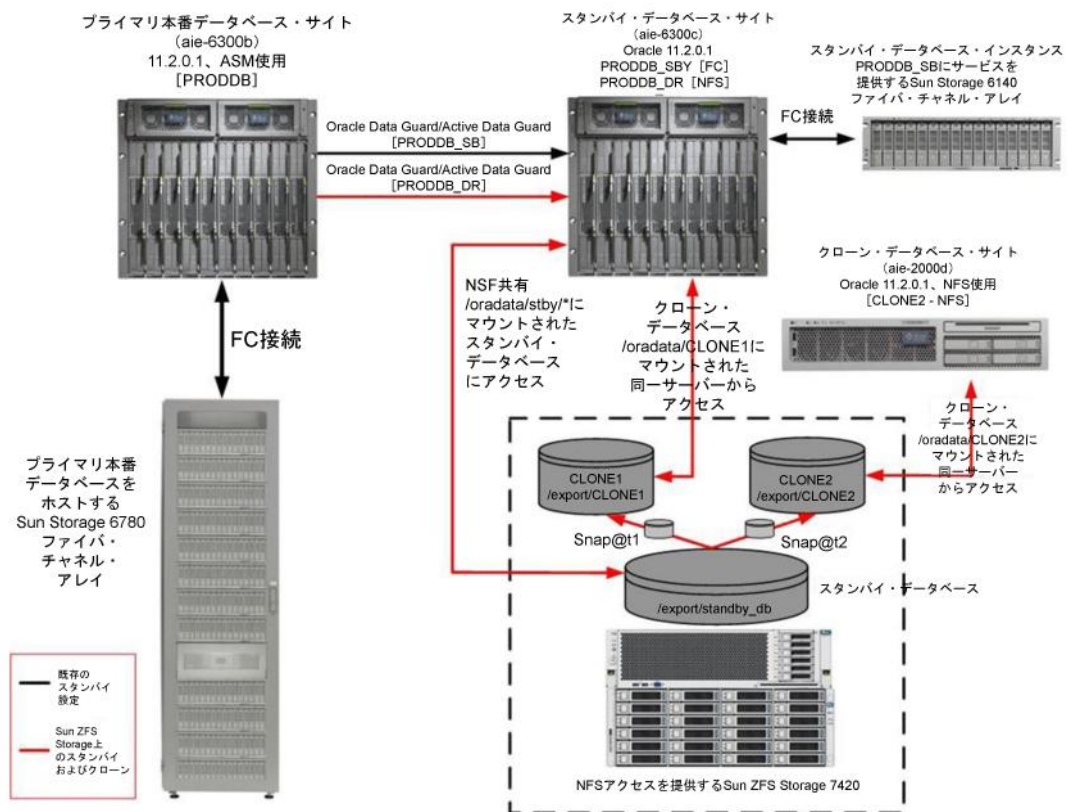


図2：データベース・クローニング用のアーキテクチャ図

フィジカル・スタンバイ・データベースの構成

ここでは、異なるプロトコルと異なるストレージを使用してフィジカル・スタンバイ・データベースを作成する方法について、簡単に説明します。このアーキテクチャでは、プライマリ・データベースはFCプロトコルを介して格納され、新しいスタンバイ・データベースはNFSを介して作成されます。

注：これは1回限りのセットアップ手順です。

プライマリ・サイトでのOracle Data Guardのセットアップ

プライマリ・データベースは、Oracle ASMを使用したファイバ・チャネル・ベースの高パフォーマンス・ストレージ (Sun Storage 6780アレイなど) 上にホストされています。データベースは、Oracle Real Application Clusters (Oracle RAC) データベースでも単一インスタンス・データベースでも構いません。プライマリ・サイトの初期化パラメータに対して、次の変更を実施します。

1. `log_archive_dest_#`を新しく追加して、新規のスタンバイ・サイトを指すように設定します。
2. `log_archive_config`に対して、新しいスタンバイ・サイトのインスタンスを追加します。
3. スタンバイ・サイトの構築が完了するまでの間、`log_archive_dest_state_#`にDEFERを設定しておきます。
4. 完了したら、DEFERステータスをENABLEに変更します。

Sun ZFS Storage Applianceでのスタンバイ・データベースのセットアップ

スタンバイ・データベースのデータファイルは、NFSプロトコルを使用して保存されます。データベースの複製にはOracle RMANが使用されます。データファイルおよびREDOログのロケーション変換は、初期化パラメータ`db_file_name_convert`および`log_file_name_convert`を介して処理されます。

1. スタンバイ・データベースの物理ファイルを格納するため、Sun ZFS Storage Appliance上にファイル・システムを作成します。詳しくは、Deploying Oracle Databases in Sun Storage 7000ドキュメントを参照してください。

プロジェクト：PRODDB_DR

アプライアンスのマウント・ポイント	データベース・サーバーのマウント・ポイント	説明
/export/standby_db/data	/oradata/stby/data	すべてのデータファイルを格納。
/export/standby_db/logs	/oradata/stby/logs	制御ファイル、オンラインREDOログ、スタンバイREDOログ (SRL) を格納。
/export/standby_db/archive	/oradata/stby/archive	送信されたアーカイブREDOログを格納。
/export/standby_db/alerts	/oradata/stby/alerts	診断の詳細情報 (オプション) を格納。データベース・サーバーにも格納可能。

2. `/etc/vfstab` (Solaris OSの場合) または`/etc/fstab` (Linuxの場合) の設定もしくはスクリプトを使用して、スタンバイ・ホスト`aie-6300d`にこれらのファイル・システムをマウントします。
3. Oracle RMANを使用してデータベースを複製します。


```

$ rman
RMAN> connect target sys/oracle@PRODDB;
connect auxiliary sys/oracle@PRODDB_DR;
run {
  allocate channel p1 type disk;
  allocate channel p2 type disk;
  allocate auxiliary channel s1 type disk;
  allocate auxiliary channel s2 type disk;
  duplicate target database for standby from active database
  nofilenamecheck;
}

```

結果 : ステップ3によって、スタンバイ・データベースの作成が本番データベースから直接開始されます。また、スタンバイREDOログも生成されます。

```
SQL > select member from v$logfile;
```

```
MEMBER
```

```

-----
/oradata/stby/logs/redo00-a.log
/oradata/stby/logs/redo01-a.log
/oradata/stby/logs/redo02-a.log
/oradata/stby/logs/redo03-a.log
/oradata/stby/logs/stby-1.log
/oradata/stby/logs/stby-2.log
/oradata/stby/logs/stby-3.log

```

```
7 rows selected.
```

```
SQL >select name from v$datafile;
```

```
NAME
```

```

-----
/oradata/stby/data/system00.dbf
/oradata/stby/data/sysaux00.dbf
/oradata/stby/data/undo00_a.dbf
/oradata/stby/data/undo01_a.dbf
/oradata/stby/data/data00.dbf
/oradata/stby/data/data01.dbf
/oradata/stby/data/data02.dbf

```

・・・その他

4. プライマリから新しいスタンバイ・データベースへのログ転送を有効化します。

```

$ sqlplus sys/oracle@PRODDB as sysdba SQL> alter system set
log_archive_dest_state_2=ENABLE;

```

5. スタンバイREDOログが使用されている場合、現在のログ・ファイルを使用して管理リカバリを実行します。

```
SQL > alter database recover managed standby database using current
logfile disconnect;
```

