



Oracle ホワイトペーパー
2011 年 4 月

Oracle Recovery Manager と Sun ZFS Storage Appliance を使用した Oracle データベースのクローニング ソリューション

はじめに	2
Oracle データベースクローニングの概要	4
Oracle RMAN と Sun ZFS Storage Appliance を使用したデータベースクローニングの手順	5
実稼働データベースサーバーの構成とバックアップの実行	6
ストレージベースのスナップショットおよびクローニング操作の実行	8
クローンデータベースサーバーの構成	9
バックアップデータの更新	10
参照テストおよび開発用セットアップ	10
Sun ZFS Storage Appliance を使ったデータベースクローニングの利点	12
クローニングソリューションのサイジングに関する考慮事項	17
容量と保護	17
ネットワーク接続	18
最適化された OPS とスループット	18
一般的推奨事項	19
結論	20
付録 A. 関連文書	21
付録 B. Sun ZFS Storage Appliance について	22
付録 C. ストレージとホストの構成	25
付録 D. コマンドとスクリプト	29
Oracle RMAN 増分バックアップ用に実稼働データベースサーバーを構成する	29
レベル 0 とレベル 1 のバックアップを実行するための Oracle RMAN スクリプト	29
Sun ZFS Storage Appliance でのスナップショット操作の実行	31
Sun ZFS Storage Appliance でのクローニング操作の実行	31
クローンデータベースサーバーの構成	33
スナップショットをクローニングするための完全なスクリプト	38

はじめに

データベース管理者は、アプリケーションの開発、テスト、およびトレーニングを目的とした継続的的要求に対応するため、大規模でミッションクリティカルなデータベースを効率的に複製するという課題に直面しています。この課題は、大規模実稼働システムに関連する多数の開発およびテスト業務をサポートするために複数のデータベースのクローンが必要な場合に深刻化します。この複製されたデータベースは、最新のデータおよびデータ構造を使ってテストや開発を実施するためには、頻繁に更新する必要があります。

ミッションクリティカルなデータベースは、Oracle Automatic Storage Management (ASM) 統合ファイルシステムとボリュームマネージャーを使用して、ファイバチャネル (FC) プロトコルベースの SAN 環境に配備されることが多いです。しかし、開発、QA、およびトレーニング環境では、通常は実稼働データベースで求められるほどの厳しいサービスレベル契約 (SLA) は要求されません。したがって、これらの環境では高価な SAN インフラと同じインフラは不要なので、実稼働データベースにほとんどあるいは全く影響を与えずに、複数の書き込み可能なデータベースのコピーを迅速に作成するためのサポートは提供されない可能性があります。これらの環境の要件は、ネットワーク接続型ストレージ (NAS) を備えた Sun ZFS Storage Appliance を使用することで満たされます。これは低コストで最適な性能と使いやすさを提供するもので、既存のイーサネットインフラに容易にプラグインすることができます。

Oracle データベースのバックアップとリカバリを管理する Oracle Recovery Manager (Oracle RMAN) ユーティリティは、Oracle データベースを複製するためのいくつかの方法を提供します。本書では、Oracle RMAN の増分更新バックアップ機能を使って、SAN ベースの ASM データベースを Sun ZFS Storage Appliance に格納されているネットワークファイルシステム (NFS) プロトコルベースのデータベースにバックアップする方法について説明します。次にストレージアプライアンスのスナップショットとクローニング機能を使用して、Oracle RMAN バックアップを複製あるいはクローニングします。このドキュメントで説明するクローニングの手順は、実稼働サイトで実行されます。

リモートサイトや障害回復 (スタンバイ) サイトを使用するデータベース複製ソリューションについては、ホワイトペーパー『オラクルの Sun ZFS Storage Appliance と Oracle Data Guard を使用した Oracle データベースのクローニングソリューション』(以下の URL) を参照してください。
<http://www.oracle.com/technetwork/database/features/availability/maa-db-clone-szfssa-172997.pdf>

本書で説明するソリューションは、表 1 に示すバージョンのオペレーティングシステムおよび Oracle データベースを使用して実装することができます。

表 1. 対応バージョン

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) Red Hat Linux 4.x, 5.x (およびそれ以上) Oracle Enterprise Linux 4.x, 5.x (およびそれ以上)
Oracle データベース	10.2.0.1 (およびそれ以上) 実稼働: シングルインスタンス/RAC (ASM または ASM 以外) クローン: シングルインスタンス Oracle Recovery Manager 注: Oracle Exadata Hybrid Columnar Compression (EHCC) を有効にした環境での使用には適していません。*
実稼働データベースストレージ	どのストレージでも可
クローンデータベースストレージ	Sun ZFS Storage Appliance
プロトコル	NFSv3、NFSv4
サポートされている接続	1 GbE、10 GbE
クローンデータベースサーバー	別のホストからサービスされるクローンインスタンス (推奨) 実稼働ホストによってサービスされるクローンインスタンス

* このソリューションは、EHCC を有効にしたプライマリデータベース環境での使用には適していません。Oracle RMAN 変換データベースには圧縮形式のテーブルが含まれており、最初に解凍しなければ Exadata 以外のインスタンスによって直接アクセスすることができません。

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、技術営業担当者を対象読者としています。読者は Oracle RMAN ユーティリティーに精通していることが前提です。Sun ZFS Storage Appliance 上にファイルシステムを作成する詳細な手順については、付録 A 「関連文書」に記載されているドキュメントを参照してください。

Oracle データベースクローニングの概要

Oracle RMAN の増分更新バックアップ機能は、データベース内で行われた増分変更をバックアップデータベースにマージするパワフルなバックアップユーティリティであり、バックアップデータの 1 つの最新バージョンを複数の目的に使用できます。Oracle RMAN バックアップは AS COPY オプションを使用して実行されます。AS COPY オプションにより、Oracle RMAN は個別のデータファイルをコピーできるので、データベースの迅速なクローニングが可能になります。

最初にすべてのデータファイルの完全バックアップが NFS のマウントされたファイルシステムに作成されます。本書ではこれをマスターコピーローケーション (MCL) と呼びます。この完全バックアップを「レベル 0 のバックアップ」とします。続いて、レベル 1 の増分バックアップがレベル 0 のバックアップ作成後に行われた変更内容をコピーし、レベル 0 のバックアップにマージします。稼働データベースからアーカイブされたログも MCL にコピーされます。

図 1 にソリューションの概要を示します。Oracle RMAN の増分更新は定期的に MCL に送信され、MCL データファイルは更新された状態に保たれます。MCL ファイルシステムの 1 つ以上のスナップショット (S*) が、Sun ZFS Storage Appliance で実行されます。これらの各スナップショットから、1 つ以上のクローン (C*) が作成されます。

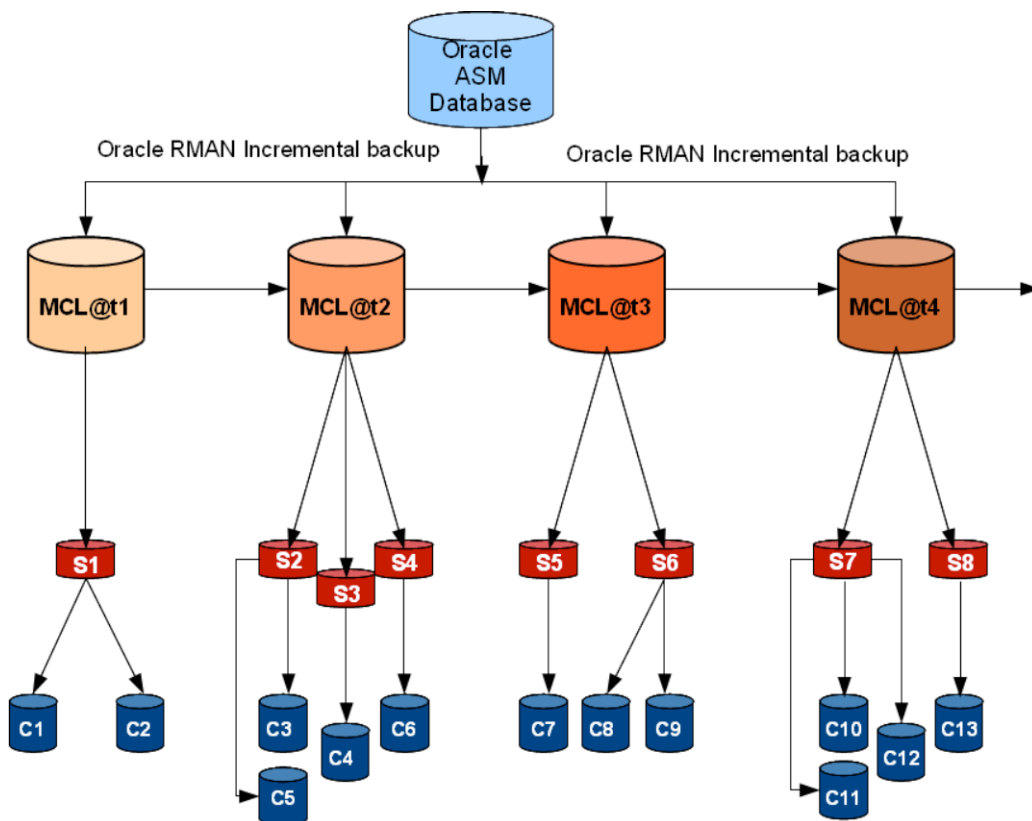


図 1. 増分バックアップを使用した Oracle データベースクローニング手順の概要

Oracle RMAN と Sun ZFS Storage Appliance を使用したデータベースクローニングの手順

クローンの Oracle データベース環境を作成する手順では、Oracle RMAN バックアップユーティリティと Sun ZFS Storage Appliance のスナップショットおよびクローニング機能を使用します。データベースのクローニング手順は、次の 4 つのステップからなります。

1. ASM ディスクグループに格納されている実稼働データベースサーバーから Oracle RMAN バックアップを構成して実行します。
2. ストレージベースのスナップショットおよびクローニングの操作を実行します。
3. クローンデータベースサーバーを構成し、データベースを開きます。
4. 実稼働データベースに対して行われた変更の増分更新バックアップを実行して、バックアップデータファイルの更新を続けます。

これらの手順については、以下の項で詳しく説明します。

図 2 に、実稼働データベースサーバー、1 つ以上のクローンデータベースサーバー、および Sun ZFS Storage Appliance をこのソリューションでどのように相互接続するかを示します。

