

Oracle Maximum
Availability Architecture

Oracle Database In-Memory

高可用性ベスト・プラクティス

Oracle ホワイト・ペーパー | 2015 年 9 月





目次

はじめに	2
ベスト・プラクティス	2
Oracle Database In-Memory の MAA で推奨されているソフトウェア最適化を適用する	2
高可用性に対応した接続を構成および管理する	3
Oracle RAC 環境での Oracle Database In-Memory	3
DUPLICATE および DUPLICATE ALL を使用する場合のベスト・プラクティス	5
FAN マネージド・サービス	5
セッション・パラメータの設定	7
パーティション交換の前に、表がインメモリに格納されるように構成する	11
標準の MAA ベスト・プラクティスに従う	12
In-Memory 表を使用した MAA 停止テスト	12
停止テスト・マトリックス	12
結論	13
参考資料	14

はじめに

Oracle Database In-Memory は分析問合せのパフォーマンスを劇的に向上させます。Oracle Database In-Memory を実装するとインメモリ列ストアが Oracle Database にシームレスに統合されるため、Maximum Availability Architecture (MAA) から得られる高可用性のあらゆるメリットが継承されます。Oracle Database In-Memory は、<http://www.oracle.com/technetwork/jp/database/features/availability/maa-094615-ja.html> に掲載されている高可用性に関するドキュメントや MAA ホワイト・ペーパーに書かれている一般的な MAA ベスト・プラクティスに従って構成すると、インスタンス障害やノード障害に強く、データを破損から保護し、破損したデータは迅速に修復し、クラスタ障害やデータベース障害や災害が発生した場合は適切なタイミングでフェイルオーバーし、データベースの停止時間をほぼゼロに抑えながらさまざまな計画保守作業に対応できます。このホワイト・ペーパーでは、計画保守作業時や障害発生時の Oracle Database In-Memory の高可用性とパフォーマンスの高いアプリケーション・サービスを実現するために、その他のベスト・プラクティスを紹介します。

ベスト・プラクティス

次のベスト・プラクティスに準拠することで、可用性の高い Oracle Database In-Memory を実装できます。

- » Oracle Database In-Memory の MAA で推奨されているソフトウェア最適化を適用する
- » 高可用性に対応した接続を構成および管理する
- » エンジニアド・システム上のすべてのノードにインメモリ列ストアを複製する
- » パーティション交換の前に、表がインメモリに保持されるように構成する
- » 標準の MAA ベスト・プラクティスに従う

このホワイト・ペーパーで紹介するベスト・プラクティスは、Oracle Real Application Clusters (Oracle RAC) データベースに関する推奨事項を実装することを前提にしています。Oracle RAC は高可用性のサポートに必要なクラスタ・サービスとデータベース・サービスをアプリケーションに提供するソリューションであり、これから説明するベスト・プラクティスの基盤でもあります。

Oracle Database In-MemoryのMAAで推奨されているソフトウェア最適化を適用する

最新の Oracle Database In-Memory MAA ソフトウェアの前提条件および推奨事項については、My Oracle Support (MOS) 2045279.1 を参照してください。

高可用性に対応した接続を構成および管理する

高可用性戦略には、クラスタのリソースが使用不能になったときやオンラインに戻ったときに、クライアント・セッションに通知する手段が含まれている必要があります。Oracle には高速アプリケーション通知 (FAN) 機能と高速接続フェイルオーバー (FCF) 機能があり、インスタンス障害やデータベース障害が発生した場合、データベースが起動した場合やロール変更があった場合に、クライアントにすみやかに通知することができます。さらに、Oracle Universal Connection Pool (UCP) などの接続プールを使用することで、実行時ロードバランシングやアプリケーション・コンティニューイティといった機能を追加することができます。

Oracle RAC環境でのOracle Database In-Memory

Oracle Database In-Memory を使用すると、新しいインメモリ列形式でデータをメモリに移入できます。このとき、In-Memory 領域と呼ばれる、Oracle Database のシステム・グローバル領域 (SGA) の新しいコンポーネントであるインメモリ列ストア (IM 列ストア) が使用されます。IM 列ストア内のデータは、Oracle Database で使用される従来の行形式では格納されません。代わりに、IM 列ストア内のデータには新しい列形式が使用されます。ディスクの表領域が複数のエクステンツで構成されるのと同様に、IM 列ストアは複数の In-Memory 圧縮ユニット (IMCU) で構成されます。