6. スタンバイREDOログが使用されていない場合は、次のコマンドを実行します。

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

7. SCNが互いを追跡していることを確認します。

```
SQL> select instance_name, substr(host_name,1,30) "Host Name" from
v¥$instance;
```

```
SQL> select DATABASE_ROLE,current_scn,protection_mode from
v¥$database;
```

```
INSTANCE_NAME Host Name
```

```
-----
PRODDB   aie-6300b
```

```
DATABASE_ROLE  CURRENT_SCN PROTECTION_MODE
```

```
-----
PRIMARY        5990864 MAXIMUM PERFORMANCE
=====
```

```
INSTANCE_NAME Host Name
```

```
-----
PRODDB_DR   aie-6300c
```

```
DATABASE_ROLE  CURRENT_SCN PROTECTION_MODE
```

```
-----
PHYSICAL STANDBY    5990862 MAXIMUM PERFORMANCE
```

結果 : Sun ZFS Storage Applianceでのスタンバイ・データベースのセットアップが完了しました。

スタンバイ・データベースのクローニング

次の手順を使用してスタンバイ・データベース上でスナップショットを実行し、クローンを作成したら、クローン・データベースからテスト用の開発データベースへと変換します。

ステップ1. スタンバイ・データベースのファイル・システムのスナップショットを作成します。

スタンバイ・データベースの管理リカバリを一時停止します。

Sun ZFS Storage ApplianceのプロジェクトPRODDB_DR内に格納されたスタンバイ・データベースに対して、ポイント・イン・タイム・スナップショットが取得されます。スナップショットを取得する前に、管理リカバリは中止され、プライマリからのログ送信も停止されます。これらのステップはシェル・スクリプト*before_snap.sh*内で処理されます。

```
$ sqlplus sys/oracle@PRODDDB
SQL> alter system set log_archive_dest_state_2=DEFER;
$ sqlplus sys/oracle@PRODDDB_DR
SQL > alter database recover managed standby database cancel;
SQL> alter database recover managed standby database disconnect ;
SQL > alter database recover managed standby database cancel;
```

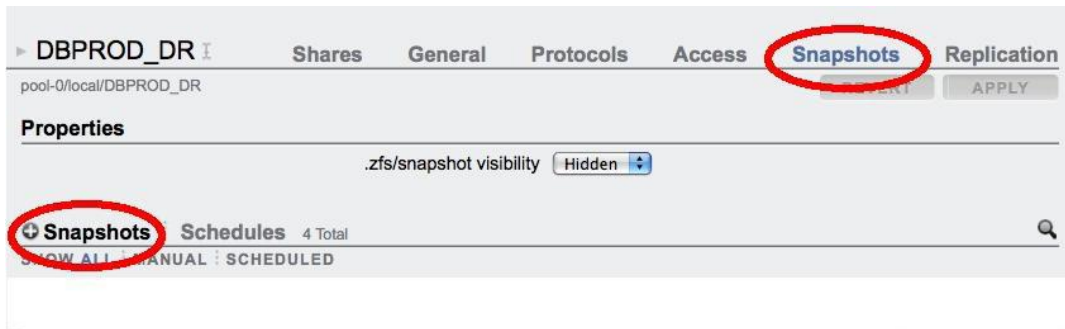
プロジェクトPRODDB_DRのスナップショット

SNAPIという名前を使用して、Sun Storage 7000アプライアンス内のプロジェクトのスナップショットを実行します。実行するには、グラフィカル・ユーザー・インタフェース (GUI) またはコマンドライン・インタフェース (CLI) を使用します。プロジェクト・レベルでスナップショットを実行すると、プロジェクト内のすべてのファイル・システムで一貫性の取れたスナップショットが取得されます。すべてのファイル・システムに対してSNAPIという名前の付いたスナップショットが作成されます。

The screenshot shows the Sun Storage 7000 GUI interface. At the top, there are navigation tabs: DBPROD_DR, Shares, General, Protocols, Access, Snapshots, and Replication. The 'Snapshots' tab is highlighted with a red circle. Below the tabs, the path 'pool-0/local/DBPROD_DR' is shown. Underneath, there are sections for 'Filesystems' and 'LUNs' (4 Total). A table lists the file systems with their names, sizes, and mount points:

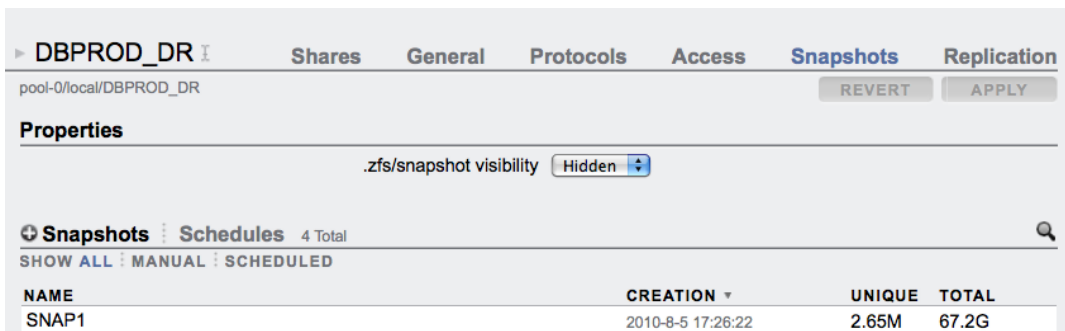
NAME ▲	SIZE	MOUNTPOINT
alerts	58.8M	/export/db_standby/alerts
archive	6.82G	/export/db_standby/archive
data	736G	/export/db_standby/data
logs	10.2G	/export/db_standby/logs

1. GUIで「DBPROD_DR」を選択し、「Snapshots」をクリックします。
2. 「(+Snapshots)」をクリックします。



3. スナップショット名として *SNAP1* を入力します。

スナップショット・リストに *SNAP1* が表示され、作成日時とサイズが表示されます。



これで、スナップショットの作成は完了です。

次のCLIスクリプト *take_snap.aksh* を使用すると、新しいスナップショットが取得されます。

注：プロジェクト内の各スナップショット名は一意である必要があります。

```
$ ssh -T root@aie-7420b < take_snap.aksh
```

```
$ cat take_snap.aksh
```

```
script
{
  var projName='DBPROD_DR';
  var snapName='SNAP1' ;
  var poolName='pool-0' ;
  printf("Snapshot the project..%n");
  run('cd /');
```

NAME	CREATION	UNIQUE	TOTAL	CLONES
SNAP1	2010-8-6 17:00:23	4.18M	63.5G	1

```

run('shares');
run('set pool=' + poolName);
run('select ' + projName);
run('snapshots snapshot ' + snapName);
}

```

スタンバイ・データベースの管理リカバリの再開

スナップショットが取得されたら、スタンバイ・データベース用の管理リカバリ・プロセスを再開します。

1. 次のスクリプトを使用して、管理リカバリ・プロセスを再開します。

```

$ sqlplus sys/oracle@PRODDDB
SQL> alter system set log_archive_dest_state_2=ENABLE;
$ sqlplus sys/oracle@PRODDB_DR
SQL> alter database recover managed standby database using current logfile
disconnect;

```

結果：スタンバイ・データベースはプライマリ本番データベースを追跡できるようになりました。

ステップ2. Sun ZFS Storage Applianceでのクローンの作成

プロジェクト・レベルのスナップショットは作成されましたが、クローンを作成できるのはファイル・システム・レベルのスナップショットからのみです。この構成では、data、logs、alerts、archiveのファイル・システムからクローンが作成されます。

クローン・プロジェクトの作成

スナップショット *SNAP1* からクローニングされたファイル・システムを格納するため、はじめに *CLONE1* という名前のプロジェクトを作成します。次に、スナップショット・プロジェクト *DB_PRODDR* に含まれる各ファイル・システムから取得された *SNAP1* スナップショットからクローンが作成され、*CLONE1* プロジェクト内に格納されます。

Sun ZFS Storage Applianceで次の処理を実行します。

1. *CLONE1* というクローン・プロジェクトがすでに存在する場合、このプロジェクトは削除されるか、または名前が変更されます。
2. *CLONE1* という名前のプロジェクトが作成されます。パフォーマンスを最適化するため、デフォルトのレコード・サイズは8Kに設定されます。読取りアクセス時間と更新時間はfalseに設定されます。

