

Oracle Maximum
Availability Architecture

Oracle ホワイト・ペーパー
2013年1月

Oracle Exadata Database Machineでの Oracle GoldenGateの構成

概要	1
構成の概要.....	2
Oracle GoldenGate.....	2
Oracle Exadata Database Machine	2
Oracle Database File System	3
Oracle Clusterware.....	3
Oracle Exadata Database Machineへの移行	4
構成のベスト・プラクティス	4
手順1：Oracle Exadata Database Machine上のDBFSのセットアップ	4
手順2：Oracle GoldenGateのインストール	9
手順3：Oracle GoldenGateとデータベースのパラメータ構成	10
手順4：DBFSでのチェックポイント・ファイルと証跡ファイルのセットアップ	11
手順5：ローカル・ファイル・システムでのページ・ファイルのセットアップ	13
手順6：Replicatコミット動作の設定.....	14
手順7：Extract、Data Pump、Replicatプロセスの自動開始の設定.....	14
手順8：Oracle Clusterwareの構成	14
付録A：Oracle GoldenGate Clusterwareリソースの作成	20
Oracle RACへの配置に関する推奨事項.....	24
付録B：エージェント・スクリプトの例	25
参考資料.....	30

概要

Oracle Exadata Database MachineとOracle Maximum Availability Architecture (Oracle MAA) ベスト・プラクティスを戦略的に統合 (Exadata MAA) することで、もっとも包括的で最善のOracle Database可用性ソリューションが実現されます。

このホワイト・ペーパーでは、Oracle Exadata Database MachineおよびExadataストレージに対するOracle GoldenGateの連携を構成する際のベスト・プラクティスについて説明します。Oracle GoldenGateは、次をはじめとするさまざまなシナリオに有用です。

- 停止時間を最小限に抑えてOracle Exadata Database Machineへの移行を実施します。
- Oracle Exadata Database Machineに加えて、Oracle GoldenGateで提供される柔軟な可用性機能 (アクティブ-アクティブ・データベースによるデータ分散と継続的な可用性、計画停止における停止時間なし (または最小化) でのシステムの移行、アップグレード、保守) を必要とするアプリケーション・アーキテクチャの一部として使用できます。
- 各種 (場合によっては異種) ソース・データベースからOracle GoldenGate経由でデータを移入することで、Oracle Exadata Database Machine上にほぼリアルタイムのデータウェアハウスまたは統合データベースを実装できます。
- Oracle Exadata Database Machine上で稼働するOLTPアプリケーションからデータを取得して、SOAタイプの統合などの下流での消費をサポートします。

このホワイト・ペーパーでは、Oracle Exadata Database Machine上で稼働するOracle GoldenGateの構成に焦点を合わせて説明します。Oracle Exadata Database MachineはOracle GoldenGate処理に対するソース・データベースまたはターゲット・データベースとして、また場合によってはソースとターゲットの両方として使用できます。

さらに、このホワイト・ペーパーではOracle GoldenGateの標準モードについて説明します。このモードでは、オンラインREDOログ・ファイルまたはアーカイブREDOログ・ファイルから継続的に論理変更が抽出されます。

構成の概要

この項では、Oracle GoldenGate、Oracle Exadata Database Machine、およびOracle Database File Systemについて説明します。これらの製品の機能について、詳しくは本書最後にある参考資料の項を参照してください。

Oracle GoldenGate

Oracle GoldenGateは、ログベースのリアルタイム・チェンジ・データ・キャプチャ機能と異種システム間での配信機能を提供します。このテクノロジーを使用すると、負荷が低く費用効果の高いリアルタイム・データ統合および継続的可用性ソリューションを実現できます。

Oracle GoldenGateはトランザクションの整合性を維持したまま、既存インフラストラクチャへのオーバーヘッドを最小限に抑えて、コミットされたトランザクションを移動します。このアーキテクチャでは、1対多、多対多、カスケード、双方向などのさまざまなデータ・レプリケーション・トポロジがサポートされています。多様なユースケースには、リアルタイム・ビジネス・インテリジェンス、問合せのオフロード、停止時間なしのアップグレードおよび移行、アクティブ-アクティブ・データベースを使用したデータ分散、データ同期、高可用性が含まれます。図1に、Oracle GoldenGateのアーキテクチャを示します。

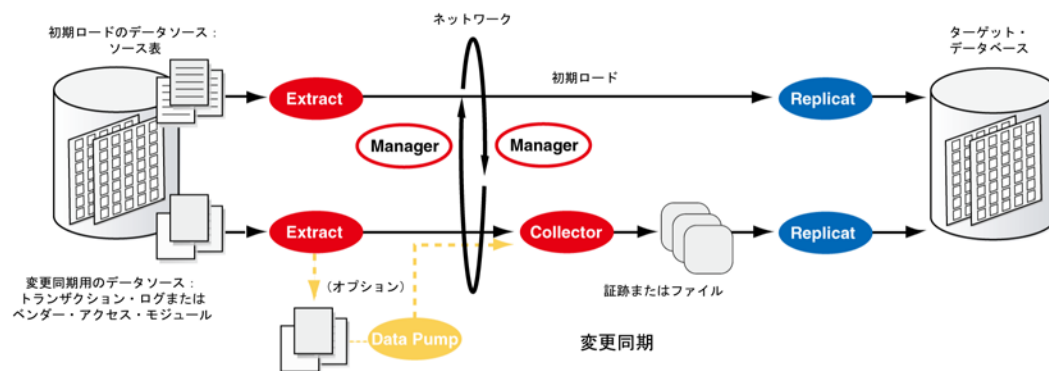


図1 Oracle GoldenGateのアーキテクチャ

Oracle Exadata Database Machine

Oracle Exadata Database Machineは配置が簡単で、あらゆるアプリケーションをサポートするOracle Databaseをホストすることを目的とした、すぐに使用できるソリューションであるとともに、最高レベルのパフォーマンスを実現します。

Oracle Exadata Database Machineは"ボックス内グリッド"で、データベース・サーバー、Oracle Exadata Storage Server、ストレージ・ネットワーク向けのInfinibandファブリック、およびOracle Databaseをホストするために必要なその他すべてのコンポーネントで構成されています。Oracle Exadata Storage ServerはOracle Databaseアプリケーションと使用する最適化されたストレージ製品であり、Oracle Exadata Database Machineのストレージ構成要素です。Oracle Exadataは、オンライン・トランザクション処理（OLTP）、データウェアハウジング（DW）、複合ワークロードの統合に対して、極めて高いI/OとSQL処理パフォーマンスを実現します。Oracle Real Application Clusters（Oracle RAC）、Exadataストレージ、Exadata Smart Flash Cache、高速InfiniBand接続、および圧縮テクノロジーを使用した超並列グリッド・アーキテクチャを利用することで、あらゆる種類のデータベース・アプリケーションで卓越したパフォーマンスが実現されます。

Oracle Database File System

Oracle Database File Systemは、データベース内に格納されたファイルに対するファイル・システム・インタフェースを作成します。Database File System（以下、DBFS）は、ローカル・ファイル・システムのように見える共有ネットワーク・ファイル・システムを提供する点においてNFSと似ています。データはデータベース内に格納されるため、このファイル・システムはデータベースで提供されるすべての高可用性およびディザスタ・リカバリ機能を引き継ぎます。

DBFSでは、サーバーはOracle Databaseであり、ファイルはSecureFiles LOBに格納されます。作成、オープン、読取り、書込み、ディレクトリ・リストなどのファイル・システム・アクセスのプリミティブは、PL/SQLプロシージャによって実装されます。データベース内のファイル・システム実装は、DBFS SecureFiles Storeと呼ばれています。DBFS SecureFiles Storeを使用すると、ユーザーはファイル・システムを作成し、クライアントからマウントできます。それぞれのファイル・システムには専用の表が含まれており、ファイル・システムの内容が格納されています。