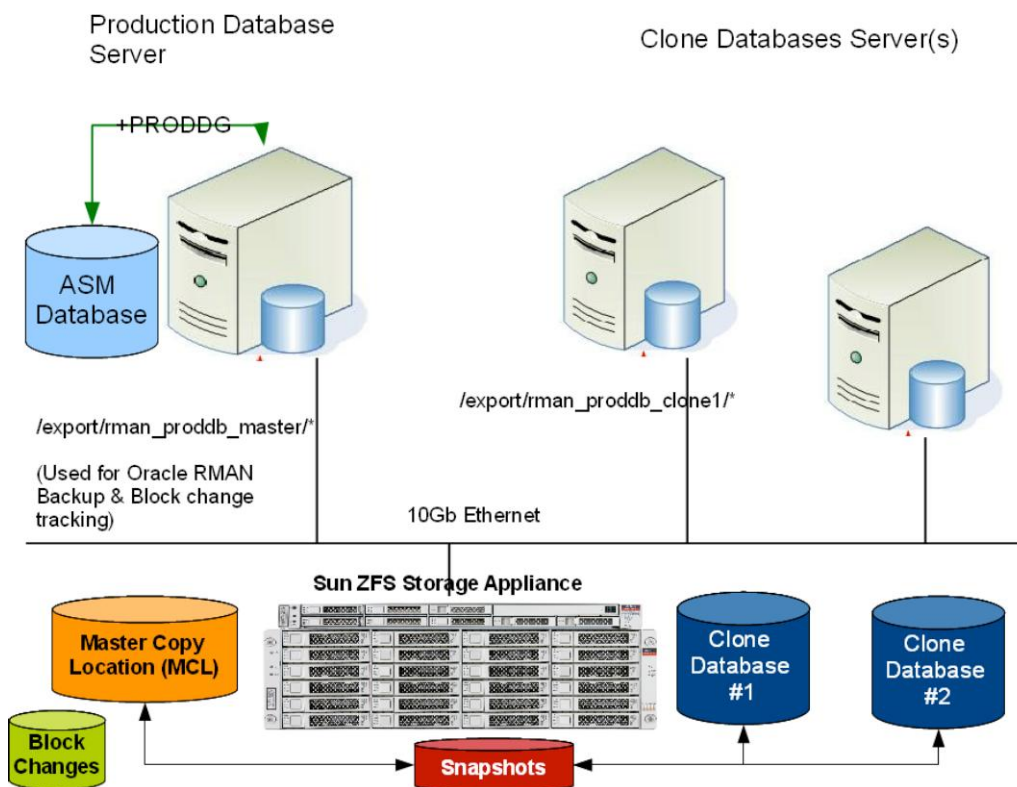


図 2. ソリューションのコンポーネントと相互接続

実稼働データベースサーバーの構成とバックアップの実行

このソリューションで、プライマリ実稼働システムは、インスタンス PRODDB がサービスする Oracle ASM データベースをベースとすることを前提とします。データファイル、再実行ログ、および制御ファイルはすべて、FC ストレージを使用して ASM ディスクグループ +PRODDG に格納されています。実稼働データベースのログアーカイブ形式は、%t_%s_%r.arc に設定されています。

MCL は Oracle RMAN が増分バックアップとマージ操作を実行するためだけに使用されます。MCL にあるデータファイルをインスタンスのどれかが直接使用することはありません。

データベースサーバーは、10 Gb イーサネットを介して Sun ZFS Storage Appliance に接続されます。ストレージプールは RAID-Z2 のレイアウトで構成されます。

データファイルのコピーとアーカイブログのコピーを格納するために、rman_prod_master というプロジェクトを作成します。このプロジェクトの下に、datafiles、archive、redologs、および alerts という 4 つのファイルシステムを作成します。redologs および alerts ファイルシステムは、スクリプトを使った高速自動クローニングを有効にするためだけに作成します。

プロジェクトレベルでのデフォルトのマウントポイントは、各ファイルシステムのマウントポイントが一意になるように /export/rman_proddb_master へと変更されます。ファイルシステム datafiles のレコードサイズはデフォルトの 128 KB から 8 KB に変更されます。すべての共有については、logbias プロパティが latency (logbias=latency) に設定されます。

ファイルシステム datafiles および archive のみが /oradata/rman_master/PRODDB ディレクトリ下の実稼働データベースサーバーにマウントされます。ディレクトリの所有権を oracle:dba に変更してください。

このセクションでは、実稼働データベースサーバーを構成し、最初の Oracle RMAN のレベル 0 バックアップを実行するための手順の概要を説明します。詳細な手順については、付録 D 「コマンドとスクリプト」に記載しています。

手順は次のとおりです。

1. 実稼働データベースのブロック変更トラッキング機能を有効にします。

実稼働データベースのブロックへの変更を追跡して、Oracle RMAN がファイル incremental.f を使用して行う増分更新バックアップを高速化します。このファイルは任意の場所に配置できます。この例では、MCL はブロック変更トラッキングファイルの格納に使用されます。

```
$ sqlplus / as sysdba
SQL > alter database enable block change tracking using file
'/oradata/rman_master/PRODDB/incremental.f' ;
```

ブロック変更トラッキングファイルの情報は、次の SQL コマンドを使って監視することができます。

```
SQL> select substr(filename,1,40) "FILENAME", status, bytes from
v$block_change_tracking ;
```

FILENAME	STATUS	BYTES
/oradata/rman_master/PRODDb/incremental.	ENABLED	11599872

2. **Sun ZFS Storage Appliance** 上に **MCL NFS** ファイルシステムを構成します。詳細については、付録 D 「Oracle RMAN 増分バックアップ用に実稼働データベースサーバーを構成する」を参照してください。
3. **AS COPY** オプションを使用して、**レベル 0 のデータベースバックアップを MCL ターゲットに対して実行します**。データベースファイルから MCL への完全コピーを、Oracle RMAN を使用して実行します。このコピーは、レベル 0 のバックアップとして機能します。この例では 8 つのチャネルを使用します。選択されるチャネルの数は、使用可能なネットワークとスループットに基づきます。NFS ストレージシステム内のファイル名が ASM ベースの実稼働データベースで使用されるファイル名 (/oradata/rman_master/%d/datafiles/%b) と同じになるように、形式 %b が使用されます。

Oracle RMAN バックアップが特にクローニングを目的に実行されたことを示すため、タグ `zfssa_clone` が使用されます。レベル 0 のバックアップが完了する前にレベル 1 のコピーを実行しようとする、Oracle RMAN はレベル 0 の完全バックアップが存在しないことを検出し、MCL の `datafiles` の場所に対して暗黙的なレベル 0 のバックアップを開始します。バックアップを実行するための Oracle RMAN コマンドの構文は次のとおりです。

```
backup incremental level 1
for recover of copy with tag 'zfssa_clone' database reuse;
recover copy of database with tag 'zfssa_clone';
```

この手順の実施に関する詳細は、付録 D の「レベル 0 とレベル 1 のバックアップを実行するための Oracle RMAN スクリプト」の項を参照してください。

4. アーカイブログを MCL にコピーします。最新の再実行ログをアーカイブした後、すべてのアーカイブログを MCL アーカイブの場所にコピーします。

```
BACKUP AS COPY skip inaccessible (archivelog all) ;
```

ここで、データベースの完全コピーをクローニングに使用できます。この手順の実施に関する詳細は、付録 D の「レベル 0 とレベル 1 のバックアップを実行するための Oracle RMAN スクリプト」の項を参照してください。

5. **Oracle RMAN のレベル 1 バックアップを定期的に実行し、MCL 上に存在するデータファイルのバックアップコピーに変更内容をマージします**。

ステップごとの詳細な手順については、付録 D 「Oracle RMAN 増分バックアップ用に実稼働データベースサーバーを構成する」を参照してください。

ストレージベースのスナップショットおよびクローニング操作の実行

データベースバックアップのスナップショットを実行し、スナップショットのクローンを 1 つ以上作成するには、以下の手順を実行します。その他の詳細については、付録 D 「コマンドとスクリプト」に記載しています。

1. クローンデータベースをホストするために、Sun ZFS Storage Appliance 内に別のプロジェクトを作成します。
2. その下のストレージアプライアンス内に MCL ファイルシステムがマウントされるプロジェクトのスナップショットを実行します。

スナップショットとは、そこからクローンが作成されるファイルシステムのその時点のイメージのコピーです。スナップショット操作はプロジェクトレベルで実行されるので、スナップショットはプロジェクトの下にあるすべてのファイルシステム上で一貫して実行されます。スナップショットは読み取り専用ですので、データベース操作に使用することはできません。

プロジェクト `rman_proddb_master` のスナップショット `snap_0` を作成すると、`rman_proddb_master` プロジェクト内のすべてのファイルシステムのスナップショットが作成されます。これは、ストレージアプライアンスのグラフィカルユーザインタフェースまたはコマンドラインインタフェースを使用して行うことができます。

この操作は、バックアップが実行されていないときに行うことをお勧めします。そのようにしないと、クローンファイルシステムに正常にデータベースを回復するための十分なデータがない場合があります。

詳細については、付録 D の「Sun ZFS Storage Appliance でのスナップショット操作の実行」の項を参照してください。

3. スナップショットをクローニングして、ファイルシステムをクローンデータベース用に作成されたプロジェクトの下に格納します。

クローンはストレージアプライアンス内で、ステップ 2 で作成したスナップショット `snap_0` から作成されます。クローンは、データベースによってアクセスできるスナップショットの読み書き可能なコピーです。各クローンファイルシステムは、アクセシビリティに関しては独立していますが、ベーススナップショットに依存しています。そのため、クローンは破棄して単一のスナップショットから再作成することが可能です。ただし、スナップショットを削除すると、そのスナップショットから作成されたすべてのクローンが破棄されます。本書の例では、`rman_proddb_clone1` というプロジェクトがデータベース PRODDB の最初のクローンを格納するために作成されます。

クローニング操作を実行するには、以下の条件を満たす必要があります。

- クローニングされる各プロジェクトには一意の名前があること。
- クローニングされるプロジェクトは、スナップショットと同じストレージプール内に作成すること。
- エクスポートされるマウントポイントがストレージアプライアンス内で一意であること。

この時点で、MCL データから作成したデータベースのクローンコピーをデータベースアクセスに使用できます。詳細な手順については、付録 D の「Sun ZFS Storage Appliance でのクローニング操作の実行」の項を参照してください。

4. 複数のスナップショットまたはクローンを作成するには、ステップ 1～3 を繰り返します。

クローンデータベースサーバーの構成

クローンデータベースサーバーの構成は、以下の手順で行います。詳細については、付録 D 「コマンドとスクリプト」に記載しています。

1. **Sun ZFS Storage Appliance から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。** 詳細については、付録 D の「クローンデータベースサーバーの構成」の項を参照してください。
2. アーカイブログを使用して、インスタンスの作成、制御ファイルの作成、およびリカバリを実行し、resetlogs オプションを使用してデータベースを開きます。

このサーバー上の Oracle データベースバイナリのバージョンは、実稼働データベースサーバー上のバージョンと同じです。Sun ZFS Storage Appliance は、10Gb イーサネット接続によってこのデータベースサーバーに接続されています。ストレージアプライアンスからクローニングされたファイルシステムは、データベースサーバーにマウントされます。

まず、datafiles、redologs、archive、alerts という 4 つのディレクトリを作成し、これらのディレクトリの下に NFS ファイルシステムをマウントします。その後、oracle ユーザーが次の手順を行います。

- oracle インスタンスの CLONE1 が initCLONE1.ora を使用して nomount 状態で開始されます。
- BACKUP TO TRACE コマンドを使用して、制御ファイルを生成するスクリプトを実稼働データベースから作成します。次に、SQL コマンドをクローンデータベース環境に合わせて変更します。
- 制御ファイル作成スクリプトを使用して、制御ファイルを新たなデータファイルの場所 /oradata/clone1/datafiles/ に作成します。
- RECOVER UNTIL CANCEL コマンドを使用して、データベースを回復させます。インスタンスが必要とするすべてのアーカイブログは最後のアーカイブログが到達するまで適用されます。このときリカバリプロセスを終了するには CANCEL を入力します。
- RESETLOGS オプションを使用してデータベースを開きます。制御ファイルの指定どおりに、/oradata/clone1/redologs の下にオンライン再実行ログが作成されます。
- 一時ファイルが作成されます。
- listener.ora や tnsnames.ora などのネットワーク構成ファイルを更新して、クライアントがデータベースに接続できるようにします。