スナップショットを使用したファイル・システムのクローニング

1. プロジェクト「DBPROD_DR」を選択します。
2. プロジェクト内（Snapshotタブ内）のすべてのファイル・システムに対して、「clone」アイコンをクリックします。
3. ポップアップ画面のドロップダウン・ボックスからプロジェクト「CLONE1」を選択します。
4. ファイル・システム名として、元のファイル・システムと同じ名前を入力します。
5. 次にスクリーンショットに示すとおり、マウント・ポイントに/export/CLONE1/と入力します。

6. data、logs、archive、alertsの各ファイル・システムに対して、上記手順を繰り返します。
CLONE1内のすべてのファイル・システムに対して、クライアントから読取り/書込み操作を実行できるようになりました。

CLONE1		
Shares	General	Protocols
Access	Snapshots	Replication
pool-0/local/CLONE1		
Filesystems LUNs 4 Total		
NAME	SIZE	MOUNTPOINT
alerts	2.80M	/export/CLONE1/alerts
archive	18K	/export/CLONE1/archive
data	106M	/export/CLONE1/data
logs	126M	/export/CLONE1/logs

これで、Sun ZFS Storage Applianceでのクローニング処理が完了しました。スクリプト `create_clone.aksh` を実行すると、GUIと同じ手順をユーザー介入なしで実行できます。

注：クローン・プロジェクト名がすでに存在する場合、スクリプトを実行する前にこのプロジェクトを削除するか、または名前を変更してください。

```
$ ssh -T root@aie-7420b < create_clone.aksh
```

```
$ cat create_clone.aksh
script
{
    var projName='DBPROD_DR';
    var cloneProj='CLONE1' ;
    var snapName='SNAP1' ;
    var poolName='pool-0' ;

    printf("Creating the clone project¥n");
    run('cd /');
    run('shares');
    run ('set pool=' + poolName) ;
    run ('project ' + cloneProj);
    try {
    run ('set recordsize=8K');
    run ('set atime=false');
    run ('confirm commit');
    } catch (err) { printf("Done..¥n"); }
    printf("Cloning the shares..¥n");
    run('cd /');
    run('shares');
    run ('set pool=' + poolName) ;
    run('select ' + projName);
    var sharesList = list() ;
    for (var i = 0; i < sharesList.length; i++) {
        run('select '+ sharesList[i]);
        run('snapshots select ' + snapName );
        printf("Cloning the share %s ¥n", sharesList[i]);
        run('clone ' + cloneProj + ' ' + sharesList[i]);
        run('set mountpoint=/export/' + cloneProj + '/' + sharesList[i] );
        run('commit');
        run('cd ..');
        run('cd ..');
        run('cd ..');
    }
    printf("Cloning the project completed..¥n");
}
}
```

結果：アプライアンスでのファイル・システムのクローニング手順が完了しました。

ステップ3. クローニングしたスタンバイ・データベースからテスト、開発向けの読み取り/書き込み可能なデータベースへの変換

このステップでは、クローン・データベース（スタンバイ・データベースでもある）が読み取り/書き込み可能なデータベースに変換されます。このデータベースは後から、テスト、開発、QA目的で使用されます。ここでは、スタンバイ・データベースとクローン・データベースが同じホスト上に存在する場合と異なるホスト上に存在する場合の構成手順を提供します。

クローン・データベースのファイル・システムのマウント

スタンバイ・データベースとクローン・データベースが同じデータベース・サーバー上に配置されている場合

1. `/oradata/CLONE1/*`ディレクトリ内に、NFSファイル・システムをマウントします。
2. 続けて、クローン・データベースをアクティブ化します。

スタンバイ・データベースとクローン・データベースが異なるデータベース・サーバー上に配置されている場合

1. `/oradata/CLONE1/*`ディレクトリ内に、NFSファイル・システムをマウントします。
2. スタンバイREDOログ（`/oradata/stby/logs`内）は、引き続きクローン・データベースの制御ファイルから参照されます。ソフト・リンクが作成されます。

```
# mkdir -p /oradata/stby >/dev/null
# ln -s /oradata/CLONE1/logs /oradata/stby/logs
```

3. 続けて、クローン・データベースをアクティブ化します。

クローニングされたスタンバイ・データベースから読み取り/書き込み可能なデータベースへの変換

1. クローニングされたデータベース・サーバーで、`CLONE1`インスタンスをマウント状態で起動します。

```
SQL> startup mount
pfile=/export/home/oracle/products/11.2.0/db/dbs/initCLONE1.ora
```

制御ファイル内のデータファイルは、スタンバイ・データベースのロケーションを指しています。

2. 次のSQLコマンドを使用してデータファイルの名前のみを変更し、`/oradata/stby/data/*`から`/oradata/CLONE1/data/*`を指すように設定します。

```
SQL>ALTER DATABASE RENAME FILE '/oradata/stby/data/data00.dbf' TO
'/oradata/CLONE2/data/data00.dbf';
```

```
SQL>ALTER DATABASE RENAME FILE '/oradata/stby/data/data01.dbf' TO
'/oradata/CLONE2/data/data01.dbf';
```

・・・その他

3. スタンバイ・データベースREDOログを削除します（クリーンアップのため）。これらのSRLは依然として、制御ファイル内で`stby`ロケーションを指していることに注意します。

```
SQL>alter database drop standby logfile '/oradata/stby/logs/stby-1.log';
```



```
SQL>alter database drop standby logfile '/oradata/stby/logs/stby-2.log';
SQL>alter database drop standby logfile '/oradata/stby/logs/stby-3.log';
```

4. クローニングされたスタンバイ・データベースを、プライマリ・データベースに対してアクティブ化します。次に、データベースを停止し、再起動します。

```
SQL> alter database activate standby database ;
SQL> shutdown immediate ;
SQL> startup
```

5. 元のスタンバイ・データベースが完全にそのままであり、プライマリの本番データベースを追跡していることを確認します。

```
-bash-3.00$ ./check_scn.sh
```

```
INSTANCE_NAME  Host Name
-----
PRODDB   aie-6300b

DATABASE_ROLE  CURRENT_SCN  PROTECTION_MODE
-----
PRIMARY        6119051  MAXIMUM PERFORMANCE
```

```
=====

INSTANCE_NAME  Host Name
-----
PRODDB_DR   aie-6300c

DATABASE_ROLE  CURRENT_SCN  PROTECTION_MODE
-----
PHYSICAL STANDBY    6119049  MAXIMUM PERFORMANCE
```

```
=====

INSTANCE_NAME  Host Name
-----
CLONE1    aie-6300c

DATABASE_ROLE  CURRENT_SCN  PROTECTION_MODE
-----
PHYSICAL STANDBY    5674818  MAXIMUM PERFORMANCE
```

結果 : クローン・データベースを、テスト、開発、品質保証を目的として使用できるようになりました。

配置シナリオの例

スタンバイ・データベースとクローン・データベースはさまざまな組み合わせで配置できます。こ

ここでは、そのうちの数例について説明します。それぞれのシナリオに、次の条件があてはまります。

- 1つ以上のスタンバイ・データベースがあらかじめ構築されており、Oracle Data Guardでのプライマリ・フェイルオーバー・ターゲットとして構成されています。
- インフラストラクチャにSun ZFS Storage Applianceが統合されています。
- Sun ZFS Storage Applianceに格納された新しいスタンバイ・データベースは、クローニングを目的としてのみ使用できます。

データベース・クローニング・インフラストラクチャは、大きく分けて次のいずれかのカテゴリに分類されます。

1. 新しいスタンバイ・データベース・インスタンスとクローン・データベース・インスタンスが、同じホスト上に配置されている。
2. 新しいスタンバイ・データベース・インスタンスとクローン・データベース・インスタンスが、異なるホスト上に配置されている。