Oracle Clusterware

Oracle Clusterwareを使用すると、サーバー同士が相互通信することで、1つの集合体として機能するように見えます。このようなサーバーの連結は一般に、クラスタとして知られています。サーバー自体はスタンドアロン・サーバーですが、サーバー間で通信するための追加プロセスが各サーバーに含まれています。このようにすることで、別々のサーバーの集合が、アプリケーションやエンドユーザーからは1つのシステムであるかのように認識されます。

Oracle Clusterwareは、Oracle Real Application Clustersの実行に必要なインフラストラクチャを提供します。また、仮想IP（VIP）アドレス、データベース、リスナー、サービスなどのリソースを管理します。

Oracle Clusterwareにアプリケーションを登録し、クラスタ環境内でのアプリケーションの管理方法をOracle Clusterwareに指示するAPIが提供されています。このAPIを使用して、Oracle GoldenGate Managerプロセスを、Oracle Clusterware経由で管理されるアプリケーションとして登録します。次に、Managerプロセスがその他のOracle GoldenGateプロセスを自動的に起動（または再起動）するように構成する必要があります。

Oracle Exadata Database Machineへの移行

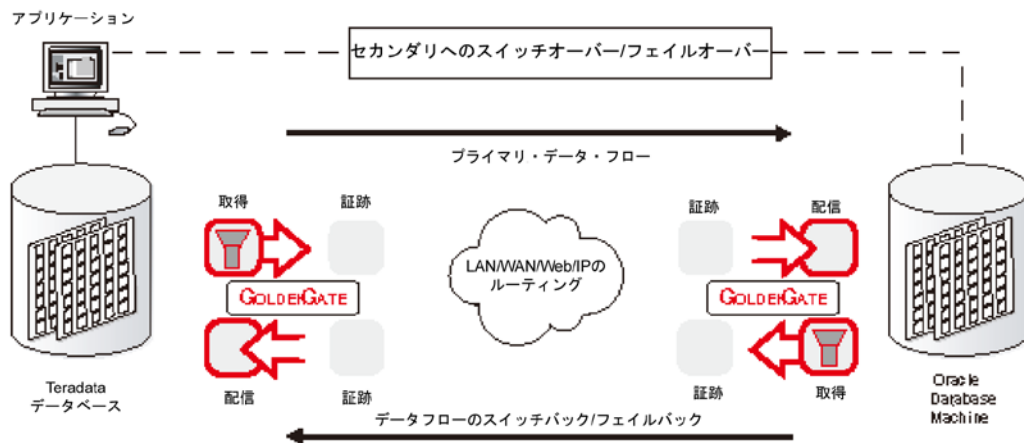


図2 Oracle Exadata Database MachineへのOracle GoldenGateの移行

Oracle GoldenGateは双方向のアクティブ-パッシブ構成をサポートしており、計画停止や計画外停止時にいつでもフェイルオーバーできるように準備されているライブ・スタンバイ・システム上にあるフル・レプリカ・データベースに対して、アクティブ・プライマリ・データベースからデータをレプリケートします。これによってOracle Exadata Database Machineへの移行機能が提供されるため、テストが完了してスイッチオーバーが計画されるまで、新しいシステムを並行して稼働できます。停止時間の短縮が要件であり、データベース移行にOracle Data Guardを使用できない場合は、Oracle GoldenGateを使用してデータベースを移行するのが最適です。特定のケース向けに最適な移行オプションを決定するには、Exadata MAAの文書『*Best Practices for Migrating to Exadata Database Machine*』を参照してください。

この文書には、Oracle Exadata Database Machine上にあり、図2のスタンバイ・データベースとして機能するターゲット・システムを構成する手順も含まれています。

構成のベスト・プラクティス

手順1：Oracle Exadata Database Machine上のDBFSのセットアップ

構成を設定する際のベスト・プラクティスは、Oracle GoldenGateの証跡ファイル、チェックポイント・ファイル、Bounded Recoveryおよび構成ファイルをDBFSに格納することで、最高のパフォーマンス、スケーラビリティ、システム障害時のリカバリ能力とフェイルオーバー機能を提供する方法です。

ノード障害が発生した場合にチェックポイント・ファイルと証跡ファイルの継続的可用性を提供するには、DBFSの使用が基本になります。クラスタ全体のチェックポイント・ファイルの可用性確保は、障害の発生後に、Extractプロセスが最後に確認されたアーカイブREDOログ・ファイルの位置からマイニングを継続し、Replicatプロセスが障害発生前と同じ証跡ファイルの位置から適用を開始できるようにするために不可欠です。DBFSを使用することで、存続しているいずれかのデータベース・インスタンスを、Extract/Data PumpプロセスのソースまたはReplicatプロセスの宛先として使用できるようになります。

Oracle Exadata Database Machine環境では、Oracle GoldenGateで使用されるDBFSデータベースを実行する際、ARCHIVELOGモードとNOARCHIVELOGモード間のパフォーマンス上の違いはほとんどありません。したがって、ARCHIVELOGモードでDBFSデータベースを実行することを推奨します。メディアで障害や破損が発生した場合のリカバリ能力が低下しないようにするためです。

ソースまたはターゲット環境に必要なライブラリ、パッチ、DBFSデータベース、必要なユーザー、および権限をインストールするには、My Oracle Support Note 1054431.1の手順に従ってください。

ソース環境（ExtractおよびData Pump）

Oracle GoldenGateの証跡ファイル、チェックポイント・ファイル、Bounded Recoveryファイル、一時ファイル、破棄ファイル、およびパラメータ・ファイルを格納するファイル・システムを1つ作成します。

最大12時間分の証跡ファイルを格納できるよう、証跡ファイルの十分なディスク領域を割り当てることを推奨します。これにより、ターゲット環境で新しい証跡ファイルを受信できないような問題が発生しても、Extract証跡ファイルの生成に十分な領域を確保できます。12時間分に必要な領域の量は、実際の本番データで証跡ファイルの生成率をテストすることによってのみ決定できます。

DBFS表領域の作成：

```
-- Connect to the DBFS database
SQL> connect system/<passwd>@<source_dbfs_tns_alias>

-- Create the tablespace:
SQL> create bigfile tablespace dbfs_gg_source_tbs datafile '+DBFS_DG' size
200g autoextend on next 8g maxsize 400g LOGGING EXTENT MANAGEMENT LOCAL
AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO;
```

サイズ・パラメータを、証跡ファイルに必要なストレージ・サイズに置換します。

ファイル・システムを作成します。

```
% cd $ORACLE_HOME/rdbms/admin
% sqlplus dbfs_user/dbfs_password
SQL> start dbfs_create_filesystem dbfs_gg_source gg_source1
```

DBFSで使用するLOBセグメントは、デフォルトのストレージ・オプションNOCACHE LOGGINGで構成する必要があります。

```
-- Connect to the DBFS database
```

```
SQL> connect system/<passwd>@<dbfs_tns_alias>
```

```
-- View current LOB storage:
```

```
SQL> SELECT table_name, segment_name, logging
       FROM dba_lobs WHERE tablespace_name='DBFS_GG_SOURCE_TBS';
```

```
-- More than likely it will be something like this:
```

```
--
```

TABLE_NAME	SEGMENT_NAME	LOGGING	CACHE
T_GOLDENGATE	LOB_SFSS\$_FST_73	YES	NO

LOBセグメントでNOCACHE LOGGINGが使用されていない場合は、代わりに次のように構成します。

```
SQL> ALTER TABLE DBFS.<TABLE_NAME> MODIFY LOB (FILEDATA)
       (NOCACHE LOGGING);-- View the new LOB storage:
```

```
SQL> SELECT table_name, segment_name, logging
       FROM dba_lobs WHERE tablespace_name='DBFS_GG_SOURCE_TBS';
```

```
--
```

