

Oracle Database Technology Night

～ 集え！オラクルの力(チカラ) ～

18c ORACLE[®]
Database

Oracle Database 18c テクノロジーシリーズ 2

「RAC/Sharding と Data Guard/HA の機能強化」

～ Data Guard & HA ～

日本オラクル株式会社
ソリューション・エンジニアリング統括
クラウド・インフラストラクチャ一本部
柴田 長



- 以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料にならないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。
文中の社名、商品名等は各社の商標または登録商標である場合があります。

アジェンダ

- 1 ➤ マルチインスタンスREDO適用時と
ブロックチェンジトラッキングの併用
- 2 ➤ NOLOGGING処理後の
スタンバイ・データベースブロックの修復
- 3 ➤ Shadow Lost Write Protect
- 4 ➤ その他 Data Guard関連新機能

アジェンダ

- 1 ➤ マルチインスタンスREDO適用時と
ブロックチェンジトラッキングの併用
- 2 ➤ NOLOGGING処理後の
スタンバイ・データベースブロックの修復
- 3 ➤ Shadow Lost Write Protect
- 4 ➤ その他 Data Guard関連新機能

マルチインスタンスREDO適用時とブロックチェンジトラッキングの併用

必要なエディション : EE, EE-ES, DBCS-EP, ExaCS

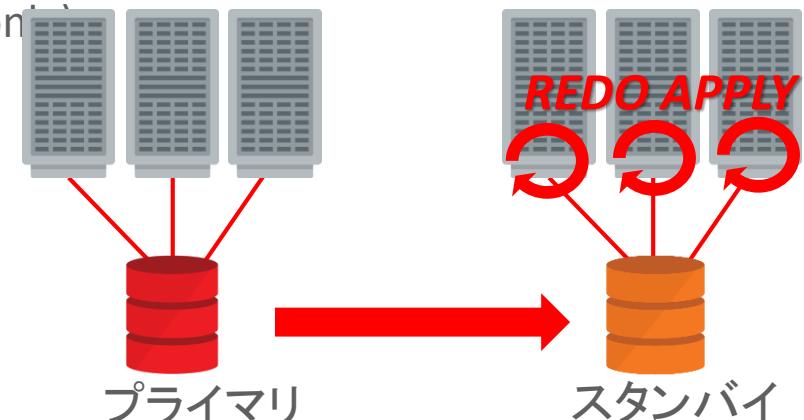
(12.2～)複数インスタンスでのパラレルREDO適用

REDO適用性能の向上

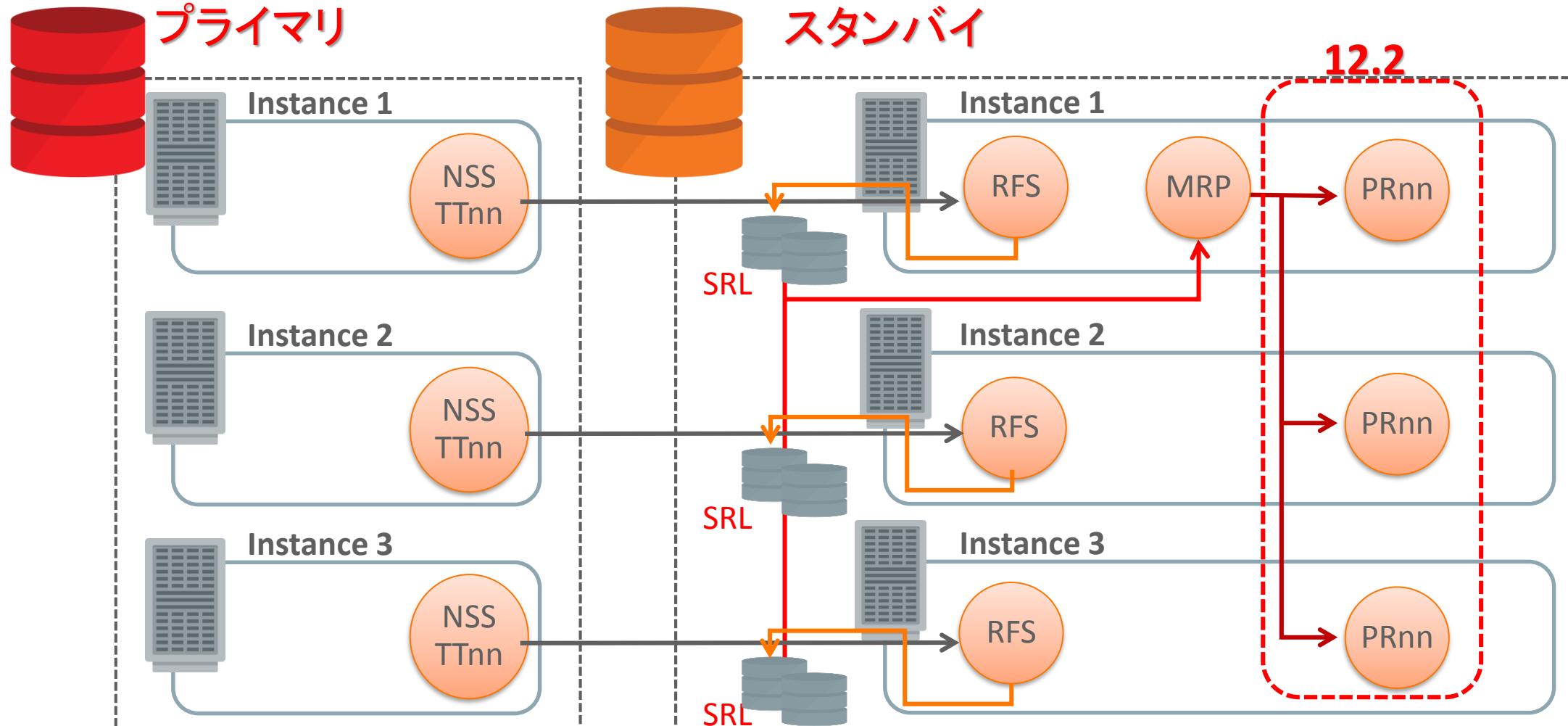
- RACの複数インスタンスでパラレルにREDO適用を行うことが可能となった
 - 高いREDO適用性能を必要とするようなREDO生成量の多い大規模環境で効果が期待できる
 - 検証結果: 8ノードRACで、3,500MB/sec 適用性能

```
SQL> RECOVER MANAGED STANDBY DATABASE DISCONNECT USING INSTANCES [ALL | <N>];
```