INMEMORY 属性が指定されているオブジェクトのみ IM 列ストアに移入されます。INMEMORY 属性の PRIORITY 副次句を NONE 以外の値に設定すると、表は自動的に IM 列ストアにロードされます。そのため、データベース・インスタンスが起動するとすぐに表がメモリに移入されます。PRIORITY 副次句を指定しないとデフォルトの NONE になり、初めてアクセスされるまでオブジェクトは IM 列ストアに移入されません。

Oracle RAC 環境ではノードごとに専用の IM 列ストアが設定されます。INMEMORY 属性が指定されている表は、デフォルトでクラスタ内のすべてのノードに自動的に分散されます。クラスタにオブジェクトを分散させる方法は DISTRIBUTE 副次句で制御します。デフォルトでは、使用されているパーティション化の種類に応じて (使用されている場合)、Oracle が最適な分散方法を決定します。データはなるべく Oracle RAC ノード全体に均等に分散されるようにする必要があります。(サブ)パーティションのデータに偏りがある場合は、DISTRIBUTE BY ROWID RANGE 指定して、行 ID の範囲ごとに分散することを推奨します。

Oracle RAC 環境でインメモリに移入されたデータには特定の Oracle RAC ノードへのアフィニティが設定されるため、パラレル・サーバー・プロセスを使用して、各 Oracle RAC ノードでそのノードの IM 列ストアに配置されているオブジェクトの断片に対して問合せを実行する必要があります。エンドユーザーのセッションに結果を返す前に、問合せコーディネータによって各パラレル・サーバー・プロセスの結果がまとめて集計されます。パラレル・サーバー・プロセスが Oracle RAC クラスタ全体に適切に分散されるようにするには、自動並列度 (AutoDOP) を使用し、データ・アフィニティまたはデータの場所が問合せコーディネータに認識されるようにする必要があります。

AutoDOP を有効化するには `parallel_degree_policy` パラメータを `AUTO` に設定します。AutoDOPを使用すると、問合せに使用する並列度はオプティマイザによって自動的に決定されます。表の修飾やヒントを使用するとオブジェクトまたは SQL 文の並列度を手動で指定できますが、AutoDOP の設定とこれらを取り違えないでください。

Oracle RAC ノードで障害が発生した場合はデータの一部分がインメモリに存在しなくなるため、バッファ・キャッシュ、フラッシュまたはディスクからの読取りが必要になります。その結果、データの欠落部分を残りのノードのインメモリ列ストアに移入できる状態になるまで、応答時間が長くなる場合があります。

Exadata や SuperCluster などの Oracle エンジンアド・システム上で Oracle Database In-Memory を実行している場合で、高可用性でなおかつ応答時間の予測値が一定していることが重要な場合は、ノード間でインメモリ表をミラー化し、アプリケーションで必要となるデータはすべてインメモリにすでに格納されているためノードが停止しても再移入する必要がない状態にし、存続しているインスタンスにデータが再移入されるまでパフォーマンスが低下することがないようにする必要があります。

そのためには、次のように INMEMORY 属性の `DUPLICATE` 副次句を指定します。

```
SQL> ALTER TABLE customers INMEMORY PRIORITY NONE DUPLICATE ;
```

こうすると、表の各断片 (IMCU) のミラー化されたコピーが、クラスタに含まれる別のノードのインメモリに格納されます。なお、`DUPLICATE` を使用する上のコマンドでは `PRIORITY NONE` を指定しましたが、これはトリガー問合せによってオンデマンドでのみ表が移入されるようにするためであり、望ましいノードで再移入が実行されるようにするためです。

複数のインスタンス障害から保護する場合は、INMEMORY 属性の `DUPLICATE ALL` 副次句を指定して、クラスタ内の各ノードの IM 列ストアにオブジェクトを移入することができます。これにより、問合せを単一ノード内で完全に実行できるようになるため、最高レベルの冗長性と直線的なスケラビリティを実現できます。

```
SQL> ALTER TABLE customers INMEMORY DUPLICATE ALL;
```