TABLE_NAME	SEGMENT_NAME	LOGGING	CACHE
T_GOLDENGATE	LOB_SFSS\$_FST_73	YES	NO

新しく作成したDBFSファイル・システムを構成して、ノード障害の発生後にOracle Cluster Ready Services (Oracle CRS) によってDBFSインスタンスおよびマウント・ポイント・リソースが自動的に起動されるようにするには、My Oracle Support Note 1054431.1の手順に従ってください。Oracle Clusterwareでリソースを登録する際には、My Oracle Support Noteで指定されているとおり、local_resourceではなくcluster_resourceとして作成してください。

```
crsctl add resource $RESNAME ¥
```

```
-type cluster_resource ¥
```

```
-attr "ACTION_SCRIPT=$ACTION_SCRIPT, ¥
```

```
CHECK_INTERVAL=30,RESTART_ATTEMPTS=10, ¥
```



```
START_DEPENDENCIES='hard(ora.$DBNAMEL.db)pullup(ora.$DBNAMEL.db)',¥
STOP_DEPENDENCIES='hard(ora.$DBNAMEL.db)',¥
SCRIPT_TIMEOUT=300"
```

ファイル・システムをマウントしたら、新しく作成したファイル・システムにディレクトリを作成してOracle GoldenGateファイルを格納します。

例：

```
% cd /mnt/dbfs_source/goldengate
% mkdir dirchk
% mkdir dirpcs
% mkdir dirprm
% mkdir dirdat
% mkdir BR
```

Oracle GoldenGateパラメータによって制御されないディレクトリに、シンボリック・リンクを作成します。

```
% ln -s /mnt/dbfs_source/goldengate/dirprm $GG_HOME/dirprm
% ln -s /mnt/dbfs_source/goldengate/dirchk $GG_HOME/dirchk
% ln -s /mnt/dbfs_source/goldengate/dirpcs $GG_HOME/dirpcs
```

Oracle GoldenGateバージョン11.1.1では、ExtractにBounded Recovery (BR) 機能が追加されました。この機能を使用すると、計画停止や計画外停止などのいかなる理由でExtractプロセスが停止していても、停止時にどれだけ多くオープン（コミットされていない）トランザクションが残っていても、これらのトランザクションがどれだけ古くても、効率的なリカバリが保証されます。Bounded Recoveryでは、Extractプロセスが停止位置まで戻り、通常処理を再開するためにかかる最大時間の上限が設定されます。フェイルオーバーの発生時に、時間のかかるオープン・トランザクションがあった場合、ExtractプロセスがBounded Recoveryを使用してリカバリの実行時間を短縮できるようにするには、Bounded Recoveryのチェックポイント・ファイルを共有ファイル・システムに配置する必要があります。Oracle GoldenGateバージョン11.2.1以降ではBounded Recoveryファイルがサポートされており、DBFSに配置することを推奨します。以前のリリースでは、Bounded RecoveryファイルをNFSストレージ（Exadataに接続されているZFS Storage Applianceなど）に格納する必要があります。ローカル・ファイル・システムにチェックポイント・ファイルを格納することもできますが、ノード障害後にExtractプロセスがリカバリを実行する際、新しいローカルのBounded Recoveryチェックポイント・ファイルが作成されるまでは、通常のチェックポイント・メカニズムが使用されます。この違いは、障害の発生時に長時間実行トランザクションがあった場合にのみ顕著になります。

Bounded Recoveryファイル・ディレクトリを設定するには、次のExtractパラメータを使用します。

```
BR BRDIR /mnt/dbfs_source/goldengate/BR
```

Bounded Recoveryについて詳しくは、『Oracle® GoldenGate Windows and UNIXリファレンス・ガイド』を参照してください。

http://docs.oracle.com/cd/E35586_01/doc.1121/b69763.pdf

ExtractおよびData Pumpの証跡ファイル・ディレクトリの場所は、プロセスの作成中に指定します。Extractの場合、EXTTRAILを使用してパラメータ・ファイルでも指定します。

ターゲット環境 (Replicat)

Replicatプロセスによって証跡ファイルが読み取られ、そのデータがターゲット・データベースに適用されるターゲット環境では、2つの個別のDBFSファイル・システム用の要件があり、証跡ファイルとチェックポイント・ファイルの異なるI/O要件を分けることができます。

証跡ファイルは、ターゲット・ホスト上のCollection Serverプロセスによって書き込まれます。この際、ファイルの最初から最後まで連続したシリアルI/Oが使用され、Data Pump構成に従ってサイジングされます。各Replicatプロセスでも、連続したシリアルI/Oリクエストを使用して、同じ証跡ファイルが読み取られます。Replicatプロセスで証跡の一部が読み取られると、通常は同じプロセスで2回読み取られることはありません。同じ証跡ファイルから複数のReplicatプロセスによる読取りを使用する場合、証跡ファイルの同じ部分から同時に読み取られて、同期していることはほとんどありません。このため、DBFSに最適な構成は、NOCACHE LOGGINGストレージ・オプションを使用することです。これは、上記のソース環境の構成で説明しています。

チェックポイント・ファイルは小さい（約4KB）ですが、頻繁に書き込まれ、以前のデータが上書きされます。このファイルのサイズは増えず、プロセスの起動中にのみ読み取られます。これにより、リカバリや開始の適切な開始点が決定されます。チェックポイント・ファイルは繰り返し書き込まれるため、CACHE LOGGINGストレージ・オプションを使用してDBFSにファイルを格納すると、パフォーマンスが最大化されます。

CACHEオプションを設定すると、DBFSインスタンスのバッファ・キャッシュに書き込まれるチェックポイント・ファイルに対し、少量のデータしか書き込まれなくなり、ディスクへの直接書込みが実行されなくなるため、I/Oの待機時間が長くなります。テストでは、DBFSでNOCACHE構成を使用した場合と比べて、チェックポイント・パフォーマンスが2～5倍向上しました。

（上記の）ソース環境上のファイル・システムとほぼ同じ方法で、チェックポイント・ファイルの2番目のDBFSファイル・システムを作成します。次の点に注意してください。

- ・ ファイル・システムはチェックポイント・ファイル専用のため、可能なサイズは100MB未満です。
- ・ 最初に作成したファイル・システムと同じユーザーを使用して、ファイル・システムを作成します。必ず同じユーザーで両方のファイル・システムを作成することが重要です。
- ・ LOBストレージ・パラメータをCACHE LOGGINGに変更します。

```
-- Connect to the DBFS database
SQL> connect system/<passwd>@<dbfs_tns_alias>

-- View current LOB storage:
SQL> SELECT table_name, segment_name, logging
      FROM dba_lobs WHERE tablespace_name='DBFS_GG_CKPT_TBS';

-- Likely it will be something like this:
--
-- TABLE_NAME          SEGMENT_NAME          LOGGING CACHE
--
-- -----
-- T_GOLDENGATE2        LOB_SFSS$_FST_75        YES        NO

SQL> ALTER TABLE DBFS.<TABLE_NAME> MODIFY LOB (FILEDATA)
      (NOCACHE LOGGING);

-- View the new LOB storage:
SQL> SELECT table_name, segment_name, logging
      FROM dba_lobs WHERE tablespace_name='DBFS_GG_CKPT_TBS';

TABLE_NAME          SEGMENT_NAME          LOGGING CACHE
-----
T_GOLDENGATE2        LOB_SFSS$_FST_75        YES        YES
```

注：Oracle GoldenGateのData Pumpプロセスを使用して、DBFSを使用しているデータベース・マシン上のソース・ホストから証跡ファイルを送信している場合、Oracle GoldenGate Supportに問い合わせるBug 10146318の修正プログラムを入手してください。このバグ修正によって、Oracle GoldenGateサーバーおよびCollectorプロセスによるDBFSでの証跡ファイル作成のパフォーマンスが向上します。これは、Oracle GoldenGateの11.1.1.0.5より前のバージョンにのみ影響します。