これで、クローンデータベースは完全に機能するデータベースとなり、テスト、開発、QA、バックアップ、その他のあらゆる目的に使用できます。ステップごとの詳細な手順については、付録 D の「スナップショットをクローニングするための完全なスクリプト」の項を参照してください。

バックアップデータの更新

Oracle RMAN の増分更新は通常、スケジュールに従って実行されますが、必要に応じて実行することもできます。RMAN 増分更新コマンドを使用して、変更内容をコピーし、MCL データファイルにマージします。レベル 0 のバックアップに使用したタグ `zfssa_clone` を、その後のレベル 1 の更新に使用します。次に最新の再実行ログをアーカイブし、すべてのアーカイブログが MCL にバックアップされます。その他の詳細については、付録 D の「レベル 0 とレベル 1 のバックアップを実行するための Oracle RMAN スクリプト」の項に記載しています。

参照テストおよび開発用セットアップ

図 3 に示すテストおよび開発テストのセットアップは、Oracle データベースのクローニングソリューション配備の参照アーキテクチャーとして提供しています。この参照アーキテクチャーで実稼働データベースは、Oracle ASM と、FC ストレージから作成した Oracle ASM ディスクグループを使用して実装されています。

Oracle RMAN の増分更新バックアップを使って、データファイルとアーカイブログを MCL の場所にバックアップします。ブロック変更トラッキング機能を使用することで、MCL データファイルの増分更新がさらに高速化します。

データベースサーバーまたはクローンデータベースインスタンスのホスティングに使用されるサーバーは、クローンファイルシステムをマウントすることによって構成されます。1 つのサーバーを使って、複数のインスタンスや複数のクローンサーバーをホストできます。

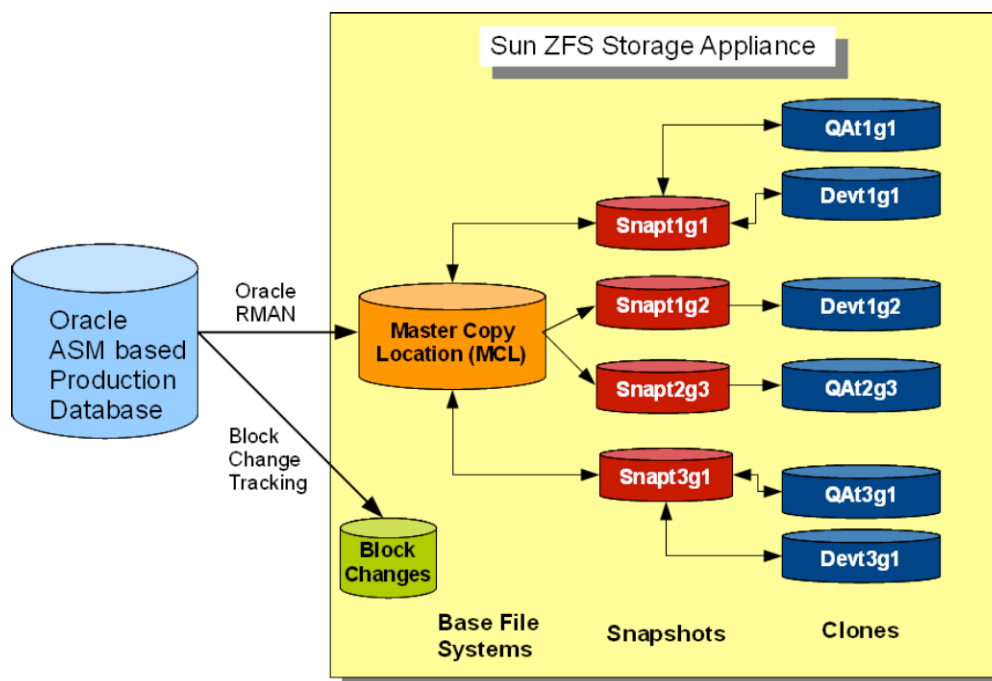


図 3. Oracle データベースのクローニングソリューションを配備するための参照アーキテクチャー

図 3 では、MCL に格納されたファイルシステムから 4 つのスナップショットが作成されています。それぞれのスナップショットは、別々のグループ (g*) が別々の時間 (t*) に、そのグループの特定のニーズを満たすために作成されたものです。

- Snapt1dg は、グループ 1 (g1) が レベル 0 バックアップ後の時間 t1 で作成したスナップショットです。
- Snapt1g2 は、グループ 2 (g2) が レベル 0 バックアップ後の時間 t1 で作成したスナップショットです。
- Snapt2g3 は、グループ 3 (g3) が最初の増分更新バックアップ操作の完了後、時間 t2 で作成したスナップショットです。
- Snapt3g1 は、グループ 1 が 2 番目の増分更新バックアップ操作の完了後、時間 t3 で取ったスナップショットです。

このセットアップで、グループ 1 は 2 つのスナップショットを作成していますが、グループ 2 とグループ 3 はそれぞれ 1 つのスナップショットを作成しています。

これらのスナップショットからクローンが作成され、グループの特定の目的に応じてデータベース上で読み書き操作が実行されます。

- Qat1g1 は、Snapt1g1 スナップショットから作成され、グループ 1 が QA を目的に使用するクローンです。
- Devt1g1 は、同じく Snapt1g1 スナップショットから作成され、グループ 1 が開発を目的に使用するクローンです。
- Devt1g2 は、Snapt1g2 から作成され、グループ 2 が開発を目的に使用するクローンです。
- Qat2g3 は、Snapt2g3 から作成され、グループ 3 が QA を目的に使用するクローンです。
- Qat3g1 は、Snapt3g1 から作成され、グループ 1 が QA を目的に使用するクローンです。
- Devt3g1 は、Snapt3g1 から作成され、グループ 1 が開発を目的に使用するクローンです。

グループ 1 がスナップショット Snapt3g1 をとってクローン Qat3g1 と Devt3g1 が作成されると、スナップショット Snapt1g1 はこのグループにとって不要になることがあります。Snapt1g1 を破棄すると、そのクローンである Qat1g1 と Devt1g1 も破棄されます。

Sun ZFS Storage Appliance を使ったデータベースクローニングの利点

Sun ZFS Storage Appliance の機能を使用して、実稼働データベースをクローニングするための柔軟で効率的なソリューションを実装することができます。以下にその利点を説明します。

Oracle がテストおよび検証したソリューション

このドキュメントで説明する手順は、Oracle によってテストと検証が行われています。提供するソリューションの説明とスクリプトのサンプルがテストおよび開発のサイクルを迅速化し、配備までの時間を短縮します。

本書では、ASM ベースの実稼働データベースを使用するソリューションを説明しますが、この手順は若干の変更を加えるだけでどんな実稼働データベース環境にも使用できます。このソリューションは、Exadata Hybrid Columnar Compression (EHCC) がデータベーステーブルに対して有効になっていなければ、Oracle Exadata Database Machine で使用することができます。

既存のインフラに簡単に統合

Sun ZFS Storage Appliance は、既存のセットアップに影響を与えることなく、既存の SAN/NAS インフラに容易に適合します。MCL データファイルへの増分変更が更新されている間、ユーザーは自分のクローンデータベース上で作業を続けることができます。

アプリケーション配備の迅速化

DTrace Analytics によって提供されるユニークで包括的なインサイトにより、組織はより迅速にアプリケーションのアップグレード、パッチのリリース、およびスピーディな開発を配備することができます。

配備および管理の容易さ

マルチプロトコルのサポートにより、Sun ZFS Storage Appliance はどのようなインフラにも適合します。ユーザーインターフェースは、ストレージアプライアンスを管理するための直観的で便利な方法を提供します。データベース全体のクローニングソリューションをスクリプトして繰り返し実行することで、クローニング操作に必要な時間とリソースを減らすことができます。

効率的なバックアップ、ビジネス継続性、および障害回復

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すべてのデータサービスは、ストレージアプライアンスの価格に含まれています。どの機能やプロトコルを有効にしても、それに伴う追加費用はありません。これらすべての機能をさまざまに組み合わせて、簡単なバックアップ (多数の主要バックアップアプリケーションによって認定)、ほとんど瞬時に復元するビジネス継続性、および障害回復を目的としたリモートレプリケーションなどの特定のニーズを満たすことができます。

容量の効率的利用

従来の多くの方法を使って、Oracle 実稼働データベースの複製を実行できます。従来の方法では、データベースの完全かつその時点のコピー (すなわちクローン) を作成します。ただしこの方法には時間がかかるとともに、ストレージも大量に必要です。たとえば、1 TB データベースのコピーを 6 つクローニングするには、ほぼ 6 倍以上のストレージ容量が必要になります。

スナップショットを作成するとき、最初に容量は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を作成するとき、新しいブロックは変更が行われる場合にのみ書き込まれます。したがって、クローンにアクセスするとき、データはスタンバイデータベースのデータブロックを参照している可能性があります。これにより、多くのクローンを配備する際に容量を効率的に利用できます。

無制限のスナップショットとクローンによる柔軟性

複雑なテストおよび開発の環境を、完全な柔軟性をもって配備することができます。MCL マスターファイルシステムは、実稼働データベースで行われた変更を定期的に増分更新バックアップすることでほぼ最新に保たれています。あらゆる時点で MCL から作成するスナップショットの数に制限はなく、これらの各スナップショットから作成するクローンの数にも制限はありません。したがって、各開発者には自分用のデータベースが用意されるので、同じ時間内により多くのテストサイクルを完了できます。クローンインスタンスで何らかの問題が発生した場合、他のスナップショットやクローンに影響を与えないように、問題のクローンを破棄していつでも再作成することができます。

データベースのクローンバージョンを作成することで生まれる可能性は無限大です。MCL からスナップショットを作成し、いつでもそのスナップショットからいくらかでもクローンを作成できます。さらに、すべてのクローンはスナップショットに依存しますが、スナップショットは、そのスナップショットから作成されたどのクローンからでも作成できます。

ハイブリッドストレージプールによる応答時間の短縮

データベースのデータは、HSP モデルの長所を活かし、DRAM、フラッシュメモリー、およびハードディスクメディアの利用によって、クライアントのより高速なアクセスを可能にします。データベースの同期書き込みは、書き込みに最適化されたフラッシュによって処理されます。読み取りに最適化されたフラッシュは、最近頻繁にアクセスされているブロックを格納しているストレージ内の第 2 層のキャッシュとして機能します。複数のクローンが同じブロックにアクセスしている場合、データは DRAM またはフラッシュから提供されるので応答時間が短縮されます。図 4 に、HSP モデルに含まれている各種コンポーネントを示します。