- N : 1～インスタンス数まで指定可能、1スレッド/インスタンス
- ALL の場合は、全インスタンスで同じモードの必要がある(mount/read-only)
- Data Guard Broker プロパティ値 : **ApplyInstances**
- ただし、下記を使用しているスタンバイ環境では利用不可
 - ブロックチェンジトラッキング(高速増分バックアップ)
 - Database In-Memory 機能
 - パラレルREDO 適用を使用する場合、ADG_IMC_ENABLED 値が全インスタンスで同じである必要あり



(12.2～)複数インスタンスでのパラレルREDO適用



(18.1～)複数インスタンスでのパラレルREDO適用

マルチインスタンスREDO適用時とロックチェンジキャッシングの併用

- これまで併用出来なかった下記2つの機能を併用できるようになった
 - スタンバイデータベースにおける複数インスタンスでのパラレルREDO適用
 - スタンバイデータベースにおけるチェンジキャッシングファイル有効化(高速増分バックアップ)

```
SQL> RECOVER MANAGED STANDBY DATABASE DISCONNECT USING INSTANCES ALL;
```

```
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '+FRA(CHANGETRACKING)' REUSE ;
```

```
SQL> SELECT * FROM V$BLOCK_CHANGE_TRACKING ;
```

STATUS	FILENAME	BYTES
ENABLED	+FRA/orcl/changetracking/ctf.282.827005385	11599872

アジェンダ

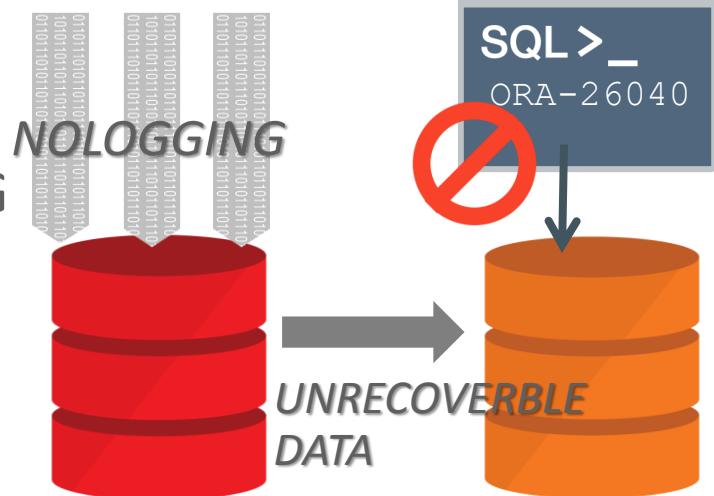
- 1 ➤ マルチインスタンスREDO適用時と
ブロックチェンジトラッキングの併用
- 2 ➤ NOLOGGING処理後の
スタンバイ・データベースブロックの修復
- 3 ➤ Shadow Lost Write Protect
- 4 ➤ その他 Data Guard関連新機能

NOLOGGING処理後のスタンバイ・データベース ブロックの修復

必要なエディション : EE-ES, DBCS-EE+, ExaCS (OnPのEEでは使えません)

(12.1以前) NOLOGGING 処理に対するリカバリが必要 スタンバイ側でのアクセス時にエラーが返る

- I/Oネックであるような環境ではデータロード時などに REDO生成を抑えて実行時間を短縮するためにNOLOGGINGを使う場合があった
- Data Guard環境においてはプライマリでのNOLOGGINGの処理はスタンバイのデータ破損を起こす
- 結果、スタンバイ側で該当データへのアクセス時にエラー返る



```
SQL> select count(*) from tab1;
select count(*) from tab1
*
```

行1でエラーが発生しました。:

```
ORA-01578: Oracleデータ・ブロックに障害が発生しました(ファイル番号15、ブロック番号131)
ORA-01110: データファイル15: '+DATA/xxx/xxx/DATAFILE/ts1.347.922632353'
ORA-26040: データ・ブロックがNOLOGGINGオプションを使用してロードされました。
```