これで、クラスタ内のすべてのノードに表が複製されるようになり、この表に対する DML はすべてのノードの IM 列ストアに同時に反映されることとなります。すべてのノードですべての IMCU を使用できるため、IMCU とパラレル・スレーブとの間のアフィニティを確保するために AutoDOP を指定する必要がなくなります。したがって、`parallel_degree_policy` はデフォルトの `MANUAL` のままにしておくことができます。`DUPLICATE ALL` 副次句を指定すると、パラレル・スレーブは操作対象のインスタンス上で必ず IMCU を検出することになるため、ヒントまたは表のプロパティを使用して並列度を手動で制御することも可能です (オプティマイザで決定することもできます)。

エンジンアド・システム以外で `DUPLICATE` または `DUPLICATE ALL` 副次句を使用しても何も効果はありません。

DUPLICATEおよびDUPLICATE ALLを使用する場合のベスト・プラクティス

DUPLICATE オプションおよび DUPLICATE ALL オプションを使用する場合のベスト・プラクティスに必要なのは、FAN マネージド・サービスを使用することと、セッション・パラメータを正しく設定することです。

FANマネージド・サービス


クライアントの FAN を有効化し、サービスを正しく構成することが、計画停止や計画外停止が発生した場合のアプリケーションへの影響を最小化するための重要な要素です。DUPLICATE または DUPLICATE ALL を使用する場合は、再移入時のアプリケーションへの影響（“一時停止”）を回避するために、再起動が完了してデータを再移入しようとしているインスタンスにクライアントが接続できないようにすることを推奨します。そのためには、停止したインスタンスのサービスを無効にして停止します。このホワイト・ペーパーと下の表 1 では、これを“マネージド”接続サービスと呼びます。サービスを管理するために必要な手順は次のとおりです。

1. たとえば、障害を起こしたインスタンス上の、インメモリ列ストアを使用する分析クライアント向けサービスを、SRVCTL を使用して停止および無効化します。

```
$ srvctl stop service -service prod_dbim_fan -database dbm -
instance dbm1
$ srvctl disable service -service prod_dbim_fan -database dbm -
instance dbm1
```

2. インスタンスまたはノードで保守作業を実行します。
3. 新しく再起動されたインスタンスで IM 列ストアへの移入が実行されていることを確認し、すべての表の移入が完了するまで待機します。再移入をトリガーするには、In-Memory 属性を設定した表に対して全表スキャンを実行します。この問合せにより IM 列ストアの再移入がトリガーされるため、これをトリガー問合せと呼びます。トリガー問合せを実行するセッションには、parallel_instance_group を正しく設定する必要があります（表 1 および下の「セッション・パラメータの設定」を参照してください）。

次の問合せを使用すると、すべてのインスタンスについて、移入が完了していない表またはパーティションのバイト数を監視できます。



```
SELECT v.inst_id, v.segment_name name, v.partition_name
, v.populate_status status
, v.bytes/1024/1024 bytes_mb
, v.bytes_not_populated/1024/1024 bytes_not_pop_mb
, (v.bytes-v.bytes_not_populated)/1024/1024 bytes_populated_mb
FROM gv$im_segments v
WHERE v.owner = '&owner_name'
ORDER BY 1
```

DUPLICATE ALL を使用している場合は、セグメント全体を In-Memory に設定していると、列 "bytes_not_populated" は 0 (ゼロ) になります。DUPLICATE 副次句を使用している場合は、次の問合せを使用してください。

```
SELECT v.segment_name name, v.partition_name,
2*max(v.bytes)/1024/1024 tot_mirrored_bytes_mb
, sum(v.bytes-v.bytes_not_populated)/1024/1024 tot_bytes_populated_mb
FROM gv$im_segments v
WHERE v.owner = '&owner_name'
GROUP BY v.segment_name, v.partition_name
ORDER BY 1
```

セグメント全体を In-Memory に設定している場合は、TOT_MIRRORED_BYTES_MB 列と TOT_BYTES_POPULATED_MB 列が等しくなります。

4. SRVCTL を使用して、新しく再起動したインスタンスの In-Memory を使用するサービスを有効化して起動します。

```
$ srvctl enable service -servcpe imquery -database dbm -instance
dbm1
$ srvctl start service -service imquery -database dbm -instance
dbm1
```