手順2：Oracle GoldenGateのインストール

- 1 次のOracle Technology Network (OTN) サイトからOracle GoldenGateソフトウェアをダウンロードします。
<http://www.oracle.com/technetwork/jp/middleware/goldengate/downloads/index.html>
- 2 Oracle RAC構成のプライマリ・ソース・ノードとターゲット・ノードにOracle GoldenGateをローカル・インストールします。すべてのノードで同じインストール・ディレクトリを使用してください。
- 3 プライマリ・ソース・ノードおよび（または）ターゲット・ノードでのOracle GoldenGate構成が完了したら、ExtractおよびReplicatプロセスを停止し、Oracle GoldenGateホーム・ディレクトリ全体をもう一方のソース・ノードとターゲット・ノードにコピーします。
- 4 次のドキュメントの第2章"Oracle GoldenGateのインストール"に記載された一般的なインストール手順に従って、ソース・マシンとターゲット・マシンへのインストールを実行します。

http://docs.oracle.com/cd/E35586_01/doc.1121/b70206.pdf

手順3：Oracle GoldenGateとデータベースのパラメータ構成

進化したデータ型サポート、透過的データ暗号化（TDE）、およびデータ圧縮を利用するには、Oracle GoldenGate Extractを統合キャプチャ・モードで構成することを推奨します。

Oracle GoldenGateバージョン11.2.1で導入されたExtractは、統合キャプチャ・モードで使用できます。ExtractはOracleデータベースのログ・マイニング・サーバーと統合されており、そのサーバーから論理変更レコード（LCR）形式の変更データを受信します。Extractを構成して、ローカルまたはダウンストリームのマイニング・データベースから取得できます。統合キャプチャはデータベースと完全に統合されているため、Oracle RAC、自動ストレージ管理（ASM）、TDE、およびデータ圧縮で使用するための追加設定は不要です。My Oracle Support Note 1411356.1に記載されているとおり、統合キャプチャはOracle 11.2.0.3以降、パッチで使用できます。またOracle 10.2.0.4以降のバージョンでは、ダウンストリームのマイニング・デプロイメントで、変更取得に使用できます。

Extractを構成して、サポート対象バージョンのOracleのREDOログから直接取得することもできます。この構成は、クラシック・キャプチャ・モードと呼ばれます。

1 統合キャプチャ・モードを使用するExtract。

- a. データベース初期化パラメータSTREAMS_POOL_SIZE = 1.25GB X #Integrated Capture Processesを設定する。

統合キャプチャ・モードでのExtractの構成について詳しくは、『Oracle® GoldenGate Oracleインストレーションおよびセットアップ・ガイド』を参照してください。

http://docs.oracle.com/cd/E35586_01/doc.1121/b70206.pdf

2 クラシック・キャプチャ・モードでのExtract。

- a. アーカイブREDOログ・ファイルに対して、デフォルトのOracle Automatic Storage Manager（Oracle ASM）ネーミング規則を使用する。
- b. Oracle ASMログ読取りの新しいAPIに対して、Oracle GoldenGateのExtractパラメータを設定する。

Oracle GoldenGateリリース11.1.1では、Oracle ASMに保存されているログ・ファイルの読取りに対して新しいメソッドが実装されています。この新しいメソッドは、Oracle ASMインスタンスに直接接続する代わりに、データベース・サーバーを使用して、REDOログ・ファイルとアーカイブREDOログ・ファイルにアクセスします。このデータベースには、APIモジュールを含むライブラリが格納されている必要があります。このライブラリは現在、Oracle Databaseリリース10.2.0.5、11.2.0.2、および11.2.0.3に付属しています。

Oracle ASMが管理するストレージ・セル上にあるOracleアーカイブREDOログ・ファイルを正しくマイニングするには、次のとおりにOracle GoldenGateのExtractパラメータを構成します。

TRANLOGOPTIONS/パラメータを設定し、新しいログ読取りAPIを使用するように指定します。次に例を示します。

```
TRANLOGOPTIONS DBLOGREADER
```

3 Data Pumpを構成します。

プロセスでマッピングや変換が実行されていない場合は、Data PumpをPASSTHRUパラメータで構成します。PASSTHRUを使用することで、Data PumpによるCPUが減ります。データベースやデータ定義ファイルから表の定義を検索する必要がないためです。

PASSTHRUを使用したExtract構成やData Pumpについて詳しくは、『Oracle® GoldenGate Windows and UNIXリファレンス・ガイド』を参照してください。

http://docs.oracle.com/cd/E35586_01/doc.1121/b69763.pdf

手順4：DBFSでのチェックポイント・ファイルと証跡ファイルのセットアップ

1 チェックポイント・ファイルをセットアップします。

チェックポイント・ファイルには、ExtractプロセスとReplicatプロセスに対する現在の読取りおよび書込み位置が格納されています。システム、ネットワーク、またはOracle GoldenGateプロセスを再起動する必要が発生した場合、チェックポイントによってデータ損失が防止され、フォルト・トレランスが提供されます。

チェックポイント・ファイルをローカル・ファイル・システムに配置すると、データベース・ノードの障害発生時に高可用性が提供されません。チェックポイント表を使用してReplicatチェックポイント情報を記録すると、フォルト・トレランスを提供するもう1つの手段として利用できます。

チェックポイント・ファイルをDBFSに保存する場合のベスト・プラクティスは、Oracle GoldenGateホーム・ディレクトリからDBFS内のディレクトリへシンボリック・リンクを作成することです。詳細は、上記の手順1のDBFSの構成で説明しています。

注：Oracle GoldenGateはチェックポイント・ファイル上のファイル・ロックを使用して、ExtractプロセスまたはReplicatプロセスがすでに実行中であるかどうかを判断します。通常、これによって、チェックポイント・ファイルにアクセスする別のOracle RACノード上で、プロセスが二重に開始されることを防止します。DBFSは、このようなファイル・ロック方法をサポートしていません。DBFSを**1つの**Oracle RACノード上にマウントすることで、別のノードからチェックポイント・ファイルへアクセスすることを防ぎます。こうすることで、ExtractプロセスやReplicatプロセスが複数ノードで同時に開始されないようにすることができます。

2 証跡ファイルのセットアップ

証跡ファイルには、アーカイブREDOログ・ファイルから抽出されたデータが格納されています。証跡ファイルは、Extractプロセスによって自動的に生成されます。

DBFS上に証跡ファイルを保存します。NFSマウントと同様に、ソース・データベースとターゲット・データベースの両方で同じDBFSディレクトリにマウントを実行すると、Extractプロセスが作成したものと同一証跡ファイルをReplicatプロセスから読み取ることができます。ソース・データベースとターゲット・データベースの両方が同じOracle Exadata Database Machine上で実行されている場合は、こうすることでOracle GoldenGate Data Pumpを使用する必要がなくなります。

ソース・データベースとターゲット・データベースが同じExadata Database Machine上にならない場合は、Oracle GoldenGate Data Pumpを使用してホスト間で証跡ファイルを送信する必要があります。証跡ファイルにフォルト・トレランスを提供するため、DBFSを必ず使用してください。

DBFSでのソース・データベース向けOracle GoldenGate証跡ファイルの構成方法

- 1 DBFSファイル・システムを作成してマウントします。

詳細は、上記の手順1のDBFSの構成で説明しています。

- 2 EXTTRAIL Extract/パラメータを設定します。

```
EXTTRAIL /mnt/dbfs/goldengate/dirdat/aa
```

- 3 Extractを作成した後で、同じEXTTRAIL/パラメータ値を使用してローカル証跡ファイルを追加します。

```
% ggsci
```

```
GGSCI (ggtest.oracle.com) 1> ADD EXTTRAIL
```

```
/mnt/dbfs/goldengate/dirdat/aa, EXTRACT ext_db, Megabytes 500
```