注：すべての構成において、Sun ZFS Storage Applianceでのスナップショットとクローニングの実行手順は同じです。

配置シナリオ1：すべてのスタンバイ・データベースおよびクローンを1つのホストに配置
 ユースケース：既存のスタンバイ・データベース・サーバーを、テストおよび開発データベース向けに構成します。

図3に示すとおり、インフラストラクチャにはSun ZFS Storage Applianceが追加されています。また、スタンバイ・サイトには、NFSベースのスタンバイ・データベースとクローン・データベースに対する適切な初期化パラメータが設定されています。スナップショットとクローニング処理が完了すると、このホストは3つのデータベース・インスタンス（2つのスタンバイと1つのクローン）にサービスを提供することになります。

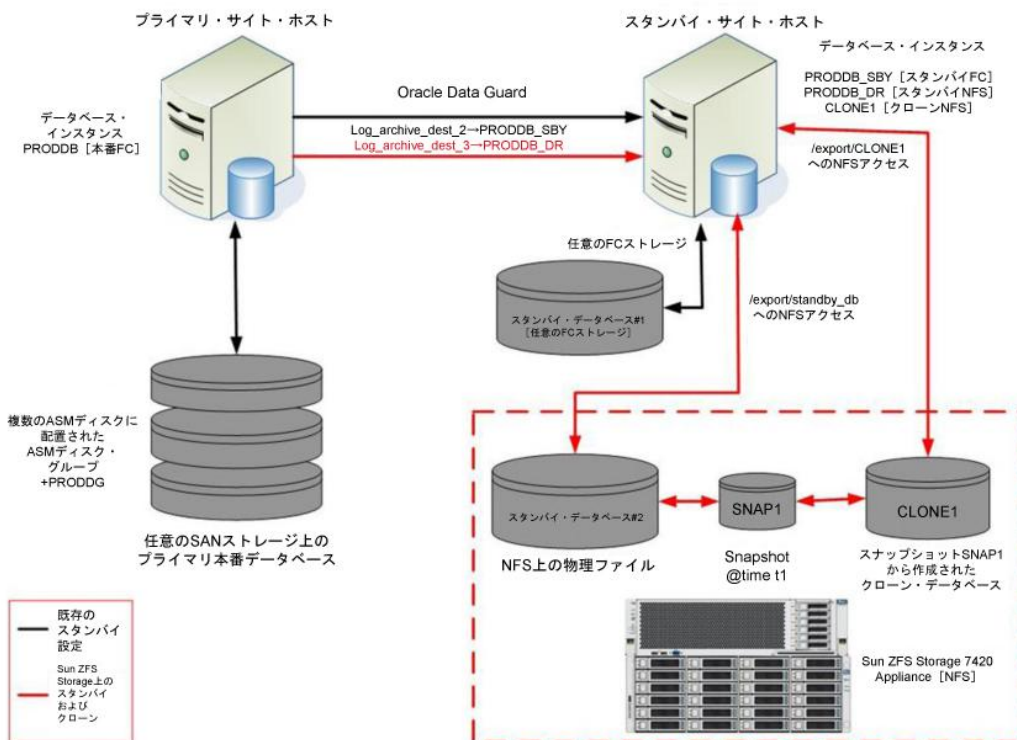


図3：シナリオ1- 同一ホスト構成による、新しいスタンバイ・データベースとクローン・データベースのホスティング

配置シナリオ2：クローン・データベースを異なるホストに配置

ユースケース：スタンバイ・データベースのホストには、もう1つのインスタンスに対応できるリソースがあります。開発およびテストのデータベースは別のサーバーにホストされます。

図4に示すとおり、既存のスタンバイ・サイトは、NSFベースのスタンバイ・データベースにもサービスを提供するように構成されています。また、1つ以上のクローン・データベース・インスタンスをホストするため、専用のホストが構成されています。この構成では、テスト、開発、およびQA処理がスタンバイ・データベースの妨げになることはありません。

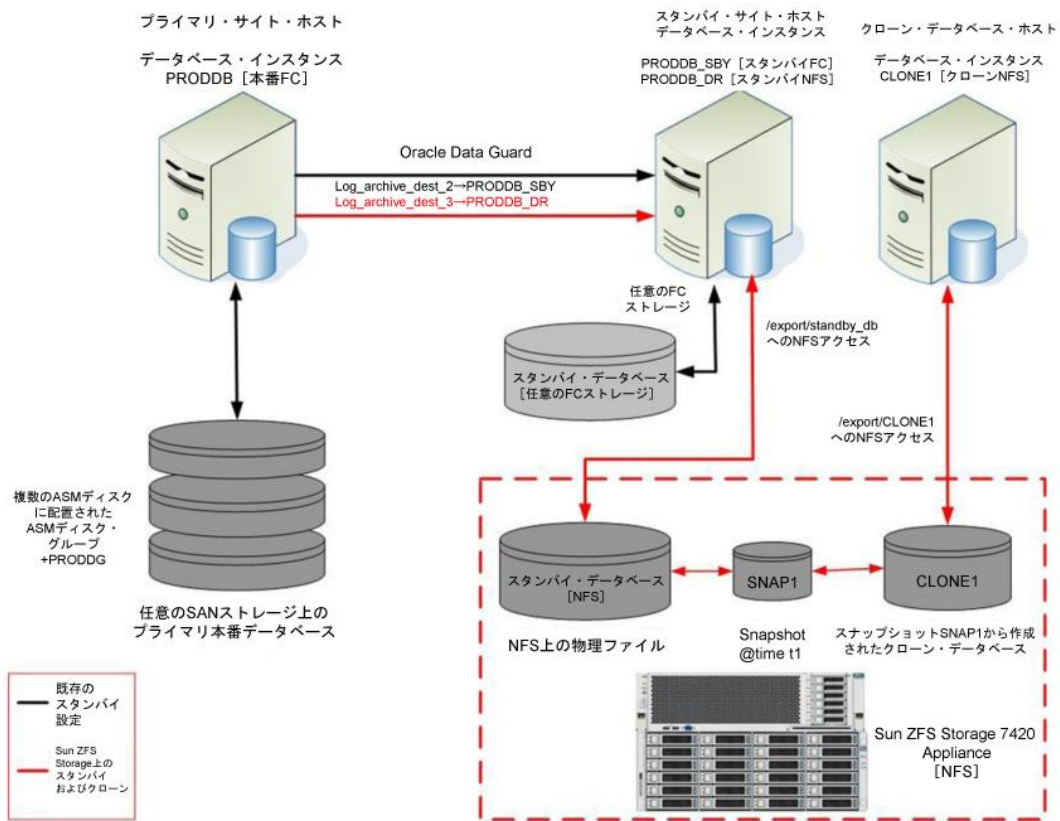


図4：シナリオ2 - クローン・データベースを異なるホストに配置

配置シナリオ3：スタンバイおよびクローンを新規ホストに配置

ユースケース：既存のスタンバイ・データベースのセットアップを、テストおよび開発用に分散することはできません。

図5に示すとおり、開発およびテスト目的のNFSベースのスタンバイ・データベース（*PRODDB_DR*）とクローン・データベース（*CLONE1*、*CLONE2*）に対して、独立したホストが配置されています。クローン・データベースは、異なるタイミングでSun ZFS Storage Applianceから取得された2つのスナップショット（*SNAP1*と*SNAP2*）をベースに作成されています。この新しいホストでは、*PRODDB_DR*、*CLONE1*、*CLONE2*という3つのインスタンスすべてで*ORACLE_HOME*が共有されています。