(12.1以前) NOLOGGING 処理に対するリカバリが必要 手動でのリカバリ

- 対処方法としては下記のいずれか
 - 中間表など一時的なデータなので無視する(のちにDROPされる)
 - 影響を受けるテーブルの再作成
 - NOLOGGING処理開始直前からの増分バックアップをプライマリで取得し、スタンバイに増分バックアップを適用する
 - スタンバイデータベースを再作成する

(12.1以前) NOLOGGING 処理への対処

影響範囲の特定/対処に時間要する



(12.2から) NOLOGGING 処理に対する検証やりカバリ

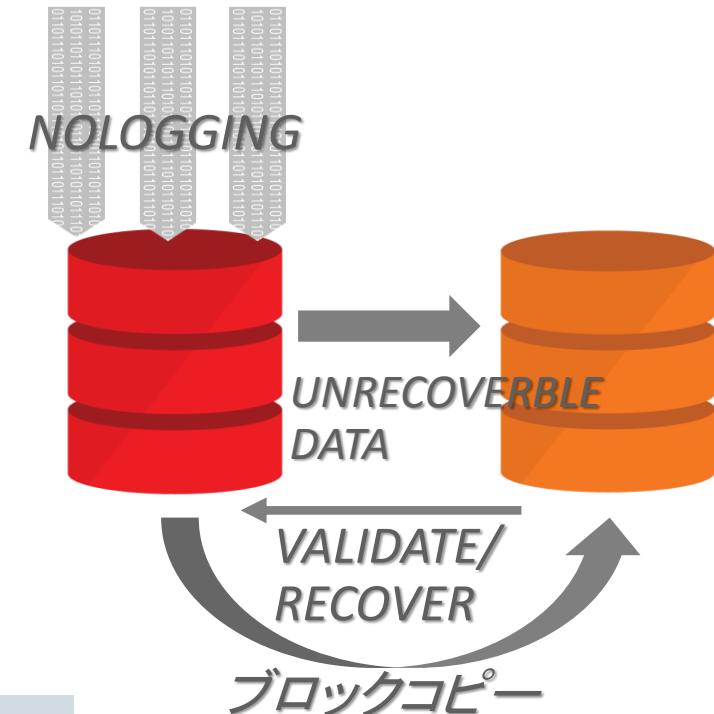
DWH向けの機能強化

- スタンバイで NOLOGGING で更新されたブロックに対する **VALIDATE/RECOVER の実行が可能**
- RMAN による検証とリカバリで、修復が簡単に
 - アクセスしないと気付けなかったNONLOGGED BLOCK に気付け、1コマンドでリカバリが可能に
 - データベース全体 or データファイルごとに実行
 - 実行時は、管理リカバリモード(MRP)の停止が必要
 - 検証

```
RMAN> VALIDATE DATABASE/DATAFILE ... NONLOGGED BLOCK;
```

- リカバリ

```
RMAN> RECOVER DATABASE/DATAFILE ... NONLOGGED BLOCK;
```



(12.2から) NOLOGGING 処理に対する検証やりカバリ

参考) 1. スタンバイ側でNOLOGGING 処理のブロックを確認

- 管理リカバリモード(MRP)の停止

```
SQL> RECOVER MANAGED STANDBY DATABASE CANCEL;
```

- 下記のいずれかの方法で検証

- 特定のデータファイルを検証

```
RMAN> VALIDATE DATAFILE <データファイルID> NONLOGGED BLOCK;
```

- データベース全体を検証

```
RMAN> VALIDATE DATABASE NONLOGGED BLOCK;
```

- V\$NONLOGGED_BLOCK を確認

```
SQL> select FILE#,BLOCK#,BLOCKS,NONLOGGED_START_CHANGE#,NONLOGGED_END_CHANGE#
      from V$NONLOGGED_BLOCK;
```

FILE#	BLOCK#	BLOCKS	NONLOGGED_START_CHANGE#	NONLOGGED_END_CHANGE#
-----	-----	-----	-----	-----
15	131	10	2701076	2701105

(12.2から) NOLOGGING 処理に対する検証やりカバリ

参考) 2. スタンバイ側でNONLOGGED のブロックをリカバリ

- 下記のいずれかの方法でリカバリ

- 特定のデータファイルをリカバリ

```
RMAN> RECOVER DATAFILE <データファイルID> NONLOGGED BLOCK;
```

- データベース全体をリカバリ

```
RMAN> RECOVER DATABASE NONLOGGED BLOCK;
```

- V\$NONLOGGED_BLOCK で該当ブロックを確認

```
SQL> select FILE#,BLOCK#,BLOCKS,NONLOGGED_START_CHANGE#,NONLOGGED_END_CHANGE#
      from V$NONLOGGED_BLOCK;
```

レコードが選択されませんでした。

- 管理リカバリモードを開始

```
SQL> RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

(12.2から) NOLOGGING 処理への対処

影響範囲の特定/対処が自動化可能になった



(12.2から) NOLOGGING 処理に対する検証やりカバリ 実行時のログ

- リカバリ時動作