Extractの作成方法については、『Oracle® GoldenGate Windows and UNIX管理者ガイド』を参照してください。http://docs.oracle.com/cd/E35586_01/doc.1121/b69762.pdf

DBFSでのターゲット・データベース向けOracle GoldenGate証跡ファイルの構成方法

- 1 ターゲット環境上にDBFSディレクトリがあらかじめ作成されていることを確認します。
- 2 次のとおりにEXTTRAIL Replicat/パラメータを設定します。

```
EXTTRAIL /mnt/dbfs/goldengate/dirdat/aa
```

- 3 Replicatを追加する際、同じEXTTRAIL/パラメータ値を使用します。

```
% ggsci
```

```
GGSCI (ggtest.oracle.com) 1> ADD REPLICAT rep_db1, EXTTRAIL
```

```
/mnt/dbfs/goldengate/dirdat/aa
```

証跡ファイルをローカル・ファイル・システムに配置すると、ノード障害発生時の再起動時間が長くなり、可用性が低下するため、ローカル・ファイル・システムには配置しないでください。

同じExadata Database Machine上にないソース・データベースとターゲット・データベース間のData Pumpの設定方法

- 1 ExtractとReplicatが構成済みであることを確認します。
- 2 Data PumpパラメータRMTHOSTを設定し、ターゲットへの接続に使用するIPまたはホスト名を指定します。下記の手順7で、Cluster Ready Servicesを使用してアプリケーション仮想IPアドレスが作成されます。こうすることで、コンピュート・ノード間で単一IPアドレスを移行できるため、障害発生ノードから存続ノードへターゲット・ホストが移行された場合にもData Pumpからの接続を続行できます。

```
RMTHOST gg_dbmachine, MGRPORT 8901
```

- 3 ターゲット・ホスト上の証跡ファイルの場所にData PumpパラメータRMTRAILを設定します。

```
RMTRAIL /mnt/dbfs/goldengate/dirdat/aa
```

- 4 ソース・ホスト上のローカル証跡ファイルの場所を使用して、Data Pumpプロセスを作成します。

```
% ggsci
```

```
GGSCI (ggtest.oracle.com) 1> ADD EXTRACT dpump_1, EXTTRAILSOURCE  
/mnt/dbfs/goldengate/dirdat/aa
```

- 5 ADD RMTRAILコマンドを使用して、ターゲット・ホスト上のリモート証跡ファイルの場所を指定します。

```
% ggsci
```

```
GGSCI (ggtest.oracle.com) 1> ADD RMTRAIL  
/mnt/dbfs/goldengate/dirdat/aa EXTRACT dpump_1, MEGABYTES 500
```

Data Pumpプロセスの作成方法については、『Oracle® GoldenGate Windows and UNIX管理者ガイド』を参照してください。

http://docs.oracle.com/cd/E35586_01/doc.1121/b69762.pdf

手順5：ローカル・ファイル・システムでのページ・ファイルのセットアップ

Oracle GoldenGateはコミットされたトランザクションのみをレプリケートするため、取得コンポーネントがトランザクションのコミットまたはロールバックを受け取るまで、各トランザクションの処理は仮想メモリ・プール（キャッシュ）内に保存されます。キャッシュがいっぱいになると、トランザクションはディスクにページアウトされます。デフォルトでは、このファイルはOracle GoldenGateインストール・ディレクトリのdirtmpサブディレクトリに保存されます。

ページ・ファイルの形式は、DBFSでは現在サポートされていないメモリ・マップ・ファイルであるため、DBFSに保存することはできません。データベース・サーバーのローカル・ファイル・システム上のOracle GoldenGateインストール・ディレクトリ内に、ページ・ファイルを保存します。

詳しくは、『Oracle® GoldenGate Windows and UNIXリファレンス・ガイド』のCACHEMGRの項を参照してください。

http://docs.oracle.com/cd/E35586_01/doc.1121/b69763.pdf

手順6：Replicatコミット動作の設定

Oracle GoldenGateバージョン11.2.1以降では、チェックポイント表が構成されていると、Replicatが自動的にCOMMIT NOWAITを使用して実行されます。トランザクションの適用時に各コミットでReplicatプロセスが待機しなくなるため、スループット・パフォーマンスが向上します。

チェックポイント表の作成方法については、『Oracle® GoldenGate Oracleインストレーションおよびセットアップ・ガイド』の第4章を参照してください。

http://docs.oracle.com/cd/E35586_01/doc.1121/b70206.pdf

Oracle GoldenGateバージョン11.1.1以前を使用している場合は、Replicatコミット動作をCOMMIT NOWAITに設定し、チェックポイント表の作成とは別に構成します。チェックポイント中のリカバリ・データ保護のため、このオプションの使用はチェックポイント表を使用している場合にのみ検討してください。

次のとおり、Replicatパラメータ・ファイルにCOMMIT NOWAITを設定します。

```
SQLEXEC "ALTER SESSION SET COMMIT_WRITE='NOWAIT'";
```

手順7：Extract、Data Pump、Replicatプロセスの自動開始の設定

Managerプロセスが開始されると、Extractプロセス、Data Pumpプロセス（使用している場合）、Replicatプロセスが自動的に開始されるように設定します。

次のパラメータをManagerパラメータ・ファイルに追加します。

```
AUTOSTART ER *
AUTORESTART ER *
```

必要に応じて、Oracle GoldenGateプロセス名のワイルドカード (*) を使用するのではなく、自動的に再起動したいプロセスに明示的に指定します。

例：

```
AUTOSTART EXTRACT EXT_1A
AUTOSTART EXTRACT DPUMP_1A
AUTORESTART EXTRACT EXT_1A
AUTORESTART EXTRACT DPUMP_1A
```

手順8：Oracle Clusterwareの構成

次のステップ・バイ・ステップの手順では、11.2.0.3 Oracle Grid Infrastructure Bundled Agentを使用して、Oracle ClusterwareでOracle GoldenGateを起動する方法を説明します。Grid Infrastructure 11.2.0.3より前のリリースを使用している場合は、付録Aの手順に従って、クラスタウェアでOracle GoldenGateリソースを作成して登録します。

1 アプリケーションVIPを作成します。

ソース・システムが同じOracle Exadata Database Machineの外部にあり、Oracle GoldenGate Data Pumpを使用して証跡ファイル・データをターゲット・データベース・マシンに送信している場合、どちらのノードがOracle GoldenGateをホストしているかに関係なく、リモートData Pumpがターゲット・データベース・マシンと通信できるようにするには、アプリケーションVIPが必要になります。

アプリケーションの仮想インターネット・プロトコル・アドレス (VIP) は、Oracle Clusterwareによって管理されるクラスターリソースです。VIPは1つのクラスターノードに割り当てられ、ノード障害が発生すると別のノードに移行されます。このため、Oracle GoldenGateのData Pumpは、新しく割り当てられたターゲット・ノードに対してデータ送信を続行できます。

同じデータベース・マシン内でExtractとReplicatの両方のプロセスが実行されており、Data Pumpが使用されていない場合、アプリケーションVIPを作成する必要はありません。手順2へ進み、エージェント・スクリプトを作成してください。それ以外の場合は、次の手順を実行してください。

- a. アプリケーションVIPを作成するには、rootユーザーとして次のコマンドを実行します。

```
$GRID_HOME/bin/appvipcfg create -network=1 ¥
    -ip=10.1.41.93 ¥
    -vipname=gg_vip_source ¥
    -user=root
```

上記の例において

- \$GRID_HOMEは、Oracle Database 11g Release 2 Grid InfrastructureコンポーネントがインストールされているOracleホームです（例： /u01/app/grid）。
- networkは任意のネットワーク番号です。Oracle Clusterwareリリース11.2.0.1では、次のコマンドを使用してネットワーク番号を確認できます。

```
crsctl stat res -p |grep -ie .network -ie subnet |grep -ie
name -ie subnet
```