5. 接続プールによってロードバランシングが行われる、または新しい接続が確立され、クライアントで新しいインスタンスの使用が始まります。

利便性と効率性のために、上記手順の一部はスクリプトを使用して自動化することができます。MOS note 1927000.1 には、インスタンスまたはノードが停止したときに自動的にサービスが無効化されるように Oracle RAC 環境で FAN をコールアウトするスクリプトの作成方法が掲載されています。

セッション・パラメータの設定

DISTRIBUTE 副次句を指定すると、IM 列ストアに移入されたデータと特定のノードの間にアフィニティが設定されるため、パラレル実行を使用するセッションごとに *parallel_degree_policy = AUTO* を設定し、インメモリにデータを含む適切なノードにプロセスが送信されるようにする必要があります。これは、DUPLICATE 副次句を指定した場合にも該当します。*parallel_degree_policy=AUTO* を使用すると高可用性を維持したままメモリ使用量を減らせるため、この設定がアプリケーションで許容される場合は DUPLICATE を使用することを推奨します。

DUPLICATE ALL オプションを指定すると、データの完全なコピーが各インスタンスに移入されます。そのため、パラレル実行プロセスがどのノードで動作していてもデータは必ずパラレル実行プロセスによって検出されることになるため、セッションの *parallel_degree_policy* を MANUAL、LIMITED、または AUTO に設定しても依然としてインメモリ列ストアのメリットが得られます。

パラレル実行の対象にするインスタンスは、*parallel_instance_group* パラメータを使用して制御します。*parallel_instance_group* パラメータに割り当てられ、アプリケーションのセッションによって使用されるサービスは、接続に使用されるサービス（前の項で説明した、マネージド接続サービスと呼ばれるサービス）とは異なる場合があります。このようにするのは、再移入時にパラレル実行プロセスのアフィニティが適切に維持されるようにするためです。接続と *parallel_instance_group* パラメータに異なるサービスを使用すべき場合についての具体的な推奨事項は、表を参照してください。

表1：DUPLICATE設定/DUPLICATE ALL設定の構成

複製モード	並列度のポリシー	表または問合せのDOP設定	アプリケーションに使用されるパラレル・インスタンス・グループ・サービス	再移入時に使用されるパラレル・インスタンス・グループ・サービス	利点
DUPLICATE	AUTO	AUTO*	すべてのインスタンスを含み、接続サービスとは異なる	すべてのインスタンスを含み、接続サービスとは異なる	<ul style="list-style-type: none"> 単一インスタンス障害の影響がほとんどない IMの能力が最大化される
DUPLICATE	LIMITEDまたはMANUAL	非推奨 - PQのアフィニティ設定は不可			
DUPLICATE ALL	AUTO、LIMITED、またはMANUAL	AUTOまたはMANUAL	“マネージド”接続サービスと同じ	すべてのインスタンスを含み、接続サービスとは異なる	<ul style="list-style-type: none"> 複数のインスタンス障害への耐性 <i>parallel_degree_policy = LIMITED</i>または<i>MANUAL</i>の使用が可能

*パラメータ *parallel_degree_policy* を AUTO に設定する場合は、表または問合せの DOP を設定しないでください。使用する DOP については、オプティマイザに全面的に任せる必要があります。

2つの例で、上の表の推奨設定を説明します（いずれの例も、フルラックのExadataエンジニアド・システムでOracle RACを使用しているものとします）。

1. Exadata フルラック・システムで DUPLICATE を使用する例

» マネージド接続サービスでは `prod_dbim_fan` サービスを使用

- アプリケーション・クライアントがこのサービスに接続します。停止時は（前述のとおり）カスタム・スクリプトでこのサービスを変更し、停止しているインスタンスにクライアントが接続しないようにします。インスタンスが再度使用可能になり、再起動されたインスタンス上での再移入が完了したら、サービスは変更されます。
- サービスは、停止しているノードがない限り8つのデータベース・ノードのいずれかに接続を送信するように構成されています。停止状態になったらすぐに、障害を起こしたインスタンスのサービスはマネージド接続サービスによって無効化および停止されます。実行中の問合せには影響が及ぶ場合がありますが、新しい接続には、そのサービスに対応している残りのインスタンスが使用されます。
- IM列ストアに配置されている表にはDUPLICATEオプションが使用されるように設定されているため、8つのインスタンスに2つのIMCUのコピーが存在します。これにより、In-Memoryオブジェクト用のメモリの使用効率が改善します。