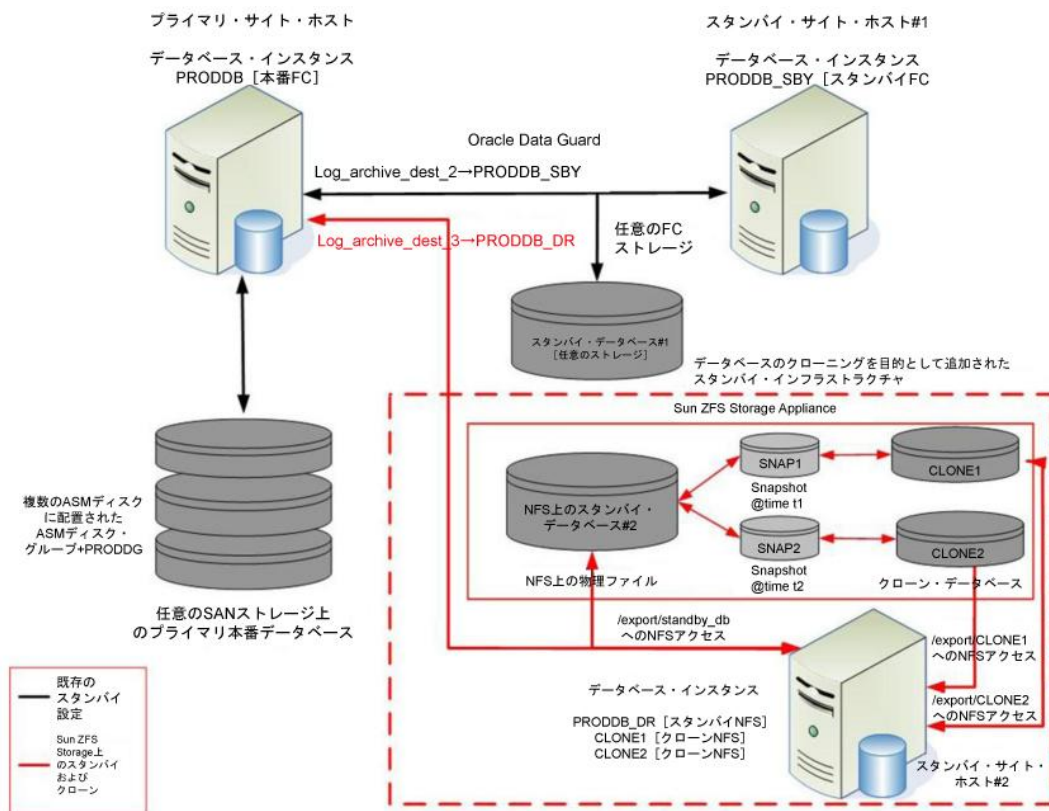


図5：シナリオ3 - スタンバイおよびクローン用の新規ホスト

Sun ZFS Storage Applianceを使用したデータベース・クローニングの利点

オラクルによるテストおよび検証済みのソリューション

本書で説明した手順は、オラクルによってテストと検証が実施されている手順です。ソリューションの説明とスクリプト・サンプルが提供されているため、テストおよび開発サイクルが迅速になり、配置までの期間が短縮されます。

配置と管理の簡素化

マルチプロトコル対応のSun ZFS Storage Applianceは、あらゆるインフラストラクチャに適合します。ユーザー・インターフェースを使用して、直感的で便利な方法でアプライアンスを管理できます。データベースのクローニング・ソリューション全体をスクリプト化し、繰り返し実行できるため、クローニング処理に必要な時間とリソースが軽減されます。

バックアップ、ビジネス継続性、および障害時リカバリの効率化

このアプライアンスには、レプリケーション、スナップショット、クローニングなどのあらゆるデータ・サービスが含まれています。また、追加コストなしで、任意の機能やプロトコルを有効化できます。これらすべての機能を特定のニーズに合わせて組み合わせることで、バックアップの簡易化（各種主要バックアップ・アプリケーションで認定済み）、ビジネス継続性の確保によるほぼ即時のリストア、リモート・レプリケーションによる障害時リカバリが実現されます。

Oracle Maximum Availability Architectureのベスト・プラクティスでは、Oracle Databaseに対する障害時リカバリの保護ソリューションとしてOracle Data Guardを利用することが推奨されています。Sun ZFS Storage Applianceは、Oracle Databaseの外部にあるファイル・システム・データに対してレプリケーションと障害時リカバリを提供することで、Oracle Data Guardを補完します。

無制限のデータベース・クローニングと効率的な領域使用

多数の従来型の手法を使用して、本番のOracleデータベースを複製できます。従来型の手法では、セカンダリ・サイトにData Guardのフル・コピーまたはポイント・イン・タイム・コピー（もしくはクローン）を作成します。しかし、これらの方法には時間がかかり、大量のストレージが必要になります。たとえば、1TBのデータベースに対して6つのクローンを作成するには、6倍近いストレージ容量が必要になります。

スナップショットが取得される際、当初は領域が割り当てられません。同様に、スナップショットからクローンが作成された場合、変更が実施された場合のみ新規ブロックが書き込まれます。このため、クローンにアクセスすると、スタンバイ・データベースのデータ・ブロックが参照される場合があります。これにより、多数のクローンを配置した場合に効率的に領域を利用できます。1つのスナップショットから複数のクローンを作成でき、作成したクローンはいつでも破棄できます。クローン・インスタンスに問題がある場合は、いったん破棄してから作成し直すことができるため、大きなメリットがあります。このアプライアンスでは無制限のスナップショットとクローンをサポートしているため、配置の柔軟性が最大化されます。

ハイブリッド・ストレージ・プールによる応答時間の短縮

Sun ZFS Storage Applianceのハイブリッド・ストレージ・プール（HSP）機能を利用すると、DRAM、Flash、およびハード・ディスクといったメディアをデータベース・データに利用することで、クライアントのアクセスを高速化できます。待機時間の影響を受けやすいデータベース書込みは、書込みが最適化されたFlashで処理されるため、1ミリ秒未満の高速な応答が可能になります。読取り用に最適化されたフラッシュがストレージ・キャッシュの第2層として機能し、最近アクセスされたブロックや頻繁にアクセスされるブロックを保存します。頻繁にアクセスされないデータは、大容量ディスクに格納されます。スナップショットやクローン・コピーに関しては、複数のクローンが同じブロックにアクセスする場合はDRAMまたはFlashからデータが提供されるため、応答時間が短縮されます。図6に、HSPモデルに含まれる各種コンポーネントを示します。