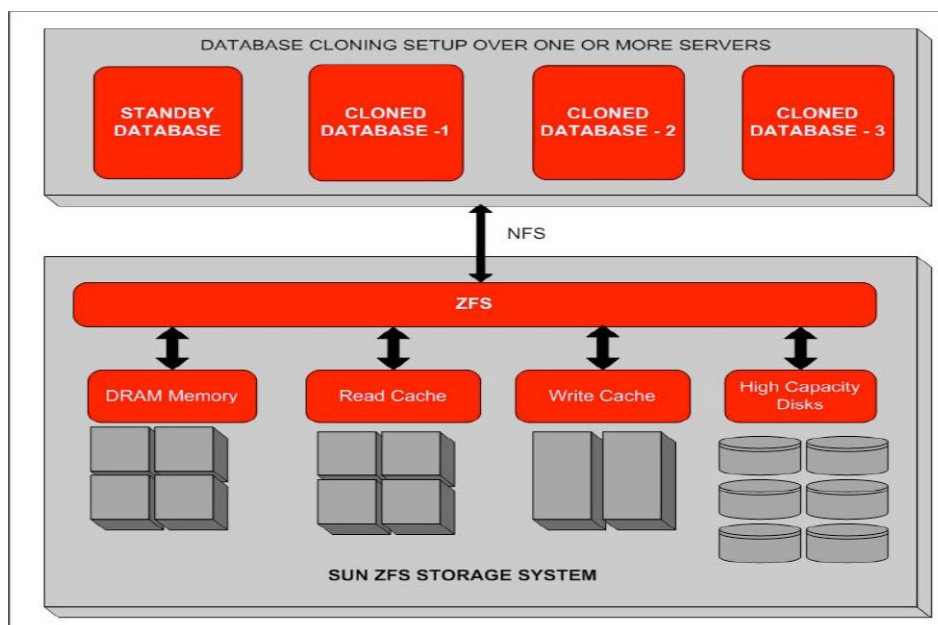


図 4. データベースクローニングに使用する Sun ZFS ストレージシステムのハイブリッドストレージプール

迅速な解決と計画のための Analytics 機能

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性能をグラフィカルに表現します。Analytics 機能によって提供される情報は、以下を行う上で役立ちます。

- 新たに開発されたコードが IO 性能全体にどのように影響して、テストサイクルを迅速化し、新しいコードの配備時間の短縮をもたらしているかを理解する。
- ストレージアプライアンス全体の健全性を把握する。
- ストレージアプライアンスの IOPS、応答時間、およびスループットを確認する。
- SLA 契約をサポートするために、クライアントのアクセスパターンを把握する。
- 容量計画の要件に対処する。
- クライアントとストレージアプライアンスの性能データを比較して問題を特定し解決する。
- 分析のために履歴データをエクスポートする。

Analytics 機能によって提供される情報の例を、以下の図にいくつか示します。

Sun ZFS Storage Appliance ダッシュボードは、システムコンポーネントの状態の概要を示します。図 5 の画面には、容量の使用率と、使用されている各プロトコルのためのアクティビティーが表示されます。

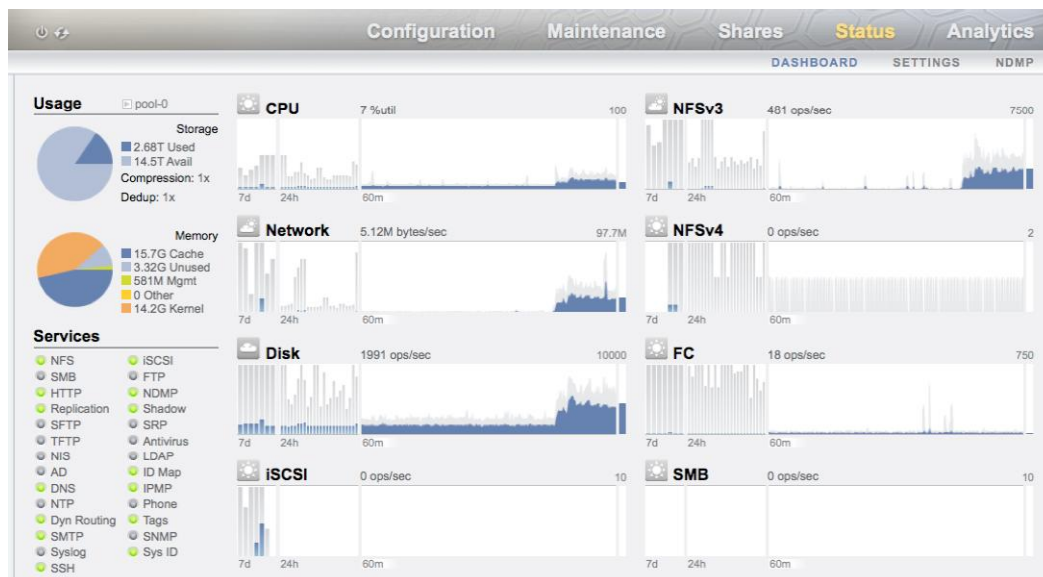


図 5. 容量使用率と使用されている各プロトコルのためのアクティビティーを示すダッシュボード

図 6 の Analytics 画面は、アプライアンスに格納されている MCL データファイルが増分更新バックアッププロセス中に更新されている様子を示しています。



図 6. NFSv3 プロトコルを使用したファイルアクセスを示す Analytics 画面

図 7 の Analytics 画面は、2 つの 10Gb インタフェース上で行なわれる Oracle RMAN バックアップを示しています。

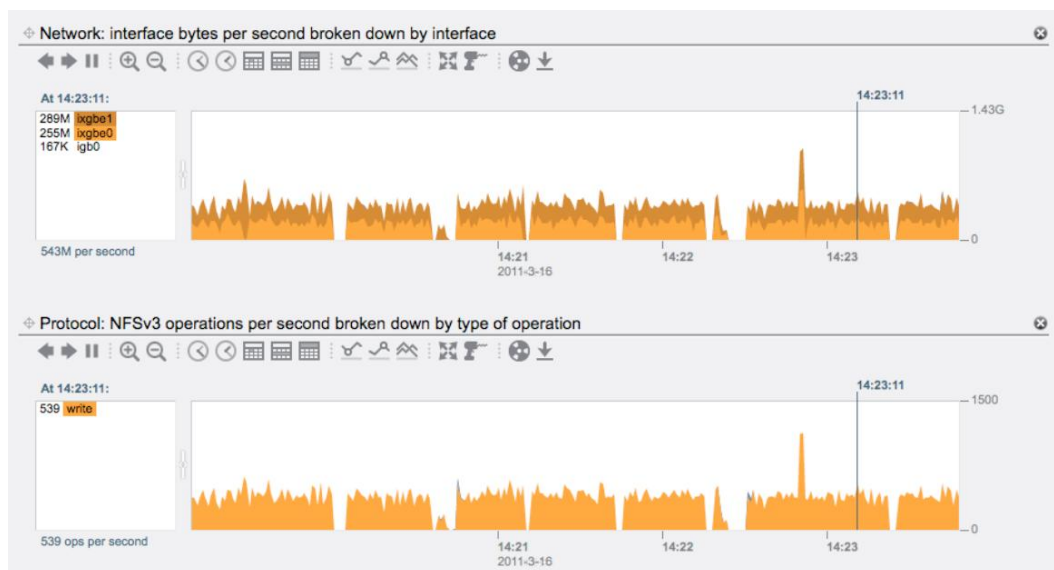


図 7. 2 つの 10Gb インタフェース上で行なわれる Oracle RMAN バックアップによって生成されるアクティビティを示す Analytics 画面

図 8 の Analytics 画面は、データベースアクセスを介してクローンファイルシステム上で実行される読み書き操作を示しています。

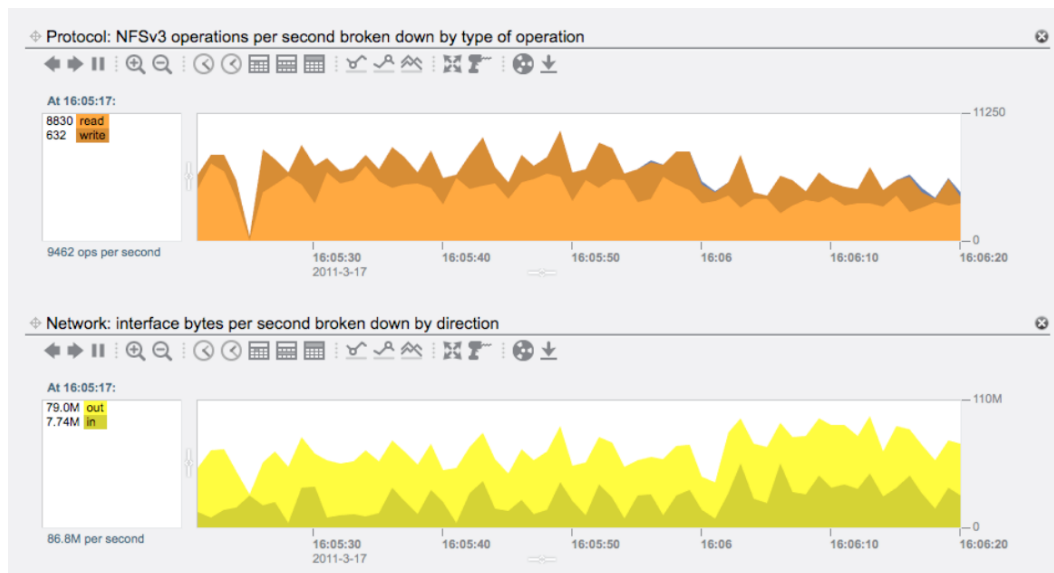


図 8. システム内の各共有について 1 秒あたりに実行される NFSv3 操作を示す Analytics 画面

図 9 の Analytics 画面は、NFSv3 操作ファイルごとの 1 秒あたりの階層的な内訳を含む、各種ファイルシステムおよびクローンデータベースファイルの詳細なアクセスパターンを示しています。

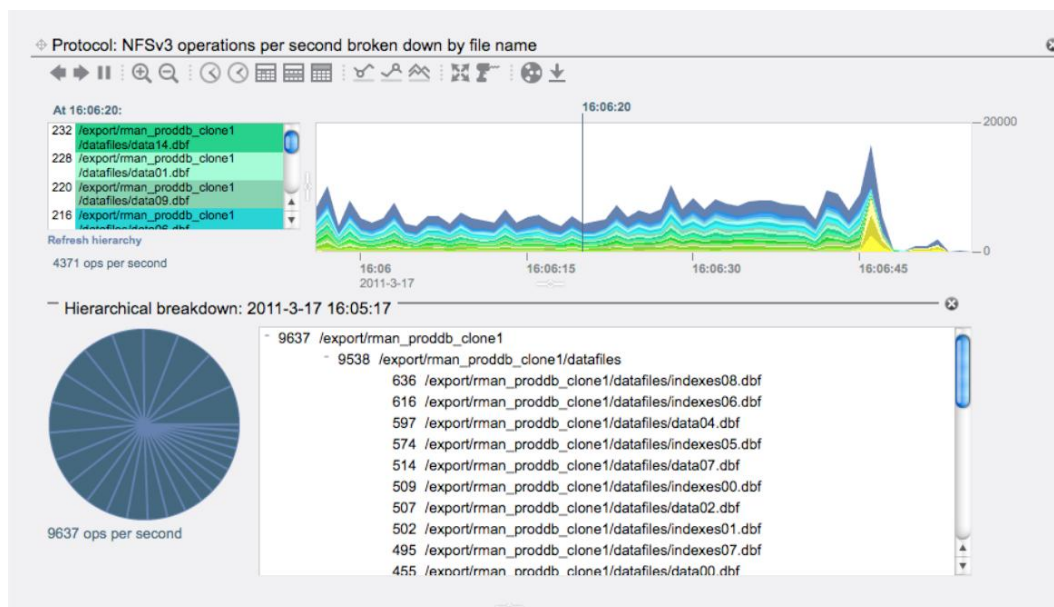


図 9. ファイルレベルでの NFSv3 操作の詳細なアクセスパターンを示す Analytics 画面

クローニングソリューションのサイジングに関する考慮事項

テストおよび開発環境は、さまざまなストレスレベル、容量、および可用性の要件を備えた IO 負荷を十分に処理できるほど堅牢でなければなりません。Sun ZFS Storage Appliance のアーキテクチャは、通常、多数の同時 IO ストリームを効果的に処理します。以下の項では、Sun ZFS Storage Appliance を使用してテストおよび開発環境を構築する際に考慮すべきいくつかの事項について説明します。