```
RMAN> recover datafile 7 nonlogged block;

recoverを18-03-29で開始しています
リカバリ・カタログのかわりにターゲット・データベース制御ファイルを使用しています
チャネル: ORA_DISK_1が割り当てられました
チャネルORA_DISK_1: SID=17 デバイス・タイプ=DISK

記録されていないブロックのリカバリを開始しています
データファイル・リスト
=====
ファイル ステータス 記録されていないブロック 調査済ブロック スキップされたブロック
-----
7      OK        0            139            660

記録されていないブロックの詳細は、v$nonlogged_blockビューから問合せできます
記録されていないブロックのリカバリが完了しました、経過時間: 00:00:00

recoverを18-03-29で終了しました
```

(12.2から) NOLOGGING 処理に対する検証やりカバリ 実行時のログ

- リカバリ時のアラートログの出力@スタンバイ

```
2018-03-29T11:04:19.320035+09:00
Started Nonlogged Block Replacement recovery on file 7 (ospid 26047 rcvid 15114019826015144720)
2018-03-29T11:04:19.441332+09:00
Data Transfer Cache defaulting to 96MB. Trying to get it from Buffer Cache for process 26049.
2018-03-29T11:04:19.530114+09:00
Finished Nonlogged Block Replacement recovery on file 7. 0 blocks remain
  Statistics for replacement block source database (service=prim)
    Blocks requested 139, blocks received 139.

  Reason replacement blocks accepted or rejected           Blocks Last block
  -----  
Accept: SCN in range for classic non-logged block      130      525
```

- リカバリ時のアラートログの出力@プライマリ

```
2018-03-29T11:04:19.131810+09:00
TT06: Start data block transfer (PID:21831)
2018-03-29T11:14:20.289630+09:00
TT06: Data block transfer ends
```

(12.2から) NOLOGGING 処理に対する検証やりカバリ 制限事項

- プライマリ側でFlashback Databaseや、OPEN RESETLOGSを実行している場合、スタンバイ側で必要なブロックが手に入らず下記のようなエラーがアラートログに出力される

```
ORA-12942: database incarnation at source does not match
```

(18.1から) NOLOGGING 処理に対する自動リカバリ

- Data Guard構成で管理リカバリモードが有効な場合、**自動的に破損ブロックをプライマリから取得して置き換える機能が動作する(MOUNTでも動作した)**
- 前提事項
 - NOLOGGINGオペレーション実施前にデータベースを適切な LOGGING モード(STANDBY NOLOGGING FOR LOAD PERFORMANCE or STANDBY NOLOGGING FOR DATA AVAILABILITY) に変更する

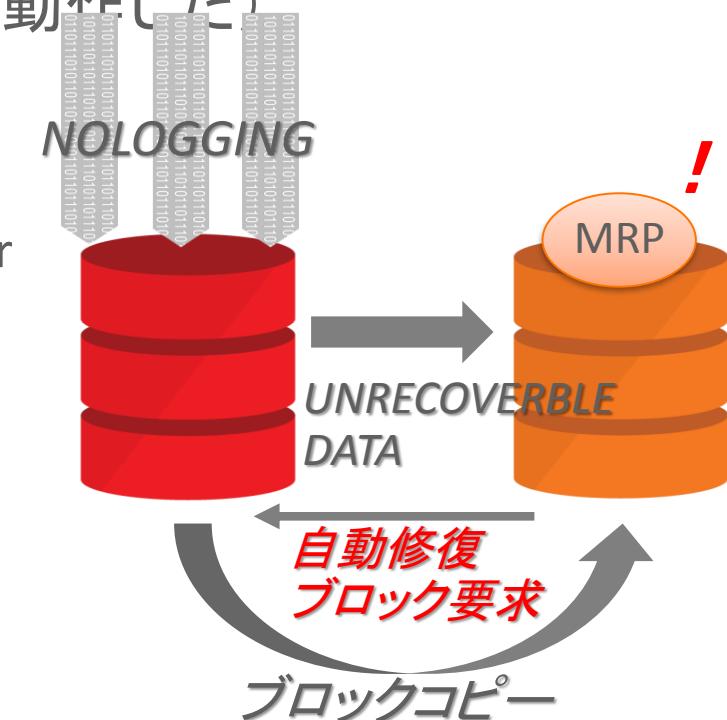
```
SQL> ALTER DATABASE SET STANDBY NOLOGGING FOR LOAD
PERFORMANCE;
```

データベースが変更されました。

```
SQL> select FORCE_LOGGING from v$database;
```

```
FORCE_LOGGING
```

```
STANDBY NOLOGGING FOR LOAD PERFORMANCE
```



(18.1から) NOLOGGING 処理に対する自動リカバリ

Data Guard 構成に利用できるLOGGINGモード一覧

LOGGING モード	説明
FORCE LOGGING	全てのロード処理がNOLOGGINGにならないようにするためのモード
SET STANDBY NOLOGGING FOR DATA AVAILABILITY 18c ~	プライマリとスタンバイ間のネットワークが確立されている場合は必要データをネットワーク越しに送り、ネットワークが切れている場合はLOGGINGモードになりREDOログに更新後データを残す(その後スタンバイに送られて反映される)
SET STANDBY NOLOGGING FOR LOAD PERFORMANCE 18c ~	スタンバイでREDOを適用する際、NOLOGGING処理が実行されたためにブロックが破損してしまう場合には、スタンバイが自動的にプライマリにデータを要求する

(18.1から) NOLOGGING 処理に対する自動リカバリ

STANDBY NOLOGGING FOR LOAD PERFORMANCEモード

- 管理リカバリプロセスがREDO適用時にNOLOGGING処理を検知すると、プライマリに対して該当ブロックを要求する
- スタンバイのアラートログには次のような出力がされる