次のサンプル出力を検討してみましょう。

```
NAME=ora.net1.network
USR_ORA_SUBNET=10.1.41.0
```

NAME=ora.net1.networkのnet1は、これがネットワーク1であることを示しています。2行目はVIPが作成されるサブネットを示しています。

- ipは、新しいアプリケーションVIPに対してシステム管理者が割り当てたIPアドレスです。このIPアドレスは、上記で特定されたものと同じサブネット内にある必要があります。
- gg_vip_sourceは、作成するアプリケーションVIPの名前です。

- b. 次のコマンドを実行して、VIPを開始するためのOracle Databaseインストール所有者権限を付与します。

```
$GRID_HOME/bin/crsctl setperm resource gg_vip_source -u
user:oracle:r-x
```

- c. Oracle Databaseのインストール所有者として、VIPリソースを開始します。

```
$GRID_HOME/bin/crsctl start resource gg_vip_source
```

- d. VIPが実行中であるかどうか、またどのノードで実行されているかを確認するには、次のコマンドを実行します。

```
$GRID_HOME/bin/crsctl status resource gg_vip_source
```

アプリケーションVIPの作成について、詳しくは次のOracle Clusterwareのドキュメントを参照してください。

http://docs.oracle.com/cd/E11882_01/rac.112/e16794/toc.htm

2 Oracle Grid Infrastructure Bundled Agentを構成します。

Oracle Grid Infrastructure Bundled Agentは64ビットLinuxの11.2.0.3で導入され、Oracle GoldenGate、Siebel、およびApacheアプリケーション用の事前定義クラスタウェアを提供します。Oracle GoldenGateにバンドルされたエージェントを使用すれば、ソース・データベース、ターゲット・データベース、アプリケーションVIP、およびDBFSマウント・ポイントの依存性を簡単に作成できます。エージェントのコマンドライン・ユーティリティ (agctl) を使用して、Oracle GoldenGateを起動および停止したり、クラスタ内のノード間でOracle GoldenGateを再配置したりすることができます。

Oracle GoldenGateにバンドルされたエージェントは、クラシック・キャプチャ・モードや統合キャプチャ・モードで実行されるExtractの使用を完全にサポートします。

現在のバージョンの認定マトリックスは次のとおりです。

Grid Infrastructure	Oracle GoldenGate	Oracle Database
11.2.0.3以降	11.2.1以降	10.2.0.5、11.2.0.2以降

バンドルされているエージェントのソフトウェアは、次の場所からダウンロードしてください。

<http://www.oracle.com/technetwork/jp/middleware/goldengate/downloads/index.html>

readme.txtファイルのインストール手順に従ってください。

バンドルされたエージェントの構成例は次のとおりです。

- a. MOS Note 1054431.1の説明に従ってDBFSを構成し、crsctlを使用してファイル・システムをマウントおよびアンマウントできることをテストで確認します。ファイル・システムをマウントすると、DBFSインスタンスが起動されます（まだ実行されていない場合）。

- b. agctlを使用して、Clusterwareリソースを作成します。

ソース環境では次のとおりです。

```
% agctl add goldengate GG_Source --gg_home
/home/oracle/goldengate ¥
--instance_type source ¥
--nodes dbm01db05,dbm01db06 ¥
--vip_name gg_vip_source ¥
--filesystems dbfs_mount --databases ggs ¥
--oracle_home /u01/app/oracle/product/11.2.0.3/dbhome_1 ¥
--monitor_extracts ext_1a,dpump_1a
```

ターゲット環境では次のとおりです。

```
% agctl add goldengate GG_Target --gg_home
/home/oracle/goldengate ¥
--instance_type target ¥
--nodes dbm03db01,dbm03db02 ¥
--vip_name gg_vip_target ¥
--filesystems dbfs_mount --databases ggt ¥
--oracle_home /u01/app/oracle/product/11.2.0.3/dbhome_1 ¥
--monitor_replicats rep_1a,rep_2a,rep_3a,rep_3b
```

Oracle GoldenGateプロセスをバンドル対象に追加したら、必ずagctlを使用して開始または停止します。バンドルしたエージェントでは、Managerの起動によりOracle GoldenGateプロセスが開始されます。これにより、自動開始を構成されているプロセス（Extract、Data Pump、Replicat）が自動的に開始されます。問題が発生してOracle GoldenGateプロセスが中断されても、Managerプロセスが実行されていれば、ggsciを使用して障害の発生したプロセスを再開できます。

Oracle GoldenGateの状態を確認するには、次のコマンドを実行します。

```
% agctl status goldengate GG_Source
Goldengate instance 'GG_Source' is running on dbm01db06
```

ggsciを使用して、Oracle GoldenGateの個々のプロセスのステータスを確認できます。Oracle GoldenGate Managerと自動開始が有効なすべてのプロセスを開始するには、次のコマンドを実行します。

```
% agctl start goldengate GG_Target [--serverpool serverpool_name  
| --node node_name]
```

注：ノード名やserverpool名を指定しない限り、Oracle GoldenGateは、コマンドの実行元のノードで起動されます。

すべてのOracle GoldenGateプロセスを停止するには、次のコマンドを実行します。

```
% agctl stop goldengate GG_Target [--serverpool serverpool_name  
| --node node_name]
```

Oracle GoldenGateを他のノードに再配置するには、次のコマンドを実行します。

```
% agctl relocate goldengate GG_Source [--serverpool serverpool_name  
| --node node_name]
```

注：Oracle GoldenGateリソースは、再配置の前に必ず実行してください。Oracle GoldenGateリソースの構成/パラメータを表示するには、次のコマンドを実行します。

```
% agctl config goldengate GG_Target
```

```
GoldenGate location is:/home/oracle/goldengate  
GoldenGate instance type is: source  
Configured to run on Nodes: dbm01db05 dbm01db06  
ORACLE_HOME location is:  
/u01/app/oracle/product/11.2.0.3/dbhome_1  
Databases needed: ggs  
File System resources needed: dbfs_mount  
Extracts to monitor: ext_1a,dpump_1a  
Replicats to monitor:
```

より詳細なクラスタウェア構成情報を表示するには、次のコマンドを実行します。

```
% crsctl stat res -w "NAME = xag.GG_Source.goldengate" -p
```

Oracle GoldenGateリソースを削除するには、次のコマンドを実行します。

```
% agctl remove goldengate GG_Source
```

Oracle Grid Infrastructure Bundled Agentについて詳しくは、以下を参照してください。

<http://www.oracle.com/technetwork/products/clusterware/overview/ogiba-reference-guide-v1-1844341.html>

付録A：Oracle GoldenGate Clusterware リソースの作成

Grid Infrastructureが11.2.0.3より前のリリースである場合、次の手順を使用して、Oracle GoldenGateをClusterwareのリソースとして登録できます。

- 1 エージェント・スクリプトを作成します（Bundled Agentを使用していない場合）。

Oracle Grid Infrastructure Bundled Agentを使用できない場合でも、Oracle Clusterwareを使用して、Oracle GoldenGateの自動起動および停止を管理できます。

Oracle Clusterwareでは、エージェントと呼ばれるエンティティ経由でリソース固有のコマンドが実行されます。エージェント・スクリプトには次のルールが適用されます。

- start、stop、check、clean、abortという5つのパラメータの値を受け取る必要があります。
- すべてのノードで同じ場所に格納されている必要があります。この例では、Grid Infrastructureの（\$GRID_HOME）ORACLE_HOME/crs/scriptディレクトリ内に保存されています。
- Oracle Grid Infrastructureユーザーによって所有されており、実行権限が付与されている必要があります。
- クラスタに含まれるすべてのノードの同じ場所からアクセスできる必要があります。
- 適切なプログラムの実行可能ファイルとOracle Net構成をCRSがみつけれられるように、ORACLE_HOME、ORACLE_SID、PATH、TNS_ADMIN、LD_LIBRARYパスに対する環境変数設定が含まれている必要があります。CRSがDBFSをマウントしようとした場合に、正しいsqlnet.ora、tnsnames.ora、またはdbfs_client実行可能ファイルが見つからない場合、マウントは失敗します。