容量と保護

テストおよび開発環境用に作成されたスナップショットおよびクローンは、作成時にディスク容量を消費しません。変更のないブロックの読み取りがクローンファイルシステムで発生した場合、そのブロックは MCL ファイルシステムからサービスされます。ファイルシステムに複数のクローンが存在する場合でも、各クローンによるブロックへのアクセスは、データの単一の MCL コピーによってサービスされます。

クローンデータベースブロックを初めて変更する場合、変更されたデータには新しいブロックが割り当てられます。その時点から、クローンはその新しいブロックにアクセスしてそのデータへの読み書きアクセスを行います。このコピー書き込み機能によって、ディスク容量を大幅に節約できます。

クローンファイルシステムの容量使用率を確認するには、データベースへの書き込みに応じて行われる新しいデータブロックの作成から生じた結果の変化率を把握することが重要です。すべてのファイルのすべてのブロックが変更されるという最悪のケースの場合、クローンファイルシステムはベースボリューム自体のサイズまで拡張されます。

Sun ZFS Storage Appliance は、ミラーリング、シングルパリティ RAID-Z、デュアルパリティ RAID-Z2 など複数の RAID レベルを提供します。テストおよび開発環境に適用可能な RAID レベルは以下のとおりです。

- 最適性能を必要とする汎用アプリケーション、データベースなどのストレージには、ミラーリングが使用されます。ミラーリングにより、総 Raw 容量は 50 %削減されます。
- RAID-Z は、適度な性能とともにミラーリングよりも 50 %大きい容量を提供し、単一のディスク障害を許容します。RAID-Z は最適な容量が必要な場合に適しています。
- RAID-Z2 は、RAID-Z に比べて容量や性能が削られますが、高可用性という利点があります。RAID-Z2 はデータのストリーミングに最適です。

テストや開発の環境には、ミラーリングか RAID-Z のいずれかを使用することができます。

ネットワーク接続

本書で説明するソリューションは、NFS プロトコルに基づいています。実稼働データベースサーバーと Sun ZFS Storage Appliance の間で使用される接続のタイプは、Oracle RMAN バックアップ操作の要件に基づいて選択されます。同様に、クローンデータベースサーバーと Sun ZFS Storage Appliance 間の接続も、テストおよび開発環境の SLA 要件に依存します。どちらの場合でも、1Gb または 10Gb イーサネット接続を 1 つ以上使用することをお勧めします。必要に応じて、Oracle Exadata Storage Server と Sun ZFS Storage Appliance 間の接続用に、iPoIB を配備することができます。

最適化された OPS とスループット

Sun ZFS Storage Appliance では、読み取りスループットと IOPS (Input/output Operations Per Second) 性能を強化するために、読み取り最適化 SSD を提供しています。これらの SSD により、読み取り要求については応答時間 5 ms 未満を達成できます。各ストレージアプライアンスは、これらの大容量デバイス (それぞれ 512 GB) を 6 つまで取り込むことができます。これはシステム内の DRAM に加え、合計キャッシュが 3 TB です。

また、あるデータブロックが 1 つのクローンの 1 人のユーザーによって参照され、次に別のクローンの 1 人のユーザーによって同じデータブロックが要求された場合、そのデータは DRAM か読み取り最適化 SSD に存在している可能性があります。これは応答時間の高速化や全体的なエンドユーザーエクスペリエンスの向上につながります。

書き込み操作は、MCL のデータファイル上で増分更新バックアップ操作中に、またはクローンデータベース上で行われます。書き込み最適化 SSD を使用して、通常データベースによって実行される同期書き込みを吸収します。それぞれの書き込み最適化 SSD は 100 MB/sec の帯域幅に制限され、1 ms 未満の応答時間を提供します。書き込み SSD のストライピングによって最高の帯域幅が提供されます。たとえば、4 つの書き込み SSD をストライピングすると、最大 400 MB/sec の書き込みスループットが達成されます。

Oracle RMAN のバックアップ中に、書き込み SSD の数がボトルネックであることが判明した場合、logbias プロパティの値を latency ではなく throughput に変更することを検討してください。これによって書き込み SSD はバイパスされます。ただしこのモードは、マルチチャネルの Oracle RMAN バックアップなど、書き込み要求の数が多いために限り良好に機能します。

一般的推奨事項

このソリューションを最適化するための一般的な推奨事項を以下にいくつか示します。

- MCL にバックアップするため、Oracle RMAN に 8 つ以上のチャネルを構成します。
- このソリューションは MCL データファイルの単一の場所を示していますが、MCL 内のデータファイル用として、RMAN チャネルと同じ数のファイルシステムを作成することをお勧めします。
- Oracle データベースのブロックサイズに合わせて、各 datafile ファイルシステムのレコードサイズを 8 KB に設定します。これは、クローンデータベース datafiles の増分更新やストレージを効率的に行う上で重要です。
- NFS のダイレクト機能を有効にして、Oracle RMAN のスループットを向上させます。
- スナップショットとクローンに明確な名前を付けます。
- 使用していないクローンやスナップショットを削除します。
- 書き込み SSD の数が制限要因であると特定された場合、logbias プロパティの値を throughput に変更します。
- このソリューションには、Sun ZFS Storage Appliance 7320 または Sun ZFS Storage Appliance 7420 のいずれかを、適切なメモリー、接続、読み取り SSD、書き込み SSD、および必要な数のディスクとともに使用することをお勧めします。このソリューションは、要件に基づいて構築することができます。
- クローンを更新する必要がある場合は、クローンを破棄し、MCL でスナップショットを実行し、再度クローンを作成します。スナップショットのすべてのクローンを更新する必要がある場合は、初めにスナップショットを削除すれば、そのすべての子クローンが破棄されます。
- Oracle Exadata Storage Server と、EHCC で圧縮したテーブルを使用するソリューションを使う場合は、ALTER TABLE MOVE コマンド (リリース 11.2.0.2 で導入) を使用して圧縮テーブルを解凍すれば、Exadata 以外のインスタンスからでも自由アクセスできるようになります。
- クローンデータベースサーバーのオペレーティングシステムは、エンディアン (バイトの並び順) と一致するように、実稼働データベースサーバーのオペレーティングシステムに基づいて選択します。

結論

Sun ZFS Storage Appliance は、データベースのクローニングを実行するための理想的プラットフォームを提供します。このアプライアンスには、管理を容易に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回復を目的としたフルセットのデータサービス、あらゆるインフラに対応するためのマルチプロトコルのサポート、監視や問題解決のための解析、およびテスト、開発、QA 業務のための応答時間短縮を目的としたハイブリッドストレージプールなどが装備されています。

スナップショットやクローニングの可能性は無限なので、実稼働データベースに影響を与えることなく、さまざまな目的のために多くのデータベースインスタンスを同時に起動することができます。このような理由から、Sun ZFS Storage Appliance はデータベースのクローニングに関して最適なソリューションプラットフォームとなっています。

付録 A. 関連文書

Oracle 最大可用性アーキテクチャー Web サイト

<http://www.oracle.com/technetwork/database/features/availability/maa-best-practices-155366.html>

Oracle データベースの高可用性の概要 (Part #B14210)

<http://otn.oracle.com/pls/db111/db111.toc?partno=b28281>

Oracle データベースの高可用性のベストプラクティス (Part B25159)

<http://otn.oracle.com/pls/db111/db111.toc?partno=b28282>

Oracle Data Guard with 11g リリース 2 テクニカルホワイトペーパー

<http://www.oracle.com/technetwork/database/features/availability/twp-dataguard-11gr2-1-131981.pdf>

Sun ZFS Storage Appliance ドキュメント

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

NFS を使用した Sun Storage 7000 Unified Storage への Oracle データベース配備

http://www.sun.com/bigadmin/features/articles/7000_oracle_deploy_2009q3.jsp

Oracle データベース用の Sun Storage 7000 の構成

<http://wikis.sun.com/display/SystemsComm/Configuring+Sun+Storage+7000+Unified+Storage+Systems+for+Oracle+Databases>

Sun Storage 7000 Unified Storage システムのスナップショット機能を使用した Oracle データベースのバックアップとリカバリ

http://www.sun.com/bigadmin/features/articles/oracle_7000_snapshot.jsp

付録 B. Sun ZFS Storage Appliance について

この項では、Sun ZFS Storage Appliance の概要を説明します。詳細な特長、機能、およびプラットフォームオフリングについては、オラクルの Sun Unified Storage の Web サイト (以下の URL) を参照してください。

<http://www.oracle.com/us/products/servers-storage/storage/unified-storage/index.html>

アーキテクチャーの概要

Sun ZFS Storage Appliance は、管理の容易さ、さまざまなプロトコルを使用した接続、ビジネス継続性のためのデータサービスを 1 つのストレージアプライアンスで提供しています。このアプライアンスは、NFS、Common Internet File System (CIFS)、Internet Small Computer System Interface (iSCSI)、InfiniBand (IB)、ファイバチャネル (FC) プロトコルをサポートしています。このアプライアンスは、シングルヘッドまたはクラスタヘッドとして使用可能です。バックアップとリカバリを目的に、このアプライアンスは Network Data Management Protocol (NDMP) をサポートしています。

Sun ZFS Storage Appliance の中心は Oracle Solaris オペレーティングシステムであり、それに Oracle Solaris ZFS ファイルシステムと Sun ZFS Storage Appliance のデータ管理機能が加わって、全データの保存、管理、およびデータサービスを行います。これらのサービスには、直観的なユーザーインターフェースかコマンドラインインターフェース (CLI) を使ってアクセスします。

ZFS ストレージアーキテクチャーは、DRAM、フラッシュメモリー、および物理ディスクがシームレスに統合され効率的なデータ配置を行うハイブリッドストレージプール (HSP) を活用しています。ユーザー IO 要求のタイプやパターンに基づき、これらの層の間でのデータ移動がストレージアプライアンスによって自動処理されます。またこのアプライアンスには Analytics というパワフルな性能監視ツールがあり、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システムコンポーネントの性能に関する詳細を、多数のドリルダウンオプションを使って提供します。たとえばこの Analytics ツールは、どのクライアントがどのファイルシステムやファイルにアクセスしているか、ネットワークやディスクの待ち時間や応答時間、データ転送サイズ、その他多くの指標を提供します。

利用可能なプラットフォーム

Sun ZFS Storage Appliance は、価格、性能、容量、保護機能などに対するお客様のご希望を満たすために、4 つのモデルを用意しています。ミドルからハイエンドのプラットフォームは最大 2 TB の読み取りキャッシュを提供するので、読み取り応答時間は通常数ミリ秒程度となります。4 つのプラットフォームすべてでの書き込み最適化フラッシュによって、1 ミリ秒未満の同期書き込み応答時間が可能になります。