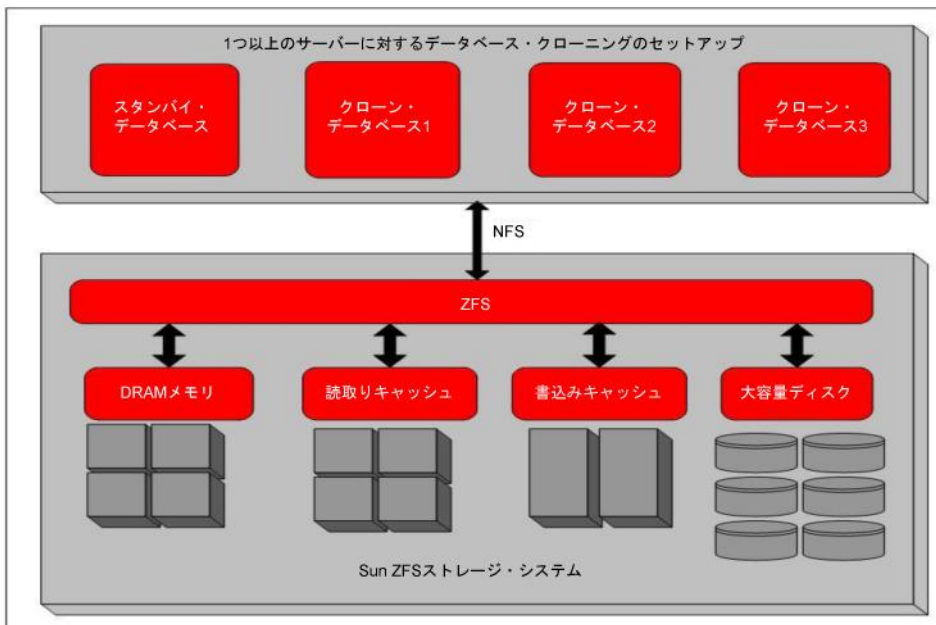
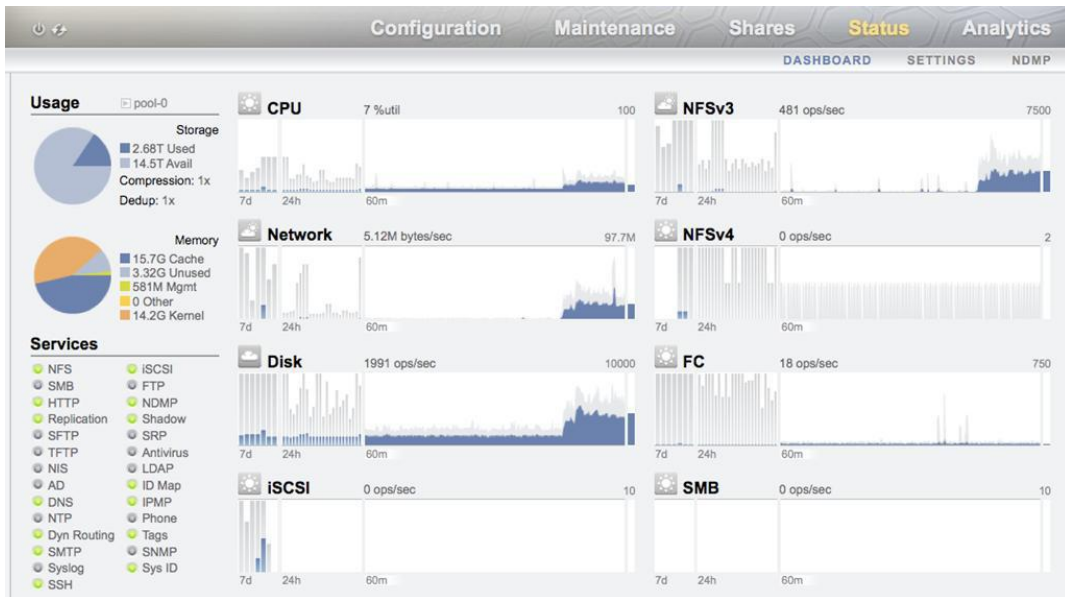


図6: ハイブリッド・ストレージ・プールを使用したデータベース・クローニング

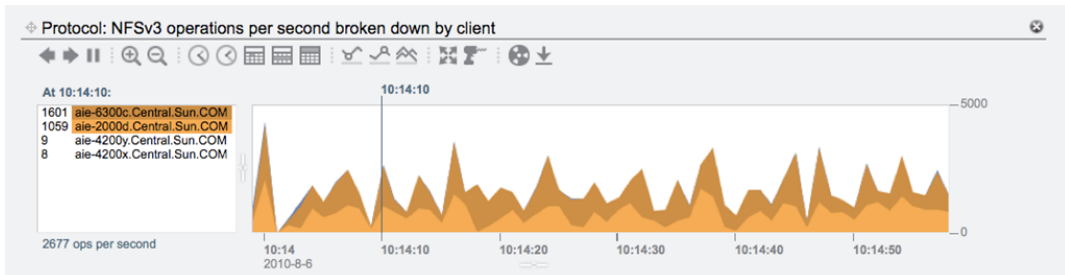
Analyticsによる解決と計画の迅速化

Sun ZFS Storage ApplianceのAnalytics機能は、ストレージ・アプライアンスの各種コンポーネントのパフォーマンスを分かりやすいグラフィカル表示で提供します。次に例を示します。

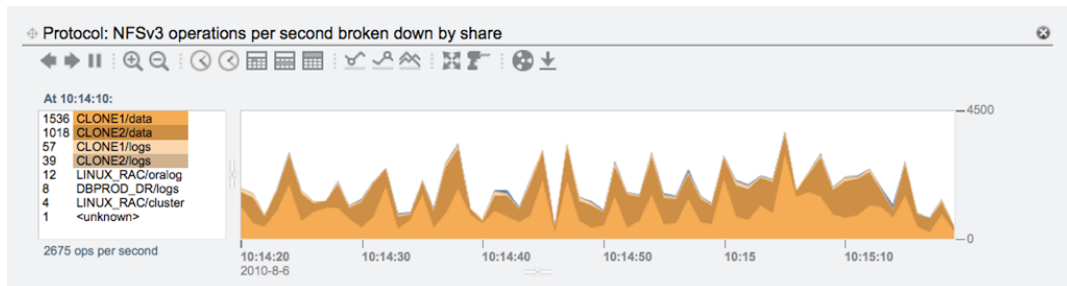
- ダッシュボードでは、システムの現状を示す概要ビューを提供します。次のスクリーンショットは、各種プロトコルに対する領域使用率とアクティビティを示しています。



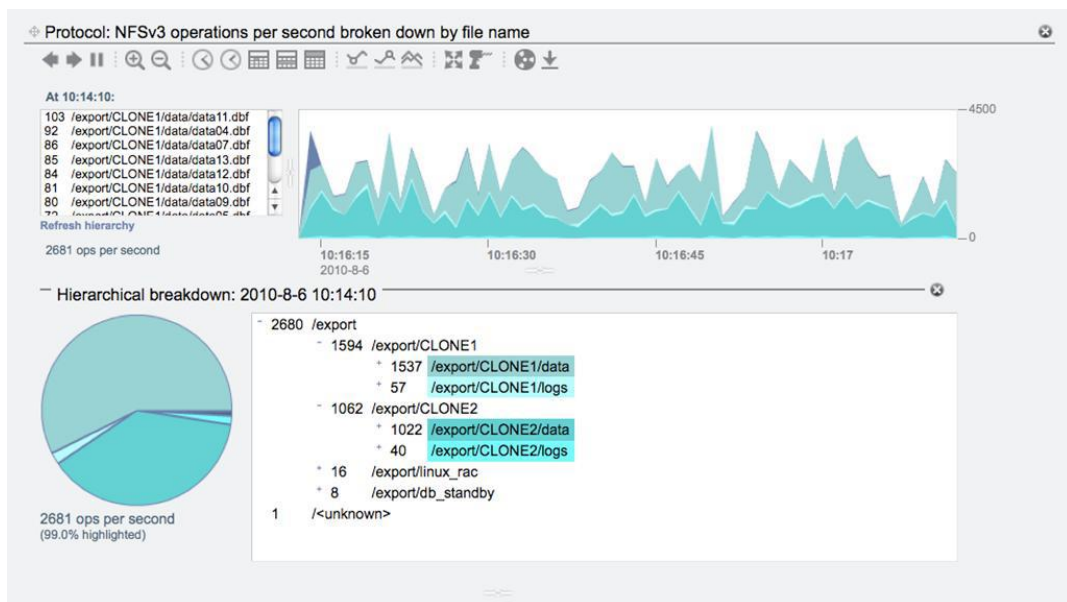
- 次のAnalyticsスクリーンショットは、NFSv3プロトコルを使用してシステムにアクセスしているクライアントを示しています。



- 次のAnalyticsスクリーンショットは、アクセスされているファイル・システムを示しています。



- 次のAnalyticsスクリーンショットは、各種のファイル・システムに対する詳しいアクセス・パターンを示しています。



Analytics機能が有用なケースは、次のとおりです。

- アプライアンス全体の状態把握
- アプライアンスのIOPS、待機時間、スループットの監視
- クライアントのアクセス・パターンの把握による品質保証契約（SLA）への対応
- 容量計画の要件
- クライアントとアプライアンスのパフォーマンス・データの比較による問題の特定と解決
- 履歴データのエクスポートによる解析など