```
2017-12-20T16:58:08.222296+09:00
Data Transfer Cache defaulting to 96MB. Trying to get it from Buffer Cache for
process 19717.
```

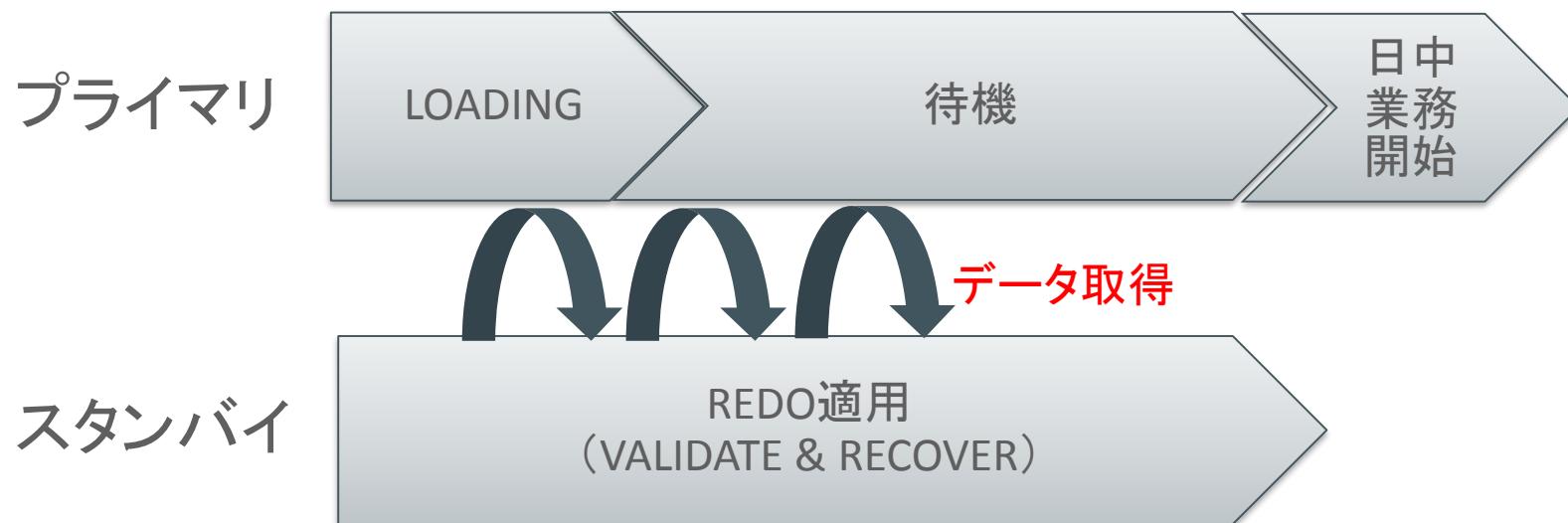
- スタンバイ側でデータを受信するプロセスの実体としてはサーバープロセス(名前はDTS:Data Transfer Service)

```
SQL> SELECT PROCESS, PID FROM V$MANAGED_STANDBY WHERE PID=19717;
```

PROCESS	PID
-----	-----
DTS	19717

(18.1から) NOLOGGING 処理への対処

影響範囲の特定と対処を効率化(LOAD PERFORMANCEモード)



(18.1から) NOLOGGING 処理に対する自動リカバリ

STANDBY NOLOGGING FOR DATA AVAILABILITYモード

- プライマリとスタンバイのネットワーク接続が正常な場合には、LOAD PERFORMANCE モードと同様の動作
- ネットワーク接続が切れている場合は、プライマリデータベースが検知をし強制的に LOGGING 処理に切り替わる