» 再移入サービスでは `prod_dbim` サービスを使用

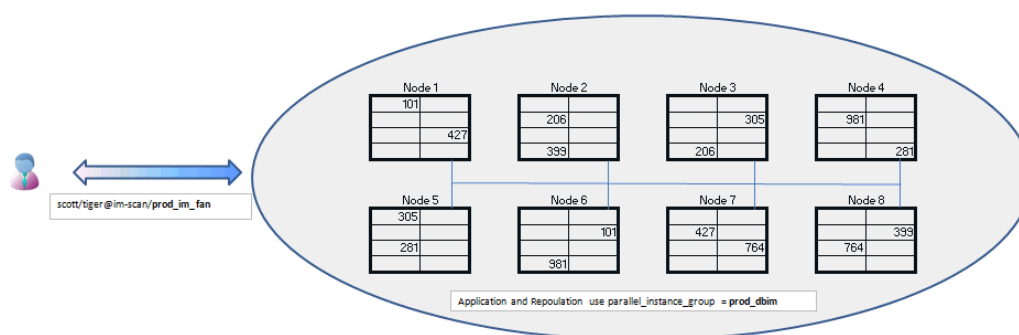
- これは、クラスタ内の8つのインスタンスすべてが含まれるように定義されています。つまり、すべてのインスタンスがパラレル問合せの対象になっています。このサービスは停止時も変更されません。
- IM列ストアを再移入するために、セッションの`parallel_instance_group`値が`prod_dbim`になり、インスタンスが利用可能になった後、新しく再起動されたインスタンス上でトリガー問合せが実行されます。これにより、クラッシュの前に保持されていたのと同じIMCUがIM列ストアに含まれるように（クラッシュ後に発生した可能性のある変更に合わせて調整され）、再起動されたインスタンス上で再移入が実行されます。

» アプリケーションではデータベース・セッションに次の設定を使用します。

- `parallel_degree_policy= AUTO`（スレーブとIMCUの間にアフィニティを設定するためにDUPLICATEとの併用が必要）
- `parallel_instance_group = prod_dbim`（クライアント接続に使用されるサービスとは別）

- » IM 列ストアの再移入が完了したら、再起動されたインスタンスに対してマネージド接続サービスが有効化および起動されます。
- » 接続サービスの有効化と起動が完了したら、新しく起動したインスタンスで接続が再開されます。

下の図では IMCU を番号で表しています。番号が同じ IMCU は互いのコピーです。クラスタ内には各 IMCU のプライマリ・コピーとセカンダリ・コピーがありますが、Oracle による読取りは、ミラー化されているいずれかの側からいつでも実行できます。



この図で示しているクライアントは"prod_im_fan"サービスで接続しています。このサービスはマネージド・サービスであるため、障害を起こしたインスタンスはサービスを提供できなくなります。アプリケーションでは、parallel_instance_group パラメータに"prod_dbim"サービスを使用しています。

障害を起こしたインスタンスの IM 列ストアの再移入のためにトリガー問合せを実行するセッションも、"prod_dbim"サービスを使用します。アプリケーションでは parallel_degree_policy=AUTO を設定し、パラレル実行プロセスとインスタンスの間にアフィニティが正しく確立されるようにする必要があります。

2. Exadata フルラック・システムで DUPLICATE ALL を使用する例

- » In-Memory 表に DUPLICATE ALL オプションが設定されているため、各ノードにすべての IMCU のコピーが含まれることとなります。したがって、メモリ使用量は増えますが、複数のノードが失われた場合でもアプリケーションのパフォーマンスを低下させずにクラスタを稼働し続けられます。また、アプリケーションでは必要に応じて parallel_degree_policy を AUTO ではなく MANUAL に設定することができます。