表 2 に、各モデルの詳細な構成を示します。

表 2. Sun ZFS Storage Appliance の各モデル

プラットフォーム	ストレージ 容量	プロセッサ	メモリー (DRAM)	書き込み 最適化 SSD	読み取り 最適化 SSD	クラスタ オプション
Sun ZFS Storage 7120	最大 60 x 2TB SAS ディスク [120TB]	1 x Quad Core Intel Westmere EP E5620 @ 2.4 GHz	最大 36 GB	最大 96 GB	-	なし
Sun ZFS Storage 7320 (詳細はコントローラに よる)	最大 96 x 2TB SAS ディスク [192TB]	2 x Quad Core Intel Westmere EP E5620 @ 2.4 GHz	最大 72 GB	最大 16 x 18 GB	最大 4 x 512 GB	あり
Sun ZFS Storage 7420 (詳細はコントローラに よる)	最大 576 x 2TB SAS ディスク [1.1PB]	4 x 6C Intel Nehalem EX E7530 @ 1.86 GHz または 4 x 8C Intel Nehalem EX X7550 @ 2 GHz	最大 512 GB	最大 96 x 18 GB	最大 4 x 512 GB	あり
Sun ZFS Storage 7720	拡張可能ラック。 各ラックは 720 TB	4 x 8C Intel Nehalem EX X7550 @ 2 GHz	コントローラ あたり 最大 512GB	ケージあたり 2 x 18GB	コントローラ あたり最大 4 x 512GB	あり (クラスタ のみ)

用語集

知っておくと役に立つ用語や概念を以下に示します。

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に類似するもの) は 1 セットの物理ディスク上に作成されます。さらにそのストレージプールの上にファイルシステムが作成されます。ユーザーはストレージプールを構成する際に、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、各ストレージプールに最適化された読み書き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や LUN はプロジェクト (整合グループとも呼ばれています) にグループ化されます。プロジェクトは、共有を管理するための共通の管理ポイントを定義します。プロジェクト内のすべての共有は、一般設定を共有したり、共有レベルに加えてプロジェクトレベルでも割り当てを適用することができます。また、プロジェクトを使うと論理的に関連した共有をまとめてグループにすることができるので、共通の属性 (蓄積スペースなど) には単一ポイントからアクセスすることができます。

共有 - 共有とは、サポートされているデータプロトコルを介してストレージ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および 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、Web-based Distributed Authoring and Versioning (WebDAV) 拡張した HTTP、および FTP 経由でアクセスすることができます。LUN はブロックベースのボリュームをエクスポートし、iSCSI 経由でアクセスすることができます。

project/share は、プール内で共有するための一意の識別子です。複数のプロジェクトに同じ名前の共有を含めることはできますが、1 つのプロジェクトの中に同じ名前を持つ共有を 2 つ含め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 の両方を含めることは可能で、その場合は同じネームスペースを共有します。

データサービス

Sun ZFS Storage Appliance には多数のデータサービスがバンドルされており、ライセンスフリーで使用できます。以下に、本書で説明するデータベースクローニングの手順に関連した重要なデータサービスをいくつか紹介します。

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、その時点のファイルシステムの読み取り専用コピーです。これは瞬時に作成され、最初に容量は割り当てられません。ベースファイルシステムに変更が行われるとブロックが割り当てられます (コピー書き込み機能)。スナップショットは、手動で開始することも、特定間隔でスケジュールを設定し自動化することもできます。スナップショットデータには、バックアップのために直接アクセスすることができます。

スナップショットブロックの読み取りは、対応するベースファイルシステムブロックによって提供されます。ベースファイルシステムで変更が作成されると、元のブロックはスナップショットによって参照され、変更された新しいブロックはファイルシステムによって参照されます。

プロジェクトのスナップショットを実行することは、プロジェクト内のすべての共有のスナップショットを実行するのと同じです。

クローン - Sun ZFS Storage Appliance は、無制限の数のクローンをサポートします。クローンとは、瞬時に作成される読み書き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を作成できます。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提示されます。通常のすべての操作は、クローンからのスナップショット作成も含めて、クローン上で行うことができます。クローンは通常、テスト、開発、QA、バックアップなどの環境で使用されます。

スナップショットを作成するときと同様、クローンを作成するときも、容量は割り当てられません。クローン内のブロックの読み取りは、対応するベースファイルシステムのブロックによって行われます。変更されたブロックは、ブロックがクローン内で変更された場合に限り割り当てられます。スナップショットとクローン間で容量が共有され、またスナップショットには複数のクローンがあるので、スナップショットは、アクティブなクローンのどれかを破棄せずに破棄されることはできません。

クライアント側から見ると、クローンファイルシステムは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に特別な要件は不要です。

スナップショットのロールバック - スナップショットのロールバックとは、ベースファイルシステムを、スナップショットを作成した時点まで戻すプロセスのことです。

Sun ZFS Storage Appliance で利用できるデータサービスについての詳細な説明は、Sun ZFS Storage Appliance のドキュメント (以下の URL) を参照してください。

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

リモートレプリケーション - データを非同期でターゲットにレプリケートし、それを障害回復の目的で使用することができます。レプリケーションは、ユーザー定義のスケジュールに応じて連続的に行うことも、あるいは必要なときに随時行うようにも設定できます。Sun ZFS Storage Appliance は、1 つのストレージアプライアンスから 1 つまたは複数のアプライアンスへのデータのリモートレプリケーションをサポートしています。

付録 C. ストレージとホストの構成

この項の表では、本書で説明する Oracle データベースのクローニングソリューションに使用されるストレージおよびホストの構成について説明します。

表 3 は、実稼働サイトの構成について説明しています。

表 3: 実稼働サイトの構成

パラメータ	値
ホスト名	aie-6300b
システム構成	Sun Blade T6300 / UltraSPARC-T1 32 core CPU / 16 GB
オペレーティングシステム	Solaris 10 U 8 (SPARC)
ストレージとの接続	FC – 500 GB LUNS
Oracle データベース	11.2.0.1 / シングルインスタンス / ASM
Oracle インスタンス名	PRODDB
ASM ディスクグループ	+PRODDG
データ、再実行、制御ファイル	+PRODDG/data/* +PRODDG/logs/*
Oracle RMAN ブロック変更 トラッキングファイル	/oradata/rman_master/PRODDB/incremental.f
Oracle RMAN 増分コピー用 NFS マウントポイント	/oradata/rman_proddb_master/datafiles /oradata/rman_proddb_master/archive
NFS マウントオプション	rw,bg,hard,nointr,rsize=131072,wsiz=131072,proto=tcp,vers=3

表 4 は、クローンデータベースホストの構成について説明しています。

表 4: クローンデータベースホストの構成

パラメータ	値
ホスト名	aie-6300c
システム構成	8 GB / CPU
オペレーティングシステム	Solaris 10 U 8 (SPARC)
ストレージとの接続	10 Gb および 1 Gb イーサネット
Oracle データベース	11.2.0.1 / シングルインスタンス / NFS
Clone インスタンス名	CLONE1 (データベース名: CLONEDB)
スナップショットをベースにしたクローン	SNAP_0
クローンデータベースの マウントポイント	クライアント: /oradata/clone1/data /oradata/clone1/lredoogs /oradata/clone1/archive /oradata/clone1/alerts
NFS マウントオプション	rw,bg,hard,nointr,rsize=32768,wsiz=32768,proto=tcp,vers=3
ORACLE_SID	CLONE1
ORACLE_HOME	/oracle/products/11.2.0/db [local to the host]
initCLONEDB.ora	*.control_files='/oradata/clone1/redologs/control0.ctl', '/oradata/clone1/redologs/control1.ctl' *.db_block_size=8192 *.db_cache_size=4096M *.db_file_multiblock_read_count=128 *.db_files=200 *.db_keep_cache_size=512M *.db_name='CLONEDB' *.db_recycle_cache_size=256M *.diagnostic_dest='/oradata/clone1/alerts' *.filesystemio_options='SETALL' *.instance_name='CLONE1' *.java_pool_size=32M

```

*.large_pool_size=32M
*.log_archive_dest_1='location=/oradata/clone1/archive'
*.log_archive_format='%t_%s_%r.arc'
*.log_buffer=104857600
*.log_checkpoint_interval=10000
*.log_checkpoint_timeout=1800
*.nls_date_format='MM/DD/YYYY HH24:MI:SS'
*.processes=400
*.service_names='CLONE1'
*.shared_pool_size=256M
*.standby_file_management='AUTO'
*.undo_management='auto'
undo_tablespace='UNDOTBS1'

```

```

listener.ora          SID_LIST_LISTENER =
                      (SID_LIST =
                        (SID_DESC =
                          (GLOBAL_DBNAME =CLONEDB)
                          (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 5 は、Sun ZFS Storage Appliance の構成について説明しています。

表 5: Sun ZFS Storage Appliance の構成

パラメータ	値	
ホスト名	aie-7320a	
システム構成	Sun ZFS Storage 7320 / 72GB メモリー	
接続	10 Gb イーサネット (データ用) 1 Gb イーサネット (管理用)	
Oracle RMAN マスターコピー ロケーション (MCL) プロジェクト	プロジェクト名: rman_proddb_master /export/rman_proddb_master/datafiles /export/rman_proddb_master/redologs /export/rman_proddb_master/archive /export/rman_proddb_master/alerts	レコードサイズ 8KB 128KB 128KB 128KB
クローンのプロジェクトおよび ファイルシステム	プロジェクト名: rman_proddb_clone1 /export/rman_proddb_clone1/datafiles /export/rman_proddb_clone1/redologs /export/rman_proddb_clone1/archive /export/rman_proddb_clone1/alerts	

表 6 は、このソリューションに使用するネットワークとストレージのスイッチの構成を示しています。

表 6: スイッチの構成

パラメータ	値
ネットワークスイッチ (ストレージおよび管理用)	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 SFPs Fabric OS: v6.1.0

付録 D. コマンドとスクリプト

以下の項では、Oracle データベースのクローニングソリューションの実装に関する詳細と、このソリューションで使用するコマンドとスクリプトについて説明します。

Oracle RMAN 増分バックアップ用に実稼働データベースサーバーを構成する

以下の手順では、マスターコピーロケーション (MCL) を設定し、NFS ファイルシステムをマウントします。

1. MCL の場所にするディレクトリを作成します。

```
# mkdir -p /oradata/rman_master/PRODDDB/datafiles
# mkdir -p /oradata/rman_master/PRODDDB/archive
```

2. /usr/sbin/mount を使用するか、/etc/vfstab (Solaris) に以下を入力して NFS ファイルシステムをマウントします。

```
# RMAN Master backup
10.80.44.11:/export/rman_proddb_master/datafiles -
/oradata/rman_master/PRODDDB/datafiles nfs - yes
rw,bg,hard,nointr,rsize=131072,wsiz=131072,proto=tcp,vers=3,noac
10.80.44.11:/export/rman_proddb_master/archive -
/oradata/rman_master/PRODDDB/archive nfs - yes
rw,bg,hard,nointr,rsize=131072,wsiz=131072,proto=tcp,vers=3,noac
```

3. ファイルシステムをマウントします。

```
# mount /oradata/rman_master/PRODDDB/datafiles
# mount /oradata/rman_master/PRODDDB/archive
# chown -fR oracle:dba /oradata/rman_master/PRODDDB
```

レベル 0 とレベル 1 のバックアップを実行するための Oracle RMAN スクリプト

このスクリプトでは、データベースの Oracle RMAN バックアップの後にアーカイブログをバックアップします。