結論

Oracle Data GuardとSun ZFS Storage Applianceを組み合わせると、データベース・クローニングに最適なソリューションが提供されます。このアプライアンスには、容易な管理を実現するためのユーザー・フレンドリーなインターフェース、ビジネスの継続性と障害時リカバリを目的としたデータ・サービス一式、あらゆるインフラストラクチャに対応したマルチプロトコル・サポート、監視と解決に役立つAnalytics、テスト、開発、QA作業の応答時間を短縮するハイブリッド・ストレージ・プールが含まれています。本書で説明したスクリプトやワークフローをOracle Data Guardと組み合わせて使用すると、クローニングの手順全体を自動化して、効率的な繰り返し実行を行うことができます。

スナップショットとクローニングの数に制限がないため、本番データベースに影響を与えることなく、さまざまな目的で多数のデータベース・インスタンスを同時に起動できます。これらすべてから、Sun ZFS Storage Applianceはデータベース・クローニングに適した実用的なソリューション・プラットフォームであると言えます。

付録

サンプル・スクリプト

スクリプト内にはハードコードされた値が含まれる点に注意してください。これらには、データベース・サーバーのホスト名、ORACLE_SID、ストレージのホスト名、ストレージ・プール名、プロジェクト名、スナップショット名、クローン・プロジェクト名、マウント・ポイントなどが含まれます。ご使用の環境に合わせて値を変更してください。

スクリプト名/説明	スクリプト
シェル・スクリプト	
<p><i>initiate_clone.sh</i></p> <p>このスクリプトはクローニング処理全体を実行します。スタンバイ・データベースを一時停止してスナップショットを作成し、スナップショットからクローンを作成してクローン・プロジェクト内に移動します。</p> <p>次に、クローニングされたファイル・システムをクライアントにマウントし、データベースをマウントして、スタンバイ・データベースからテスト/開発/QA用の読み取り/書き込み可能なデータベースへ変換します。</p>	<pre>echo "Suspending the automatic recovery of the standby database before taking the snapshot.." ./before_snap.sh echo "Doing the snapping and cloning process in the Sun ZFS Storage.." echo "Enter password for the root user for the S7000 storage at the prompt.." ssh -T root@aie-7420b.central.sun.com < s7000_clone_project.aksh echo "Resuming the automatic recovery of the standby database.." ./after_snap.sh echo "Unmounting and remounting the clone mount points in the DB server" echo "Enter password for the root user for this system at the prompt.." ssh root@aie-6300c/export/orahome/DBCLONE_1/scripts/mount_clone.sh \$1 echo "Converting the clone standby to a clone primary DB and open for read/write" #echo "Press <Enter> to continue, <Ctrl+C> to abort.." #read c ORACLE_SID=\$1; export ORACLE_SID sqlplus -S / as sysdba @clonedb echo "Cloning the database is complete !!. Press <Enter> to continue..<Ctrl+C> to abort.." read b echo "Verifying the content.." ./verify_db.sh PRODDb ./verify_db.sh \$1 echo "Cloning process complete !!" ./check_scn.sh</pre>
<p><i>before_snap.sh</i></p> <p>このプロシージャは、スナップショットを開始する前にスタンバイ・サイトで実行します。</p>	<pre>ORACLE_SID=PRODDb_DR; export ORACLE_SID sqlplus -S / as sysdba << EOF select DATABASE_ROLE,current_scn,protection_mode from v¥\$database; alter database recover managed standby database cancel; alter database recover managed standby database disconnect; alter database recover managed standby database cancel; EOF</pre>
<p><i>after_snap.sh</i></p> <p>このプロシージャはスナップショットの取得後に実行します。</p>	<pre>ORACLE_SID=PRODDb_DR; export ORACLE_SID sqlplus -S / as sysdba << EOF alter database recover managed standby database using current logfile disconnect; EOF</pre>
<p><i>check_scn.sh</i></p> <p>このスクリプトは、プライマリ・データベース、スタンバイ・データベース、クローン・データベースに対して、ステータスと現在のSCNを検証します。</p>	<pre>sqlplus -S sys/oracle@PRODDb as sysdba << EOF1 select instance_name, substr(host_name,1,30) "Host Name" from v¥\$instance; select DATABASE_ROLE,current_scn,protection_mode from v¥\$database; exit EOF1 echo "===== ORACLE_SID=PRODDb_DR; export ORACLE_SID sqlplus -S / as sysdba << EOF2 select instance_name, substr(host_name,1,30) "Host Name" from v¥\$instance; select DATABASE_ROLE,current_scn,protection_mode from v¥\$database; exit EOF2 echo "===== ORACLE_SID=CLONE1; export ORACLE_SID sqlplus -S / as sysdba << EOF3 select instance_name, substr(host_name,1,30) "Host Name" from v¥\$instance; select DATABASE_ROLE,current_scn,protection_mode from v¥\$database; exit EOF3 echo "=====</pre>

<pre>mount_clone.sh</pre> <p>マウント・ポイントを作成し、クローン・データベースのシェアをマウントします。</p>	<pre>#!/bin/sh echo "Creating directories and mounting the clone \$1" shares="data logs archive alerts" for i in \$shares; do echo "Processing the share \$i" /usr/bin/mkdir -p /oradata/\$1/\$i > /dev/null /usr/sbin/umount /oradata/\$1/\$i > /dev/null /usr/sbin/mount -F nfs -o rw,bg,hard,nointr,rsize=32768,wsz=32768,proto=tcp,vers=3,actimeo=0 aie-7420b:/export/\$1/\$i /oradata/\$1/\$i done df -k grep \$1</pre>
<p>スタンバイ・データベースのSQLスクリプト</p>	
<pre>mount_standbydb.sql</pre> <p>このSQLスクリプトはスタンバイ・データベースをマウントし、現在のログ・ファイルを使用して管理リカバリを有効化します。</p>	<pre>startup nomount pfile=/oracle/products/11.2.0/db/dbs/initPRODDB_DR.ora alter database mount standby database; alter database recover managed standby database using current logfile disconnect; recover managed standby database cancel; alter database open read only; alter database recover managed standby database disconnect using current logfile; select instance_name, host_name from v\$instance; select DATABASE_ROLE,current_scn,protection_mode from v\$database; exit</pre>
<pre>Clonedb.sql</pre> <p>ORACLE_SID=CLONE1 このSQLスクリプトはクローニングされたスタンバイ・データベースをマウントし、データファイルの名前を変更してから、プライマリ・データベースに対し、スタンバイをテスト/開発/QA用にアクティブ化します。</p>	<pre>shutdown immediate startup mount pfile=/oracle/products/11.2.0/db/dbs/initCLONE1.ora @rename_files select name from v\$datafile; alter database drop standby logfile '/oradata/stby/logs/stby-1.log'; alter database drop standby logfile '/oradata/stby/logs/stby-2.log'; alter database drop standby logfile '/oradata/stby/logs/stby-3.log'; select member from v\$logfile; # alter database activate standby database ; # shutdown immediate startup pfile=/oracle/products/11.2.0/db/dbs/initCLONE1.ora select instance_name, substr(host_name,1,25) "Host Name",status,instance_role from v\$instance; exit</pre>
<p>Sun ZFS Storage Applianceのスクリプト</p>	
<pre>take_snap.aksh</pre> <p>このスクリプトは、snapNameという名前のスナップショットを作成します。</p>	<pre>script { var projName='DBPROD_DR'; var snapName='SNAP1'; var poolName='pool-0'; printf("Snapshotting the project..%n"); run('cd /'); run('shares'); run('set pool=' + poolName); run('select ' + projName); run('snapshots snapshot ' + snapName); }</pre>