» マネージド接続サービスでは `prod_dbim_fan` サービスを使用

- アプリケーション・クライアントはこのサービスで接続します。前の例と同様に、停止時はサービスが変更され、再移入が完了するまで、クライアントは停止しているインスタンスに接続できなくなります。インスタンス障害の発生後、障害を起こしたインスタンス上で稼働していたサービスは、マネージド接続サービスによってただちに無効化および停止されます。実行中の問合せには影響が及ぶ場合がありますが、新しい接続には、障害を起こしたインスタンスのIMCUのミラー化コピーを保持している残りのインスタンスが使用されます。
- アプリケーションはマネージド接続サービスに設定された `parallel_instance_group` を使用するため、再移入中に、新しく再起動されたインスタンスでパラレル実行プロセスが起動されることはありません。

» 再移入サービスでは `prod_dbim` サービスを使用

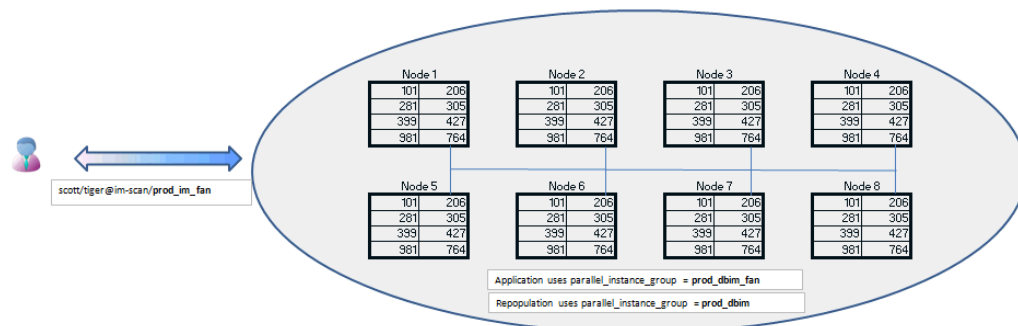
- これには、パラレル問合せの対象にしたいすべてのインスタンスが含まれます。このサービスは停止時も変更されません。IM列ストアを再移入するには、`parallel_instance_group` 値を `prod_dbim` に設定したセッションを使用して、新しく再起動され再度使用可能になったインスタンス上でトリガー問合せを実行する必要があります。このサービスには、再移入が必要なものも含めたすべてのインスタンスが含まれます。アプリケーションでは、`parallel_instance_group` (`prod_dbim_fan`) としてマネージド接続サービスを使用し続けますが、これは、再移入を実行するとき、使用可能なインスタンス上のIMCUとパラレル実行プロセスの間にアフィニティが確立されるようにするためです。

» アプリケーションではデータベース・セッションに次の設定を使用します。

- `parallel_degree_policy = AUTO` または `MANUAL`
- 問合せまたは表のDOPを `AUTO` または `MANUAL` に設定可能
- `parallel_instance_group = prod_dbim_fan` (これは接続サービスと同じです)

IM列ストアの再移入が完了したら、再起動されたインスタンスに対してマネージド接続サービスが有効化および起動されます。新しく再起動されたインスタンスで接続が再開されます。

下の図では IMCU を番号で表しています。番号が同じ IMCU は互いのコピーです。各ノードにすべての IMCU が保持されていることがわかります。



この図で示しているクライアントは"prod_im_fan"サービスで接続しています。このサービスはマネージド・サービスであるため、障害を起こしたインスタンスはサービスを提供できなくなります。

アプリケーションは parallel_instance_group パラメータに"prod_dbim_fan"サービスを使用し、再移入中のインスタンス上でパラレル実行プロセスが実行されないようにします。

アプリケーションでは parallel_degree_policy=AUTO、MANUAL、または LIMITED を設定できます。

障害を起こしたインスタンスの IM 列ストアに再移入するためにトリガー問合せを実行するセッションでは、"prod_dbim"サービスを使用します。

エンジニアド・システムを使用していない場合は、DUPLICATE または DUPLICATE ALL に設定しても In-Memory 高可用性機能のメリットは得られませんが、Oracle RAC システムで parallel_degree_policy を AUTO に設定した場合は、パラレル問合せを使用することでパフォーマンス上および能力上のメリットが得られます。