Oracle GoldenGate Manager、Extract、Data Pump、およびReplicatプロセスを開始および停止するエージェント・スクリプト例については、付録Bを参照してください。

次の手順に進む前に、エージェント・スクリプトで、Oracle GoldenGateプロセスの開始および停止を手動でテストすることが重要です。

- 2 Oracle Clusterwareにリソースを登録します。

crsctlユーティリティを使用し、Oracle GoldenGateをリソースとしてOracle Clusterwareに登録します。DBFSを使用してOracle GoldenGateファイルを格納する場合は、（上記の）My Oracle Support Note 1054431.1の構成ガイドラインに従うことを推奨します。DBFSのマウントは、Clusterwareリソースによって実行され、DBFSインスタンスに開始依存性があります。DBFSリソースが起動されておりインスタンスが実行されていない場合は、インスタンスが自動的に開始されます。Oracle GoldenGateプロセスの開始前に必要なファイル・システムがマウントされるよう、DBFSマウント・リソースがOracle GoldenGateリソースの開始依存性として指定されます。

また、ExtractプロセスやReplicatプロセスがデータベースに接続できない場合にこれらのプロセスが失敗しないよう、ソース・データベースまたはターゲット・データベースをOracle GoldenGateリソースの開始依存性とすることを推奨します。

- 1 開始依存性のリソースとして、DBFS、ソース・データベース、またはターゲット・データベースの名前を特定するには、次のコマンドを実行します。

```
crsctl status resource | grep -i dbfs|source/target DB name
```

- 2 Oracle Grid Infrastructureユーザー（この例ではoracle）を使用して、次のコマンドを実行します。

```
$GRID_HOME/bin/crsctl add resource GG_Source ¥
    -type cluster_resource ¥
    -attr
    "ACTION_SCRIPT=/u01/app/11.2.0/grid/crs/script/11gr2_gg_action.scr,
    CHECK_INTERVAL=30,
    START_DEPENDENCIES='hard(gg_vip_source,dbfs_mount,ora.ggs.db)
    pullup(gg_vip_source)',      STOP_DEPENDENCIES='hard(gg_vip_source)',
    SCRIPT_TIMEOUT=300"
```

アプリケーションVIPが使用されていない場合は、次のコマンドを実行します。

```
$GRID_HOME/bin/crsctl add resource GG_Source ¥
    -type cluster_resource ¥
    -attr
    "ACTION_SCRIPT=/u01/app/11.2.0/grid/crs/script/11gr2_gg_action.scr,
    CHECK_INTERVAL=30,
    START_DEPENDENCIES='hard(gg_vip_source,dbfs_mount,ora.ggs.db)
    pullup(gg_vip_source)',      STOP_DEPENDENCIES='hard(gg_vip_source)',
    SCRIPT_TIMEOUT=300"
```

このホワイト・ペーパーでは、ソース（Extract）またはターゲット（Replicat）のいずれかのホストに対して1台のOracle Exadata Database Machineが使用されていることを想定しています。データベース・マシンが別々のクラスタに分かれており、ソースとターゲットが同じデータベース・マシン上で実行されている場合、ExtractプロセスとReplicatプロセスが指定されたクラスタ・ノードに限定されていることを確認します。

```
$GRID_HOME/bin/crsctl add resource GG_Source ¥
    -type cluster_resource ¥
    -attr
```

```
"ACTION_SCRIPT=/u01/app/11.2.0/grid/crs/script/11gr2_gg_action.scr,
CHECK_INTERVAL=30,
START_DEPENDENCIES='hard(gg_vip_source,dbfs_mount)
pullup(gg_vip_source)', STOP_DEPENDENCIES='hard(gg_vip_source)',
HOSTING_MEMBERS='dbm01db05 dbm01db06', PLACEMENT='restricted',
SCRIPT_TIMEOUT=300"
```

crsctl add resource コマンドとそのオプションについては、『*Oracle® Clusterware 管理およびデプロイメント・ガイド*』を参照してください。

http://docs.oracle.com/cd/E11882_01/rac.112/e16794/toc.htm

3 リソースを起動します。

リソースが追加された後は常に、Oracle Clusterwareを使用してOracle GoldenGateを起動します。Oracle Grid Infrastructureユーザー（oracle）としてログインし、次のコマンドを実行します。

```
% $GRID_HOME/bin/crsctl start resource GG_Source
```

アプリケーションの状態を確認するには、次のコマンドを実行します。

```
% $GRID_HOME/bin/crsctl status resource GG_Source
```

次に例を示します。

```
% crsctl status resource GG_Source
```

```
NAME=GG_Source
```

```
TYPE=cluster_resource
```

```
TARGET=ONLINE
```

```
STATE=ONLINE on dbm01db05
```

4 アプリケーションを管理します。

別のクラスタ・ノードにOracle GoldenGateを再配置するには、

'\$GRID_HOME/bin/crsctl relocate resource' APIを使用して、force (-f) オプションを含めます。Grid Infrastructureユーザー（oracle）としてログインしている場合、クラスタ内のどのノード上でもこのコマンドを実行できます。

アプリケーションVIPに依存性がある場合は、VIPリソースの再配置が必要です。これにより、Oracle GoldenGateが停止され、新しいノードで再配置および再起動されます。

次に例を示します。


```
[oracle@dbm01db05 ~]$ crsctl relocate resource gg_vip_source -f
CRS-2673:Attempting to stop 'GG_Source' on 'dbm01db05'
CRS-2677:Stop of 'GG_Source' on 'dbm01db05' succeeded
CRS-2673:Attempting to stop 'gg_vip_source' on 'dbm01db05'
CRS-2677:Stop of 'gg_vip_source' on 'dbm01db05' succeeded
CRS-2672:Attempting to start 'gg_vip_source' on 'dbm01db06'
CRS-2676:Start of 'gg_vip_source' on 'dbm01db06' succeeded
CRS-2672:Attempting to start 'dbfs_mount' on 'dbm01db06'
CRS-2676:Start of 'dbfs_mount' on 'dbm01db06' succeeded
CRS-2672:Attempting to start 'GG_Source' on 'dbm01db06'
CRS-2676:Start of 'GG_Source' on 'dbm01db06' succeeded
```

Oracle GoldenGateリソースを停止するには、次のコマンドを入力します。

```
$GRID_HOME/bin/crsctl stop resource GG_Source
```

5 CRSをクリーンアップします。

Oracle Clusterwareの管理対象からOracle GoldenGateを削除するには、次のタスクを実行します。

- a) Oracle GoldenGateを停止します（Oracle Grid Infrastructureユーザー（oracle）としてログインする）。

```
$GRID_HOME/bin/crsctl stop resource GG_Source
```

- b) rootユーザーとしてOracle GoldenGateリソースを削除します。

```
$GRID_HOME/bin/crsctl delete resource GG_Source
```

- c) 必要ない場合、エージェントのアクション・スクリプト11gr2_gg_action.scrを削除します。

これにより、Clusterwareリソースだけが削除されます。Oracle GoldenGateやDBFSの構成は削除されません。

Oracle RACへの配置に関する推奨事項

Oracle RAC環境にOracle GoldenGateが構成されている場合は、次の推奨事項に従ってください。

- Oracle RAC構成に含まれるすべてのデータベース・ノードで、DBFSデータベースのインスタンスが実行されるようにします。

これによって、ノード障害後にOracle GoldenGateが再起動された場合にOracle GoldenGateへのアクセスが提供されます。

- Oracle RAC構成に含まれるすべてのデータベース・ノードで、DBFSファイル・システムがマウント可能であることを確認します。Clusterwareリソースを使用してDBFSをマウントして、ExtractプロセスまたはReplicatプロセスが複数のノードで同時に開始されないようにし、Oracle GoldenGateが実行されているノードでのみファイル・システムをマウントします。シームレスなフェイルオーバーを実現するため、すべてのノード上で同じマウント・ポイント名を使用します。