```
2018-03-30T12:35:28.215173+09:00
.... (PID:31966): Error 12541 received logging on to the standby
.... (PID:31966): Check whether the listener is up and running.
WARNING! A session under STANDBY NOLOGGING for DATA AVAILABILITY mode failed to
complete all necessary setup. The operation will continue but without sending
data blocks to any standbys. Instead block image redo will be logged. Check
network connectivity to the standbys.
```

(18.1から) NOLOGGING 処理への対処

影響範囲の特定と対処を効率化(DATA AVAILABILITYモード)



アジェンダ

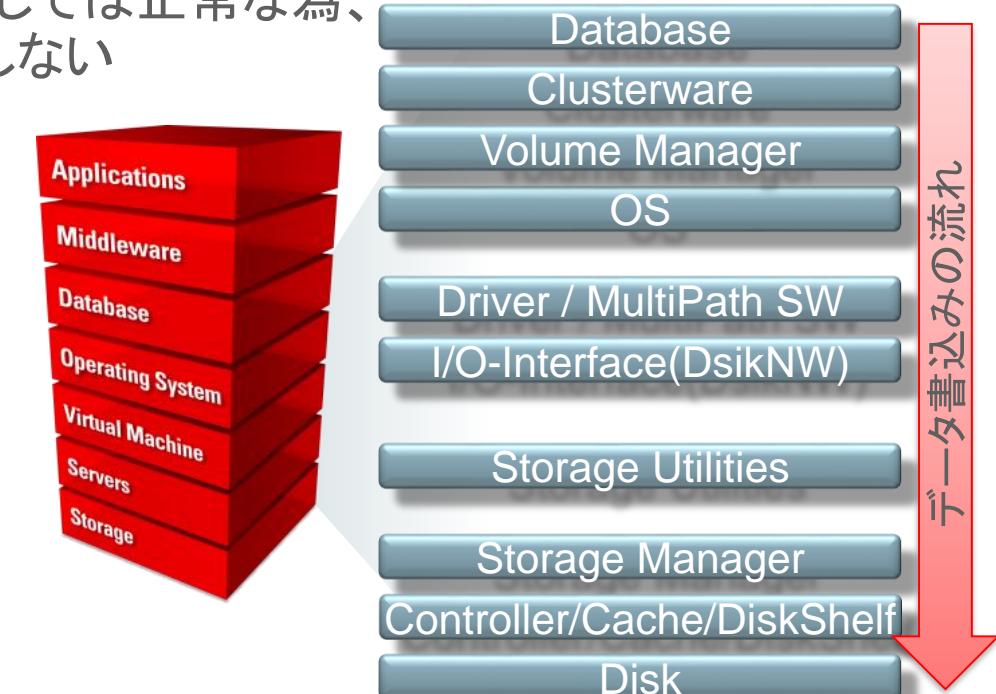
- 1 マルチインスタンスREDO適用時と
ブロックチェンジトラッキングの併用
- 2 NOLOGGING処理後の
スタンバイ・データベースブロックの修復
- 3 Shadow Lost Write Protect
- 4 その他 Data Guard関連新機能

Shadow Lost Write Protect

必要なエディション : EE, EE-ES, DBCS-EE+, ExaCS

Lost Writeとは?

- 例えば、ストレージ装置からBlockの書き込み完了が通知されたにも関わらず、実際にはDiskに書き込まれていない事象
 - ディスク上には更新前のData Blockが存在、Blockの構造としては正常な為、Lost Write が発生したBlockにアクセスしてもエラーは発生しない
 - 不正なデータをユーザー/顧客に提供するリスク
 - 不正なデータ汚染が広がるリスク
 - Lost Writeによる影響の例
 - 在庫が無いにも関わらず、在庫有りとして注文を受けてしまう