パーティション交換の前に、表がインメモリに格納されるように構成する

Oracle Database In-Memory は、表レベルだけでなくパーティション・レベルでも機能します。パーティション交換を実行する場合は、まず、パーティション表に交換する非パーティション表に In-Memory 属性を指定し、表が IM 列ストアに移入されたことを確認することを推奨します。パーティションが交換されたときにはデータがすでにインメモリにあり使用できる状態になっているため、データが移入されるまで待機する必要がなくなります。IM 列ストアに移入されている表のパーティションは、それまで非 In-Memory 表で実行できていたのと同様に、ワークロードを静止させなくてもいつでも交換できます。下のコマンド例では、先にスタンドアロンの表、lineorder_tab を変更してインメモリ属性を設定し、表がインメモリに移入されたことを確認してから次のコマンドを使用してパーティション交換の操作を実行します。

```
ALTER TABLE lineorder EXCHANGE PARTITION p1 WITH TABLE lineorder_tab;
```

標準のMAAベスト・プラクティスに従う

ここまでに紹介したベスト・プラクティスは、OTN (www.oracle.com/jp/goto/maa) にある高可用性のベスト・プラクティス・ドキュメントや MAA に関連するその他のホワイト・ペーパーです。すでに公開されている MAA のベスト・プラクティスに追加して適用するものです。

In-Memory表を使用したMAA停止テスト

停止テスト・マトリックス

次のマトリックスは、In-Memory 表を使用しているクライアントについてテストしたさまざまな停止シナリオとクライアント・アプリケーションで観察された影響をまとめたものです。

IN-MEMORY表を使用した場合のMAA停止マトリックス

停止	停止のシミュレーション・プロセス	アプリケーションへの影響と観察事項
DBノードの計画保守作業	<ol style="list-style-type: none"> すべてのインスタンスのインメモリリストアが事前にロードされている状態でワークロードを起動 1つのインスタンスで正常にサービスを停止 影響を受けたインスタンスのサービスを無効化 すべての接続が他のノード上のインスタンスに移動されたことを確認 インスタンスまたはノードを停止 ノード上で保守作業を実行 インスタンスまたはノードを再起動 問合せを実行し、再起動されたインスタンスのインメモリリストアをロード 影響を受けたインスタンスですべての表がIn-Memoryにロードされるまで待機 影響を受けたインスタンスのサービスを有効化 影響を受けたインスタンスのサービスを起動 再起動されたインスタンスを使用しているクライアントを観察 	<ul style="list-style-type: none"> 再構成時のOracle RAC内のアプリケーションへの影響は最小限（ただし、保守作業時にインスタンスの停止と再起動を要求した場合）
サービスの強制停止	<ol style="list-style-type: none"> アプリケーションにアプリケーション・コンティニューイティが構成されていて、ORA-12805エラーまたはORA-40エラーの後にエラーを捕捉して再試行できることを確認 ワークロードを起動 1つのインスタンスで強制的にサービスを停止 同じインスタンスでサービスを再起動 再起動されたサービスに徐々にロードが戻ることを観察 	<ul style="list-style-type: none"> 次の理由により、障害発生時に一時的に応答時間のスパイクが発生： <ul style="list-style-type: none"> 問合せの中断 アプリケーション・コンティニューイティが使用される際の問合せの再実行/再現 インスタンス障害やインスタンス再起動は発生していないため、インメモリリストアの再ロードは不要
データベース・インスタンス障害	<ol style="list-style-type: none"> ORA-12805エラーまたはORA-40エラーの後にこれらを捕捉して再試行できることを確認 ワークロードを起動 1つのインスタンス上でSMONをkill 影響を受けたインスタンスのサービスをすみやかに無効化 インスタンスが再起動するまで待機 問合せを実行してIn-Memory表を強制的にロードし、表のロードが完了するまで待機 影響を受けたインスタンスでサービスを再有効化 再起動されたインスタンスに徐々にロードが戻ることを観察 	<ul style="list-style-type: none"> 次の理由により、障害発生時に一時的に応答時間のスパイクが発生： <ul style="list-style-type: none"> 問合せの中断 Oracle RACの再構成 アプリケーション・コンティニューイティが使用される際の問合せの再実行/再現 障害を起こしたインスタンスがクラスタに再結合するとき一時的に応答時間のスパイクが発生