```
$ rman target /
rman > run {
    configure device type disk parallelism 8 backup type to copy;
    allocate channel ch1 device type disk format
'/oradata/rman_master/%d/datafiles/%b';
    allocate channel ch2 device type disk format
'/oradata/rman_master/%d/datafiles/%b';
    allocate channel ch3 device type disk format
'/oradata/rman_master/%d/datafiles/%b';
    allocate channel ch4 device type disk format
'/oradata/rman_master/%d/datafiles/%b';
```

```

    allocate channel ch5 device type disk format
'/oradata/rman_master/%d/datafiles/%b';
    allocate channel ch6 device type disk format
'/oradata/rman_master/%d/datafiles/%b';
    allocate channel ch7 device type disk format
'/oradata/rman_master/%d/datafiles/%b';
    allocate channel ch8 device type disk format
'/oradata/rman_master/%d/datafiles/%b';
    backup incremental level 1
    for recover of copy with tag 'zfssa_clone' database reuse;
    recover copy of database with tag 'zfssa_clone';
}

```

上記のコマンドの実行後、Oracle RMAN はデータファイルを MCL にコピーします。これが完了したら、以下の説明に従ってアーカイブログのコピーを行います。まず初めに、現在のオンライン再実行ログファイルをアーカイブします。

```
rman> sql 'alter system archive log current' ;
```

(次に、アーカイブログのコピー操作を行います。この手順では、実稼働データベースにあるアーカイブログの場所から MCL にすべてのアーカイブログをコピーします。使用する形式は %h_%e_%a.arc で、これは実稼働データベースで使用するログアーカイブ形式の %t_%s_%r.arc に対応しています。この Oracle RMAN バックアップは主にクローニングを目的として行われるので、アーカイブログはコピーが完了した後も削除されません。8 つのチャネルに対して割り当てが行われます。)

```

run {
    configure device type disk parallelism 8 backup type to copy;
    allocate channel ch1 device type disk format
'/oradata/rman_master/%d/archive/%h_%e_%a.arc';
    allocate channel ch2 device type disk format
'/oradata/rman_master/%d/archive/%h_%e_%a.arc';
    allocate channel ch3 device type disk format
'/oradata/rman_master/%d/archive/%h_%e_%a.arc';
    allocate channel ch4 device type disk format
'/oradata/rman_master/%d/archive/%h_%e_%a.arc';
    allocate channel ch5 device type disk format
'/oradata/rman_master/%d/archive/%h_%e_%a.arc';
    allocate channel ch6 device type disk format
'/oradata/rman_master/%d/archive/%h_%e_%a.arc';
    allocate channel ch7 device type disk format
'/oradata/rman_master/%d/archive/%h_%e_%a.arc';
    allocate channel ch8 device type disk format
'/oradata/rman_master/%d/archive/%h_%e_%a.arc';
    BACKUP
        AS COPY skip inaccessible (archivelog all) ;
}

```

Sun ZFS Storage Appliance でのスナップショット操作の実行

アプライアンスのユーザーインターフェースかコマンドラインインターフェースのいずれかを使用して、Sun ZFS Storage Appliance 内に MCL マスタープロジェクトのスナップショットを作成することができます。

ユーザーインターフェースの場合は次の操作を行います。

1. **Shares** をクリックして Shares 画面を表示します。次に **Projects** をクリックし、**rman_proddb_master** プロジェクトを選択します。プロジェクト内の共有が右側に一覧表示されます。
2. **Snapshots** をクリックし、次に + アイコンをクリックします。スナップショットの名前を入力します (この例では snap_0)。

これで、クローンをそこから実行できるスナップショットの snap_0 が利用できるようになりました。

--または--

CLI の場合は次の操作を行います。

2. ssh 経由でストレージアプライアンスにログインします。
3. スナップショットを作成するには、以下の操作シーケンスを入力します。

```
$ ssh root@<appliance>
appliance:> cd /
appliance:> shares
appliance:> set pool=[pool name]
appliance:> select rman_proddb_master
appliance:> snapshots snapshot snap_0
```

Sun ZFS Storage Appliance でのクローニング操作の実行

スナップショットからクローンを作成するには、Sun ZFS Storage Appliance ユーザーインターフェースを使用して各手順を行うか、この項で後述するスクリプトを実行します。

1. ストレージアプライアンスのユーザーインターフェースから Shares、Projects の順にクリックし、次に + アイコンをクリックするとプロジェクト **rman_proddb_clone1** が作成されます。以下にデフォルトのマウントポイントを設定します。
/export/rman_proddb_clone1.
2. **rman_proddb_master** プロジェクト内の各共有について、以下の操作を行います。
 - **Snapshots** をクリックしてからスナップショット snap_0 の横にある + アイコンをクリックすると、図 10 のような Create Clone メニューが表示されます。
 - 作成先のプロジェクトとして **rman_proddb_clone1** を選択します。
 - **Apply** をクリックします。

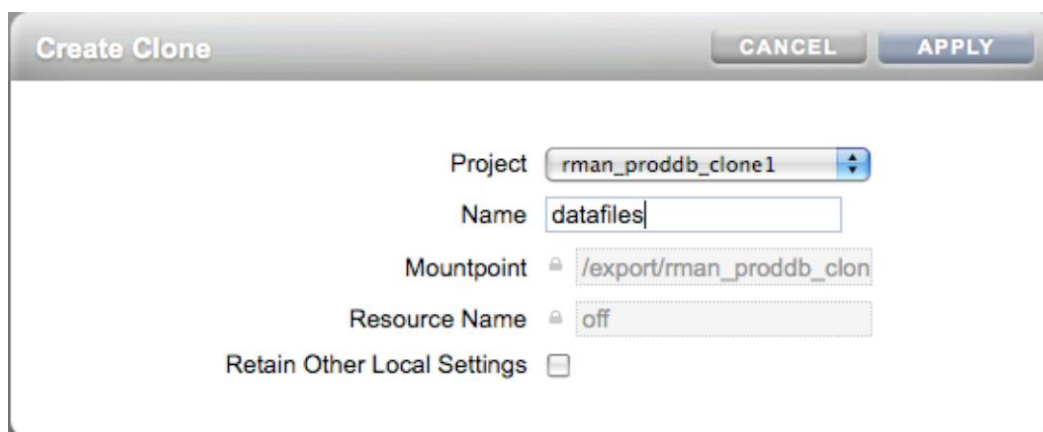


図 10. Create Clone メニュー

共有の datafiles、redologs、archive、alerts のスナップショット snap_0 のクローンが NFS エクスポート /export/rman_proddb_clone1 の下に作成されます。マウントポイントは継承されるので、datafiles 共有は /export/rman_proddb_clone1/datafiles の下にエクスポートされます。他のクローンも同様です。

これでデータベースは、データベースサーバーからアクセスできるようになります。

3. ステップ 1 と 2 を繰り返して、同じスナップショットから他のクローンを作成します。クローンは以下の要件を満たす必要があります。
 - 各プロジェクト名が一意であること。
 - 各共有のマウントポイントがストレージアプライアンス内で一意であること。

クローニング操作は、アプライアンスのコマンドラインインターフェースから aksh スクリプトを実行することによって行うこともできます。次の例のスクリプトは、**rman_proddb_clone1** の下の **rman_proddb_master** プロジェクト内で各共有のスナップショットを自動的にクローニングします。

```
$ ssh -T root@<appliance> < clone_the_shares.aksh
$ cat clone_the_shares.aksh
script
{
    var projName='rman_proddb_master';
    var cloneProj='rman_proddb_clone1' ;
    var snapName='snap_0' ;
    run('cd /');
    run('shares');
    run('set pool=pool-0');
    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");
}

```

クローンデータベースサーバーの構成

クローンデータベースサーバーを構成するには、root ユーザーとして以下の手順を実行します。

```

# mkdir -p /oradata/clonel/datafiles
# mkdir /oradata/clonel/redologs
# mkdir /oradata/clonel/archive
# mkdir /oradata/clonel/alerts

```

/etc/vfstab (Solaris) または /etc/fstab (Linux) 内にエントリを作成します。あるいは、これらの NFS 共有を mount コマンドを使って対話式でマウントすることもできます。

以下のエントリを /etc/vfstab (Solaris) または /etc/fstab (Linux) 内に作成します。または、NFS 共有を mount コマンドを使って対話式でマウントすることもできます。

```

# RMAN Clone
10.80.44.11:/export/rman_proddb_clonel/datafiles - \
/oradata/clonel/datafiles nfs - yes \
rw,bg,hard,nointr,rsize=131072,wsiz=131072,proto=tcp,vers=3,noac
10.80.44.11:/export/rman_proddb_clonel/archive - \
/oradata/clonel/archive nfs - yes \
rw,bg,hard,nointr,rsize=131072,wsiz=131072,proto=tcp,vers=3,noac
10.80.44.11:/export/rman_proddb_clonel/redologs - \
/oradata/clonel/redologs nfs - yes \
rw,bg,hard,nointr,rsize=131072,wsiz=131072,proto=tcp,vers=3,noac
10.80.44.11:/export/rman_proddb_clonel/alerts - \
/oradata/clonel/alerts nfs - yes \
rw,bg,hard,nointr,rsize=131072,wsiz=131072,proto=tcp,vers=3,noac

```

ファイルシステムは次のようにマウントします。

```
# mount /oradata/clone1/datafiles
# mount /oradata/clone1/redologs
# mount /oradata/clone1/archive
# mount /oradata/clone1/alerts
```

これで、MCL のポイントインタイムスナップショット snap_0 に基づく datafiles とアーカイブログをクローンデータベースサーバーが使用できるようになりました。

この時点では、redo ログや control ファイルはまだ使用できません。そこで、制御ファイルを作成し、データベースを回復させて resetlogs モードで開く必要があります。oracle ユーザーとして以下の操作を行います。

1. **ORACLE_SID 環境を CLONE1 に設定します。**

```
$ export ORACLE_SID=CLONE1
```

2. **クローンデータベースインスタンスを開始するために、適切な値で initCLONE1.ora ファイルを作成します。** initCLONE1.ora ファイルの例を以下に示します。

```
-bash-3.00$ cat initCLONE1.ora
*.control_files='/oradata/clone1/redologs/control0.ctl','/oradata/clone1/redologs/control1.ctl'
*.db_block_size=8192
*.db_cache_size=4096M
*.db_file_multiblock_read_count=128
*.db_files=200
*.db_keep_cache_size=512M
*.db_name='CLONEDB'
*.db_recycle_cache_size=256M
*.diagnostic_dest='/oradata/clone1/alerts'
*.filesystemio_options='SETALL'
*.instance_name='CLONE1'
*.java_pool_size=32M
*.large_pool_size=32M
*.log_archive_dest_1='location=/oradata/clone1/archive'
*.log_archive_format='%t_%s_%r.arc'
*.log_buffer=104857600
*.log_checkpoint_interval=10000
*.log_checkpoint_timeout=1800
*.nls_date_format='MM/DD/YYYY HH24:MI:SS'
*.processes=400
*.service_names='CLONE1'
*.shared_pool_size=256M
*.standby_file_management='AUTO'
*.undo_management='auto'
undo_tablespace='UNDOTBS1'
```

3. **initCLONE1.ora** を使用して、**nomount** 状態でデータベースインスタンスを開始します。