付録B：エージェント・スクリプトの例

次のエージェント・スクリプトの例では、起動時とフェイルオーバー時にDBFSファイル・システムをマウントおよびアンマウントするとともに、Oracle GoldenGate Manager、Extract、Replicatのプロセスを開始および停止しています。このエージェント・スクリプトでは、パラメータ値start、stop、check、clean、およびabortを使用できます。

```
#!/bin/bash
#llgr2_gg_action.scr

# Edit the following environment variables:
# NOTE: The ORACLE_SID will be different on each node
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.3/dbhome_1
export ORACLE_SID=GG1
GG_HOME=/home/oracle/goldengate/latest
export PATH=$ORACLE_HOME/bin:$PATH
export TNS_ADMIN=$ORACLE_HOME/network/admin

#Include the GoldenGate home in the library path to start GGSCI
export LD_LIBRARY_PATH=${ORACLE_HOME}/lib:${LD_LIBRARY_PATH}:${GG_HOME}

# Edit this to indicate the DBFS mount point
DBFS_MOUNT_POINT=/mnt/dbfs

# Edit this to indicate at least one of the file systems mounted in
# the DBFS mount point
DBFS_FILE_SYSTEM=/mnt/dbfs/goldengate

# Edit for correct Extract/Datapump name if running this script on
# the source (wildcard is appended to these names in the script):
EXTRACT=EXT
DATAPUMP=DPUMP

# Edit for current Replicat names if running script on the target
# (wildcard is appended to these names in the script):
REPLICAT=REP

# Specify delay after start before checking for successful start
start_delay_secs=5

### Syslog facility name (default user)
```

```
### Changed default from local3 to user for Solaris default support on 17-
FEB-2012
### This will allow us to log messages to the syslog
### (/var/log/messages on Linux, /var/adm/messages on Solaris)
LOGGER_FACILITY=user

#####
### No editing is required below this point
#####
### determine platform
UNAME_S=`uname -s`
if [ $UNAME_S = 'Linux' ]; then LINUX=1; SOLARIS=0;
elif [ $UNAME_S = 'SunOS' ]; then LINUX=0; SOLARIS=1;
fi

LOGGER="/bin/logger -t GoldenGate"

logit () {
    ### type: info, error, debug
    type=$1
    msg=$2
    if [ "$type" = "info" ]; then
        echo $msg
        $LOGGER -p ${LOGGER_FACILITY}.info "$msg"
    elif [ "$type" = "error" ]; then
        echo $msg
        $LOGGER -p ${LOGGER_FACILITY}.error "$msg"
    elif [ "$type" = "debug" ]; then
        echo $msg
        $LOGGER -p ${LOGGER_FACILITY}.debug "$msg"
    fi
}

# check_process validates that Manager/Extract/Replicat process is
# running #at PID that GoldenGate specifies.
check_process () {
    PROCESS=$1
    if [ ${PROCESS} = mgr ]
    then
        PFILE=MGR.pcm
    fi
}
```

```

elif [ ${PROCESS} = ext ]
then
    PFILE=${EXTRACT}*.pce
elif [ ${PROCESS} = rep ]
then
    PFILE=${REPLICAT}*.pce
else
    PFILE=${DATAPUMP}*.pce
fi
if ( [ -f "${GGS_HOME}/dirpcs/${PFILE}" ] )
then
    pid=`cut -f8 "${GGS_HOME}/dirpcs/${PFILE}"`
    if [ ${pid} = `ps -e |grep ${pid} |grep ${PROCESS} |cut -d " " -f2` ]
    then
        logit info "${SCRIPTNAME}(check_process) - Procese(s) ${PROCESS} IS
running"
        exit 0
    else
        if [ ${pid} = `ps -e |grep ${pid} |grep ${PROCESS} |cut -d " " -f1` ]
        then
            logit info "${SCRIPTNAME}(check_process) - Procese(s) ${PROCESS} IS
running"
            exit 0
        else
            logit error "${SCRIPTNAME}(check_process) - Procese(s) ${PROCESS} is
NOT running"
            exit 1
        fi
    fi
else
    logit error "${SCRIPTNAME}(check_process) - Procese(s) ${PROCESS} is NOT
running - no pid file"
    exit 1
fi
}

# call_ggsci is a generic routine that executes a ggsci command
call_ggsci () {
    ggsci_command=$1
    ggsci_output=`${GGS_HOME}/ggsci << EOF
        ${ggsci_command}
    exit

```

```
EOF`
}

stop_everything () {
# Before starting, make sure everything is shutdown and process files are
removed
#attempt a clean stop for all non-manager processes

logit info "${SCRIPTNAME}(stop_everything) - Stopping all processes"

call_ggsci 'stop er *'

#ensure everything is stopped
call_ggsci 'stop er *!'

#in case there are lingering processes
call_ggsci 'kill er *'

#stop Manager without (y/n) confirmation
call_ggsci 'stop manager!'

#Remove the process files:
rm -f $GGS_HOME/dirpcs/MGR.pcm
rm -f $GGS_HOME/dirpcs/*.pce
rm -f $GGS_HOME/dirpcs/*.pcr
}

case $1 in
'start')
# stop all GG processes and remove process files

logit info "${SCRIPTNAME} - start - Starting all processes"

stop_everything
sleep ${start_delay_secs}

#Now can start everything...
#start Manager
logit info "${SCRIPTNAME} - start - Starting Manager, autostarting
processes"
call_ggsci 'start manager'
```

```
#there is a small delay between issuing the start manager command
#and the process being spawned on the OS <96> wait before checking
sleep ${start_delay_secs}

#start Extract or Replicats
call_ggsci 'start er *'

#check whether Manager is running and exit accordingly
logit info "${SCRIPTNAME} - start - Checking Manager status"

check_process mgr
sleep ${start_delay_secs}

#Check whether Extract is running
logit info "${SCRIPTNAME} - start - Checking GoldenGate statuses"

check_process ext
check_process dpump
;;
'stop')
# stop all GG processes and remove process files
logit info "${SCRIPTNAME} - stop - Stopping all processes"

stop_everything

#exit success
exit 0
;;
'check')
logit info "${SCRIPTNAME} - check - Checking all processes"

check_process mgr
check_process ext
check_process dpump
check_process rep
;;
'clean')
# stop all GG processes and remove process files
logit info "${SCRIPTNAME} - clean - Stopping all processes"

stop_everything
```

```
#exit success
exit 0
;;
'abort')
# stop all GG processes and remove process files
logit info "${SCRIPTNAME} - abort - Stopping all processes"

stop_everything

#exit success
exit 0
;;
esac
```

参考資料

- Oracle® GoldenGate Windows and UNIX管理者ガイド11gリリース2 (11.2.1.0.1)
- Oracle GoldenGate Oracleインストールおよびセットアップ・ガイドリリース11.2.1.0.1
- Oracle® GoldenGate Windows and UNIXリファレンス・ガイド11gリリース2 (11.2.1.0.1)
- Oracle Database SecureFilesおよびラージ・オブジェクト開発者ガイド (DBFS関連の箇所を参照)
- Oracle Clusterware管理およびデプロイメント・ガイド

Oracle Maximum Availability ArchitectureのWebサイト

<http://www.oracle.com/technetwork/jp/database/features/availability/maa-094615-ja.html>



Oracle Exadata Database Machineでの
Oracle GoldenGateの構成

2013年1月

著者：Stephan Haisley

共著者：MAAチーム

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：
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200
www.oracle.com



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

Copyright © 2013, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

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

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

Hardware and Software, Engineered to Work Together