 - 注文を受けたはずなのに、注文を受けていないことになってしまう



(12.2まで) Lost Write Protect

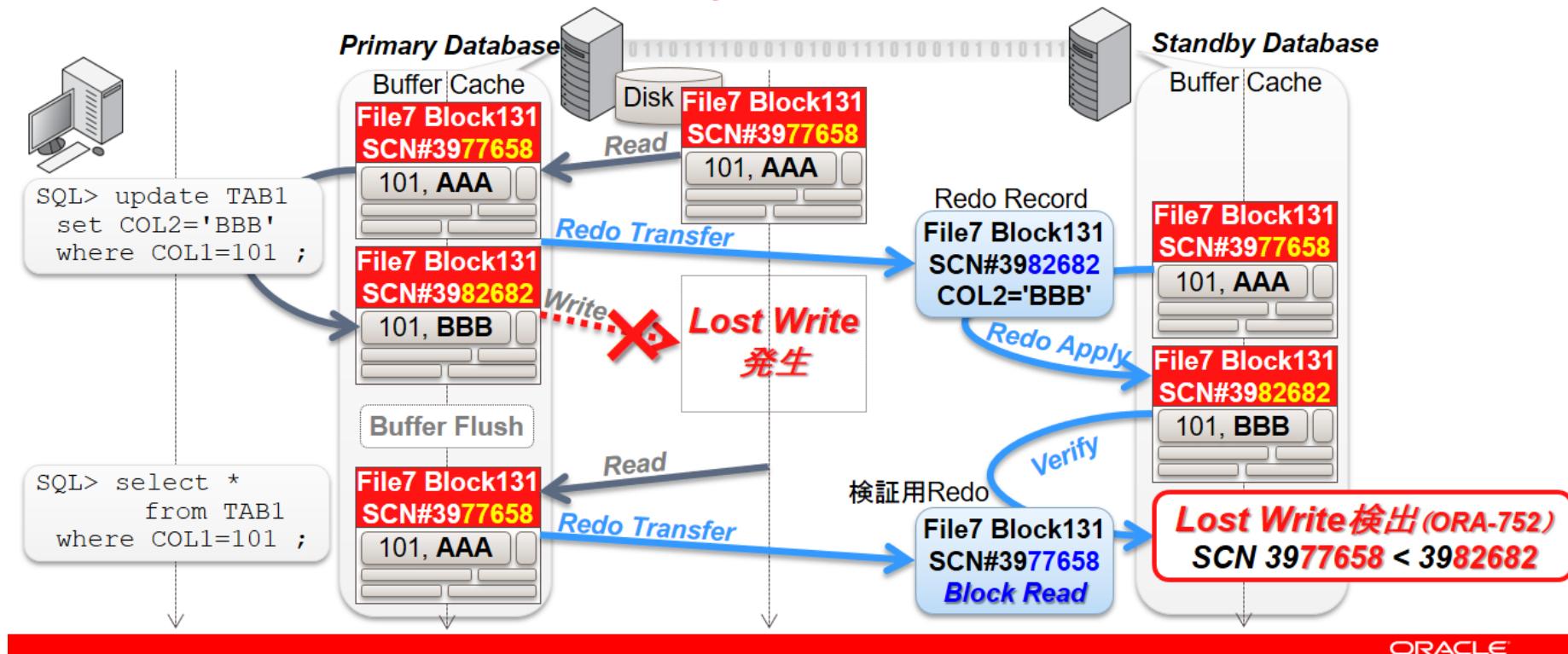
DB_LOST_WRITE_PROTECTパラメータの有効化

- Data Guard (Physical Standby Database) で Lost Write を検出する仕組みを提供
 1. Primary Database で Block を Disk から読み出す際に、検証用 Redo を生成
 - *Data File Number*
 - *Data Block Address (DBA)*
 - *System Change Number (SCN)*
 2. Data Guard の仕組みで Physical Standby Database へ Redo を転送
 3. Standby Database 側の対象 Block と Redo 内の SCN を比較検証

→ もし、SCN が不一致の場合は、Lost Write が発生していると判定可能

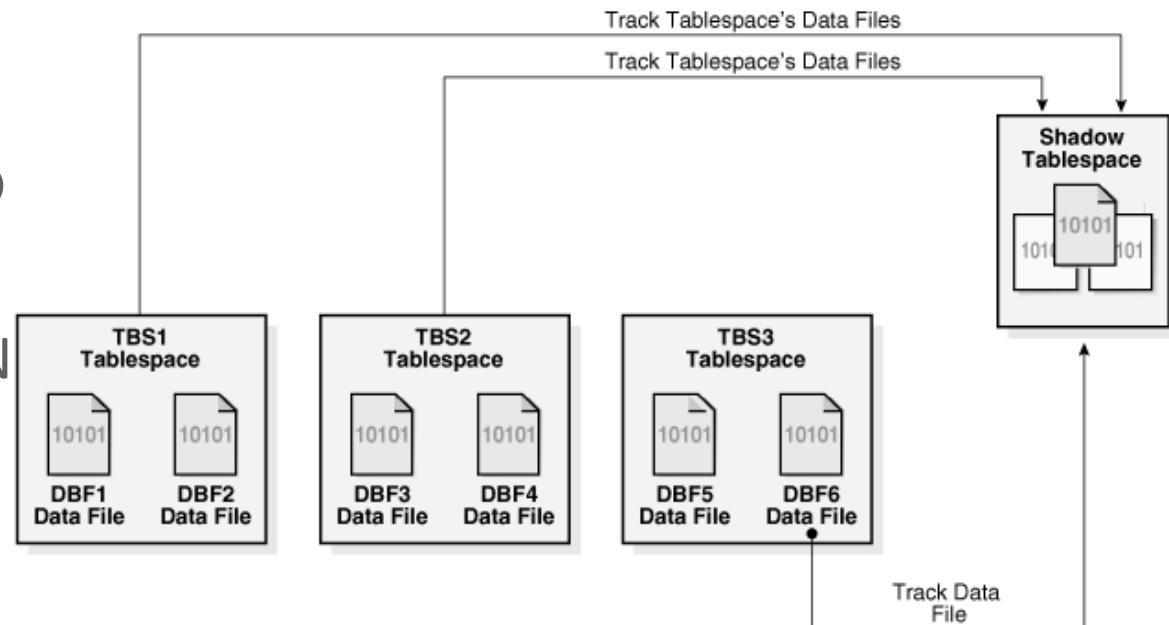
DB_LOST_WRITE_PROTECT

Lost Write 検出の流れ (Primary側でLost Writeしたケース)



Shadow Lost Write Protection

- Data Guard 構成無し(フィジカルスタンバイデータベース無し)でデータベースや表領域、データファイル単位でのロストライト検知を有効化できる機能
- Data Guard 構成が無い環境においてもデータ破損によるデータロストを最小化できる
- 本機能を有効化すると対象表領域上のバックファ上に任意のデータがディスクに書き出されるタイミングでそのブロックのSCNを記録する
- ディスクからの読み込み時に実際のSCNと上記SCNを比較してロストライトが発生していないかどうかを判断する(実際SCN < 上記SCNなら発生)



Shadow Lost Write Protection

有効化手順

- Shadow Lost Write Protect検知用に下記のように表領域を作成する