<p><i>create_clone.aksh</i></p> <p>このスクリプトはクローン・プロジェクトを作成して、このプロジェクト内に含まれるすべてのプロジェクト・シェアのスナップショットを割り当てます。</p>	<pre>script { var projName='DBPROD_DR'; var cloneProj='CLONE1'; var snapName='SNAP1'; var poolName='pool-0'; printf("Creating the clone project\n"); run('cd /'); run('shares'); run ('set pool=' + poolName); run ('project ' + cloneProj); try { run ('set recordsize=8K'); run ('set atime=false'); run ('confirm commit'); } catch (err) { printf("Done..\n"); } printf("Cloning the shares..\n"); run('cd /'); run('shares'); run ('set pool=' + poolName); run('select ' + projName); var sharesList = list(); for (var i = 0; i < sharesList.length; i++) { run('select '+ sharesList[i]); run('snapshots select ' + snapName); printf("Cloning the share %s \n", sharesList[i]); run('clone ' + cloneProj + ' ' + sharesList[i]); run('set mountpoint=/export/' + cloneProj + '/' + sharesList[i]); run('commit'); run('cd ..'); run('cd ..'); run('cd ..'); } printf("Cloning the project completed..\n"); }</pre>
---	---

用語集

GIFS

Common Internet File Systemプロトコル。別名Server Message Block (SMB : サーバー・メッセージ・ブロック) プロトコル。

クローン

このアプライアンスは、無制限の数のクローンをサポートしています。クローンは、即時作成される読取り/書き込み可能なスナップショットのコピーです。1つのスナップショットから、1つまたは複数のクローンが作成されます。これらのクローンは、通常のファイル・システムとしてユーザーに表示されません。クローンに対しては、クローンからのスナップショットの取得を含めたすべての通常操作を実行できます。クローンは通常、テスト、開発、QA、およびバックアップ環境で使用されます。

スナップショットと同様に、クローンの作成時には、領域の割当ては行われません。クローンの読取りは、ベース・ファイル・システムのブロックで処理されます。クローン内のブロックが変更された場合にのみ、変更されたブロックが割り当てられます。スナップショットとクローン間で領域が共有され、1つのスナップショットに複数のクローンを格納できるため、スナップショットを破棄するとアクティブなクローンも破棄されます。

クライアントからは、クローンのファイル・システムが独立したファイル・システムであるかのように見えます。クローンにアクセスするための特別な要件は必要ありません。

NDMP

Network Data Management Protocol。

NFS

Network File Systemプロトコル。

プロジェクト

プロジェクトでは、シェアを管理する共通管理コントロール・ポイントを定義します。プロジェクト内のシェアはすべて共通設定を共有可能で、割当てはシェア・レベルの他、プロジェクト・レベルで実施できます。プロジェクトはまた、論理的に関連したシェアをグループ化する目的でのみ使用されるため、共通属性（蓄積領域など）へのシングル・ポイント・アクセスが可能です。すべてのファイル・システムとLUNは、プロジェクトとしてグループ化されます。通常、すべてのアプリケーションに固有のプロジェクトが作成されます。プロジェクトは“コンシステンシ・グループ”としてもみなされます。

リモート・レプリケーション

ターゲットに対してデータが非同期にレプリケートされ、障害時リカバリの目的で使用されます。レプリケーションは設定を通じて、連続的に実行したり、ユーザーが定義したスケジュールに従って実行したり、オンデマンドで実行したりすることができます。Sun ZFS Storage Applianceでは、1つのアプリケーションから1つまたは複数のアプライアンスに対するリモート・レプリケーションがサポートされています。

シェア

シェアとは、サポートされているデータ・プロトコルを使用してアプライアンスのクライアントに

エクスポートされるファイル・システムやLUNです。シェアは、プロジェクト内に作成されます。ファイル・システムはファイルベースの階層をエクスポートし、CIFS、NFS、HTTP/WebDav、FTPを使用してアクセスされます。LUNはブロックベースのボリュームをエクスポートし、iSCSIを使用してアクセスされます。

プロジェクト/シェアは、プール内のシェアに対する一意の識別子です。複数のプロジェクトに同一名を持つ複数のシェアを格納することはできますが、1つのプロジェクトに同一名を持つ複数のシェアを格納することはできません。1つのプロジェクトにファイル・システムとLUNの両方を格納して、同じ名前空間を共有できます。

スナップショット

Sun ZFS Storage Applianceには、無制限のスナップショット機能が含まれています。スナップショットはファイル・システムの読取り専用のポイント・イン・タイム・コピーであり、即時作成されますが、初期領域の割当ては行われません。ブロックは、ベース・ファイル・システムが変更された場合に割り当てられます (Copy-On-Write)。スナップショットは、手動で開始するか、特定の間隔で自動実行されるよう設定できます。このスナップショット・データへは、バックアップ目的で直接アクセスできます。

スナップショット・ブロックの読取りは、ベース・ファイル・システムのブロックで処理されます。ベース・ファイル・システムが変更されると、古いブロックはスナップショットによって参照され、新しく変更されたブロックはファイル・システムによって参照されます。

プロジェクト・スナップショットは、プロジェクト内のすべてのシェアでスナップショットを実行する処理と同じです。

スナップショットのロールバック

スナップショットが取得された時点でベース・ファイル・システムを戻すプロセス。ロールバック・プロセスは、スナップショットの取得時からロールバックの実行時まで発生したベース・ファイル・システムへのすべての変更を破棄します。これによって、データのリストア・プロセスは不要になります。

ストレージ・プール

ストレージ・プール (ボリューム・グループとほぼ同義) は、物理ディスク・セット上に作成されます。その後、ストレージ・プール上にファイル・システムが作成されます。使用可能な物理ディスク上に1つまたは複数のストレージ・プールが作成され、フラッシュ・ドライブが割り当てられます。ストレージ・プールは、ミラー、RAID-Z (シングル・パリティ)、RAID-Z2 (デュアル・パリティ) などの配置で構成されます。

リソース

- Oracle Maximum Availability Architecture Web site
<http://www.oracle.com/technetwork/jp/database/features/availability/maa-094615-ja.html>
- *Oracle Database High Availability Overview (Part #B14210)*
http://otndnld.oracle.co.jp/document/products/oracle11g/111/doc_dvd/server.111/E05765-03/toc.htm
- *Oracle Database High Availability Best Practices (Part B25159)*
http://otn.oracle.com/pls/db111/db111.to_toc?partno=b28282
- *Oracle Data Guard with Oracle Database 11g Release 2 Technical Information*
<http://www.oracle.com/technetwork/database/features/availability/dataguardoverview-098960.html>
- *Oracle Data Guard 11g Release 2 Technical White Paper*
<http://www.oracle.com/technetwork/jp/content/twp-dataguard-11gr2-134591-ja.pdf>
- *Sun ZFS Storage Appliance Documentation*
<http://wikis.sun.com/display/FishWorks/Documentation>
- *Deploying Oracle Databases on Sun Storage 7000 Unified Storage*
http://www.sun.com/bigadmin/features/articles/7000_oracle_deploy_2009q3.jsp
- *Configuring Sun Storage 7000 for Oracle Databases*
<http://www.oracle.com/technetwork/articles/systems-hardware-architecture/ss700-config-oracle-db-163902.pdf>
- *Backup and Recovery of Oracle Databases with Sun Storage 7000*
http://i.zdnet.com/whitepapers/OS_US_EN_WP_Snapshot%5B1%5D.pdf

ORACLE®

Sun ZFS Storage ApplianceとOracle
Data Guardを使用したOracle Databaseの
クローニング・ソリューション

2012年1月

著者 : Sridhar Ranganathan
共著者 : Joseph Meeks、Larry Carpenter、
David Krenik

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 © 2010, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracleは米国Oracle Corporation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。

0109

Hardware and Software, Engineered to Work Together