```
$ sqlplus / as sysdba
SQL > startup nomount pfile=initCLONE1.ora
ORACLE instance started.
Total System Global Area 5547376640 bytes
Fixed Size 2157664 bytes
Variable Size 335549344 bytes
Database Buffers 5100273664 bytes
Redo Buffers 109395968 bytes
SQL>
```

4. 次の SQL コマンドを使用して制御ファイルを作成します。このファイルはもともと、実稼働データベース上で ALTER DATABASE BACKUP CONTROLFILE TO TRACE コマンドを使って作成されたものです。redologs および datafiles の場所は、NFS の場所に変更されます。

```
-bash-3.00$ cat create_control.sql
STARTUP NOMOUNT PFILE=initCLONE1.ora
CREATE CONTROLFILE SET DATABASE "CLONEDB" REUSE RESETLOGS ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 64
    MAXINSTANCES 2
    MAXLOGHISTORY 292
LOGFILE
    GROUP 1 '/oradata/clone1/redologs/redo00-a.log' SIZE 512M
BLOCKSIZE 512,
    GROUP 2 '/oradata/clone1/redologs/redo01-a.log' SIZE 512M
BLOCKSIZE 512,
    GROUP 3 '/oradata/clone1/redologs/redo02-a.log' SIZE 512M
BLOCKSIZE 512,
    GROUP 4 '/oradata/clone1/redologs/redo03-a.log' SIZE 512M
BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
    '/oradata/clone1/datafiles/system00.dbf',
    '/oradata/clone1/datafiles/sysaux00.dbf',
    '/oradata/clone1/datafiles/undo00_a.dbf',
    '/oradata/clone1/datafiles/undo01_a.dbf',
    '/oradata/clone1/datafiles/data00.dbf',
    '/oradata/clone1/datafiles/data01.dbf',
    '/oradata/clone1/datafiles/data02.dbf',
    '/oradata/clone1/datafiles/data03.dbf',
    '/oradata/clone1/datafiles/data04.dbf',
    '/oradata/clone1/datafiles/data05.dbf',
```

```

'/oradata/clone1/datafiles/data06.dbf',
'/oradata/clone1/datafiles/data07.dbf',
'/oradata/clone1/datafiles/data08.dbf',
'/oradata/clone1/datafiles/data09.dbf',
'/oradata/clone1/datafiles/data10.dbf',
'/oradata/clone1/datafiles/data11.dbf',
'/oradata/clone1/datafiles/data12.dbf',
'/oradata/clone1/datafiles/data13.dbf',
'/oradata/clone1/datafiles/data14.dbf',
'/oradata/clone1/datafiles/indexes00.dbf',
'/oradata/clone1/datafiles/indexes01.dbf',
'/oradata/clone1/datafiles/indexes02.dbf',
'/oradata/clone1/datafiles/indexes03.dbf',
'/oradata/clone1/datafiles/indexes04.dbf',
'/oradata/clone1/datafiles/indexes05.dbf',
'/oradata/clone1/datafiles/indexes06.dbf',
'/oradata/clone1/datafiles/indexes07.dbf',
'/oradata/clone1/datafiles/indexes08.dbf',
'/oradata/clone1/datafiles/indexes09.dbf',
'/oradata/clone1/datafiles/tools00.dbf'
CHARACTER SET US7ASCII
;

```

制御ファイルは、上記の SQL コマンドを実行することによって作成されま

5. **datafiles** に複数のファイルシステムを使用する場合は、場所を変更しま

```
SQL > @create_control
```

```

30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45
Control file created.

```

6. **RECOVER UNTIL CANCEL** コマンドを使ってデータベースを回復します。デフォルトでは、このコマンドはアーカイブのデフォルトの場所 /oradata/clone1/archive にあるアーカイブログをポイントします。回復プロセスでは、コピーされる各アーカイブログの名前の入力を要求されます。
- コピーされる最後のアーカイブログの番号をメモしておきます。
 - 各アーカイブログファイルのコピーが終わったら、<Enter> を押します。
 - ステップ a でメモしたアーカイブログのコピーが終わり、回復プロセスが次のアーカイブログファイルの入力を要求したら、CANCEL と入力して <Enter> を押します。これにより、メディアの回復プロセスがキャンセルされます。

```
SQL> RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE ;
ORA-00279: change 24855560 generated at 03/11/2011 15:02:57 needed
for thread 1
ORA-00289: suggestion : /oradata/clonel/archive/1_295_723053455.arc
ORA-00280: change 24855560 for thread 1 is in sequence #295
```

```
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
[ Hit Enter ]
```

```
ORA-00279: change 24856093 generated at 03/11/2011 15:07:12 needed
for
thread 1
ORA-00289: suggestion : /oradata/clonel/archive/1_296_723053455.arc
ORA-00280: change 24856093 for thread 1 is in sequence #296
ORA-00278: log file '/oradata/clonel/archive/1_295_723053455.arc' no
longer
needed for this recovery
```

```
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
[ Hit Enter ]
```

```
ORA-00279: change 24856159 generated at 03/11/2011 15:07:32 needed
for
thread 1
ORA-00289: suggestion : /oradata/clonel/archive/1_297_723053455.arc
ORA-00280: change 24856159 for thread 1 is in sequence #297
ORA-00278: log file '/oradata/clonel/archive/1_296_723053455.arc' no
longer
needed for this recovery
```

```
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
CANCEL
Media recovery cancelled.
```

7. データベースを **resetlogs** モードで開きます。

/oradata/clonel/redologs の場所の下にオンライン再実行ログが作成されます。

```
SQL> alter database open resetlogs;
```

```
Database altered.
```

8. 一時ファイルを追加します。

```
SQL> ALTER TABLESPACE TEMP ADD TEMPFILE
'/oradata/clone1/datafiles/temp04.dbf'
SIZE 1038090240 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;
SQL> ALTER TABLESPACE TEMP ADD TEMPFILE
'/oradata/clone1/datafiles/temp02.dbf'
SIZE 1028653056 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;
SQL> ALTER TABLESPACE TEMP ADD TEMPFILE
'/oradata/clone1/datafiles/temp01.dbf'
SIZE 1041235968 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;
SQL> ALTER TABLESPACE TEMP ADD TEMPFILE
'/oradata/clone1/datafiles/temp00.dbf'
SIZE 1054867456 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;
```

これで実稼働データベース PRODDB のクローンが作成されます。

9. listener.ora および tnsnames.ora ネットワーク構成ファイルを必要に応じて更新し、クライアントが CLONE1 インスタンスに接続できるようにします。エントリの例を以下に示します。

```
CLONE1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = aie-6300c) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = CLONE1)
      (ORACLE_SID=CLONE1)
    )
  )
```

スナップショットをクローニングするための完全なスクリプト

スナップショットをクローニングするための完全なスクリプトを以下に示します。

```
#!/bin/sh
#
#
# 7000 ak script to clone all of the shares from a specified project
# (projName) and snapshot (snapName) stored in a specific pool
# (poolName) as a new project # (cloneName). This script checks for the
# existence of the clone project, and if it exists, it exits. The idea
# is that we don't want to accidently destroy existing data if the user
# enters the wrong project name for the clone target.
#
# The last arguments (root access ip addresses list) are a space delimited
# list of ipaddr/netmask that will be given root access to the shares.
# omitting these addresses sets nfs access to default (on).
#
# The script will either use an existing snapshot or create the snapshot
```

```

# if it does not exist. We can make multiple clones from a single
# snapshot, and this is probably what we want to do for things like
# dev/test/QA all using the same data set.
#
# In addition to creating the clone project, this script also sets
# logbias=latency on each cloned share. The intent is that if the
# source is a backup from Exadata we need logbias=throughput to
# optimize throughput when the database clone is created, but we need
# to switch this to latency when we use the clone for business
# processing.
#
#####

if [ $# -lt 5 ]
then
    echo "Usage: clonesnap.sh <login string: user@nashead> <poolName>
<projName> <snapName> <cloneName> [root access ip address list:
xxx.xxx.xxx.xxx/yy xxx.xxx.xxx.xxx/yyy ...]"
    exit 1
fi

LOGINSTRING=$1
POOLNAME=$2
PROJNAME=$3
SNAPNAME=$4
CLONENAME=$5

if [ $# -eq 5 ]
then
    SHARENFSSTRING=on
else
    shift 5
    SHARENFSSTRING=""
    while [ $# -gt 0 ]
    do
        if [ ${SHARENFSSTRING}x = "x" ]
        then
            SHARENFSSTRING=rw,root=@$1
        else
            SHARENFSSTRING=${SHARENFSSTRING}:rw,root=@$1
        fi

        shift 1
    done
fi

```



```

ssh -T $LOGINSTRING << EOF
script
{
    var projName='$PROJNAME';
    var cloneName='$CLONENAME';
    var snapName='$SNAPNAME';
    var poolName='$POOLNAME';

    printf( "Creating clone project %s from snapshot %s on project %s on pool
%s\n",
        cloneName,
        snapName,
        projName,
        poolName
    );
    run ( 'cd /' );
    run ( 'shares' );
    try
    {
        run ( 'set pool=' + poolName );
    }
    catch ( err )
    {
        printf( "Error selecting pool name %s\n", poolName );
        exit( 1 );
    }

    try
    {
        run ( 'project ' + cloneName );
    }
    catch ( err )
    {
        printf( "Clone %s already exists - double-check you really want to do
this and destroy the clone if appropriate.\n",
            cloneName
        );
        exit( 1 );
    }

    run ( 'set recordsize=8K' );
    run ( 'set atime=false' );
    run ( 'set sharenfs="${SHARENFSSTRING}"' );
    run ( 'commit' );
    run ( 'cd /' );
}

```

```
try
{
    run ( 'shares select ' + projName + ' snapshots select ' + snapName );
    printf( "Using precreated snapshot %s in project %s on pool %s.\n",
           snapName,
           projName,
           poolName
           );
}
catch ( err )
{
    printf( "Snapshot %s does not exist in project %s - creating\n",
           snapName,
           projName
           );
    try
    {
        run ( 'shares select ' + projName + ' snapshots snapshot ' +
             snapName );
    }
    catch ( err )
    {
        printf( "Unable to create snapshot %s on project %s in pool %s.\n",
               snapName,
               projName,
               poolName
               );
        exit( 1 );
    }
}

printf( "Cloning of project %s snapshot %s as project %s on pool %s.\n",
        cloneName,
        snapName,
        projName,
        poolName
        );
try
{
    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 ' + cloneName + ' ' + sharesList[i] );
run( 'set mountpoint=/export/' + cloneName + '/' + sharesList[i] );
run( 'set logbias=latency' );
run( 'commit' );
run( 'cd ../../..' );
}
}
catch (err)
{
printf( "Error cloning shares in projeect %s in pool %s\n",
cloneName,
poolName
);
exit( 1 );
}

printf( "Cloning of project %s as project %s on pool %s completed.\n",
cloneName,
projName,
poolName
);
exit( 0 );
}
.
EOF
```



Oracle Recovery Manager と Sun ZFS Storage Appliance を使用した Oracle データベースのクローニングソリューション

2011 年 4 月, Version 1.0

執筆者: Sridhar Ranganathan

協力執筆者: David Krenik

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

海外からのお問い合わせ窓口:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

oracle.com



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

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件を提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

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

AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。Intel、Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスに基づいて使用される SPARC International, Inc の商標または登録商標です。UNIX は X/Open Company, Ltd. からライセンス提供された登録商標です。1010

Hardware and Software, Engineered to Work Together