データベース・ノード障害	<ol style="list-style-type: none"> ORA-12805エラーまたはORA-40エラーの後にこれらを捕捉して再試行できることを確認 ワークロードを起動 データベース・ノードの電源を停止 影響を受けたインスタンスのサービスをすみやかに無効化 電源を投入し、インスタンスが再起動するまで待機 問合せを実行してIn-Memory表を強制的にロードし、表のロードが完了するまで待機 影響を受けたインスタンスでサービスを再有効化 再起動されたインスタンスに徐々にロードが戻ることを観察 	<ul style="list-style-type: none"> 次の理由により、障害発生時に一時的に応答時間のスパイクが発生： <ul style="list-style-type: none"> 問合せの中断 Oracle RACのMISCOUNTIによる遅延（通常1分）。この待機時間は、12.1.0.2 BP7以上のエンジニアード・システムでは数秒に短縮されています。 Oracle RACの再構成 アプリケーション・コンティニューイティが使用される際の問合せの再実行/再現 障害を起こしたインスタンスがクラスタに再結合するとき一時的に応答時間のスパイクが発生
データベース障害時のData Guardフェイルオーバー	<ol style="list-style-type: none"> ワークロードを起動 スタンバイ・データベースのサービスが無効化されている（かつ、プライマリでは有効化されている）ことを確認 クラスタ内のすべてのノードをABORTモードで停止 スタンバイ・データベースがオープンされるまで待機 問合せを実行してIn-Memory表を強制的にロードし、表のロードが完了するまで待機 新しいプライマリ・データベースでサービスを有効化 新しいプライマリ・データベースでロードが発生し始める様子を観察 	<ul style="list-style-type: none"> 次の理由により、障害発生時に応答時間のスパイクが発生： <ul style="list-style-type: none"> 問合せの中断 スタンバイのオープン キャッシュ内のIn-Memory表のロード アプリケーション・コンティニューイティが使用される際の問合せの再実行/再現 <p><i>注：データベースがプライマリとして起動した直後にクライアントやアプリケーションを接続できるようにすると、データベースおよびアプリケーションのリカバリ時間目標（RTO）は非常に短くなります。In-Memory表があらかじめロードされるまで待機してからアプリケーションが接続できるようにすると、停止時間またはRTOは長くなります。企業および管理者は、RTOを短くするか長くするか、両者の長所と短所をじっくり検討する必要があります。後者を選択すると、新しいプライマリ・データベースに接続した直後からインメモリ問合せの応答時間が短くなります。</i></p>

上記のテストに関しては、以下の点に注意してください。

- » テスト・アプリケーションは、12.1.0.2 の UCP 接続プールを使用するように構成されていました。
- » 残りのノードには、保守作業のために停止されたインスタンスのロードを引き継げるだけの十分な CPU 能力とメモリ容量がありました。使用している環境がこれとは異なる場合は、オラクルのリソース管理機能を活用し、パフォーマンス SLA への影響を容認する必要があります。

結論

Oracle Database In-Memory は、Oracle Database の信頼性と高可用性を維持しながら、パフォーマンスを劇的に向上させます。このホワイト・ペーパーで紹介したベスト・プラクティスを実装することで、Oracle Database In-Memory を実装したときの可用性を最大化することができます。



参考資料

Oracle Technology Network : Oracle Database In-Memory のページ

<http://www.oracle.com/jp/products/database/options/database-in-memory/overview/index.html>

Oracle Database In-Memory ブログ、<https://blogs.oracle.com/In-Memory/tags/rac>

FAN イベントに基づいてサービスを自動的に無効化するシェル・スクリプト、
<http://support.oracle.com>、MOS note ID 1927000.1

アプリケーションの中断を伴わない、Oracle RAC での正常なアプリケーション・スイッチオーバー、
<http://support.oracle.com>、MOS note ID 1593712.1

クライアントおよびアプリケーションのフェイルオーバーの検証マトリックス、
<http://support.oracle.com>、MOS note ID 1617163.1

『高可用性 Oracle Database でのクライアント・フェイルオーバーのベスト・プラクティス - Oracle Database 12c』、

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



CONNECT WITH US



blogs.oracle.com/oracle



facebook.com/oracle



twitter.com/oracle



oracle.com

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

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

Hardware and Software, Engineered to Work Together

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

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

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。0615

Oracle Database In-Memory：高可用性ベスト・プラクティス
2015年9月
著者：Hector Pujol



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