```
SQL> create bigfile tablespace SHADOW_TBS datafile  
'/u01/app/oracle/oradata/PRIM/shadow_tbs_01.dbf' size 10M lost write protection;
```

表領域が作成されました。

- 正しく作成されたかどうかを確認します

```
SQL> select CONTENTS from DBA_TABLESPACES where TABLESPACE_NAME='SHADOW_TBS';  
  
CONTENTS  
-----  
LOST WRITE PROTECTION
```

Shadow Lost Write Protection

有効化手順

- データベース全体でShadow Lost Write Protectionを有効化します

```
SQL> alter database enable lost write protection;
```

データベースが変更されました。

- 対象とする表領域/データファイルに対してShadow Lost Write Protectionを有効化します

```
SQL> alter tablespace users enable lost write protection;
```

表領域が変更されました。

```
SQL> select LOST_WRITE_PROTECT from DBA_TABLESPACES where  
TABLESPACE_NAME='USERS';
```

LOST_WR

ENABLED

Shadow Lost Write Protection

- ロストライトを検知すると下記のエラーが出力されます

```
SQL> select * from lwp1;
select * from lwp1
*
```

行1でエラーが発生しました。:

ORA-65478: シャドウ消失書込み保護 - 消失書込みが見つかりました

アラートログの内容

2018-03-30T16:46:33.696232+09:00

ERROR - I/O type:buffered I/O found lost write in block with file#:7 rdba:0x1c00f84,
Expected SCN:0x00000000001e078a SCN in block:0x00000000001dfe88, approx current
SCN:0x00000000001e0aa4, RAC instance:1 pdb:1

An internal routine has requested a dump of selected redo.

This usually happens following a specific internal error, when
analysis of the redo logs will help Oracle Support with the
diagnosis.

It is recommended that you retain all the redo logs generated (by
all the instances) during the past 12 hours, in case additional
redo dumps are required to help with the diagnosis.

Shadow Lost Write Protection

その他補足事項

- ダイレクトパスロードでも有効に動作します
- RMANバックアップによるブロックアクセスでも動作します
- 少なくとも2%の領域を余分に使います
- DB_LOST_WRITE_PROTECT パラメータの設定有無に関わらず動作します

アジェンダ

- 1 ➤ マルチインスタンスREDO適用時と
ブロックチェンジトラッキングの併用
- 2 ➤ NOLOGGING処理後の
スタンバイ・データベースブロックの修復
- 3 ➤ Shadow Lost Write Protect
- 4 ➤ その他 Data Guard関連新機能

その他 Data Guard関連新機能

v\$dataguard_process

Data Guard に関するプロセスの状態を確認できる

- v\$managed_standby の置き換え

```
SQL> select NAME, PID, ROLE, ACTION from
  v$dataguard_process;
```

NAME	PID	ROLE	ACTION
LGWR	31581	log writer	IDLE
TMON	31613	redo transport monitor	IDLE
TT00	31625	gap manager	IDLE
TT01	31627	redo transport timer	IDLE
ARC0	31623	archive local	IDLE
ARC1	31629	archive redo	IDLE
ARC2	31631	archive redo	IDLE
ARC3	31633	archive redo	IDLE
TT02	31635	async ORL multi	WRITING
TT03	31637	heartbeat redo informer	IDLE

```
SQL> select NAME, PID, ROLE, ACTION from
  v$dataguard_process;
```

NAME	PID	ROLE	ACTION
LGWR	17077	log writer	IDLE
TMON	17109	redo transport monitor	IDLE
TT00	17118	gap manager	IDLE
TT01	17122	redo transport timer	IDLE
ARC0	17120	archive local	IDLE
ARC1	17124	archive redo	IDLE
ARC2	17126	archive redo	IDLE
ARC3	17128	archive redo	IDLE
MRP0	17165	managed recovery	IDLE
PR00	17167	recovery logmerger	APPLYING_LOG
PR01	17169	recovery apply slave	IDLE
PR02	17171	recovery apply slave	IDLE
PR03	17173	recovery apply slave	IDLE
PR04	17175	recovery apply slave	IDLE
rfs	17207	RFS async	WRITING
rfs	17209	RFS archive	IDLE
rfs	17211	RFS ping	IDLE

ネットワーク越しでのスタンバイDBのリフレッシュ

- プライマリで増分バックアップを取得し、その増分バックアップでスタンバイデータベースをリフレッシュ可能です
- スタンバイ側でリカバリに必要なアーカイブREDOログをプライマリ、スタンバイ双方で消してしまった場合の復旧に便利
- スタンバイDBを再起動、マウント状態にしてプライマリからのバックアップをリストアします

```
RMAN> recover standby database from service prim ;
```

ネットワーク越しでのスタンバイDBのリフレッシュ

- 次のような動作が自動実行される
 1. スタンバイを停止
 2. スタンバイをNOMOUNT
 3. プライマリでスタンバイ制御ファイル作成し、制御ファイルをリストア
 4. スタンバイをMOUNT
 5. `standby_file_management=manual`
 6. スタンバイの現行SCN以降の増分バックアップをプライマリで取得、スタンバイにリストア
 7. リカバリ
 8. `standby_file_management=auto`

テック・ナイトアーカイブ資料と お役立ち情報

各回テック・ナイトセッション資料 ダウンロードサイト



技術コラム しば ちょう先生の 試して納得！ DBAへの道



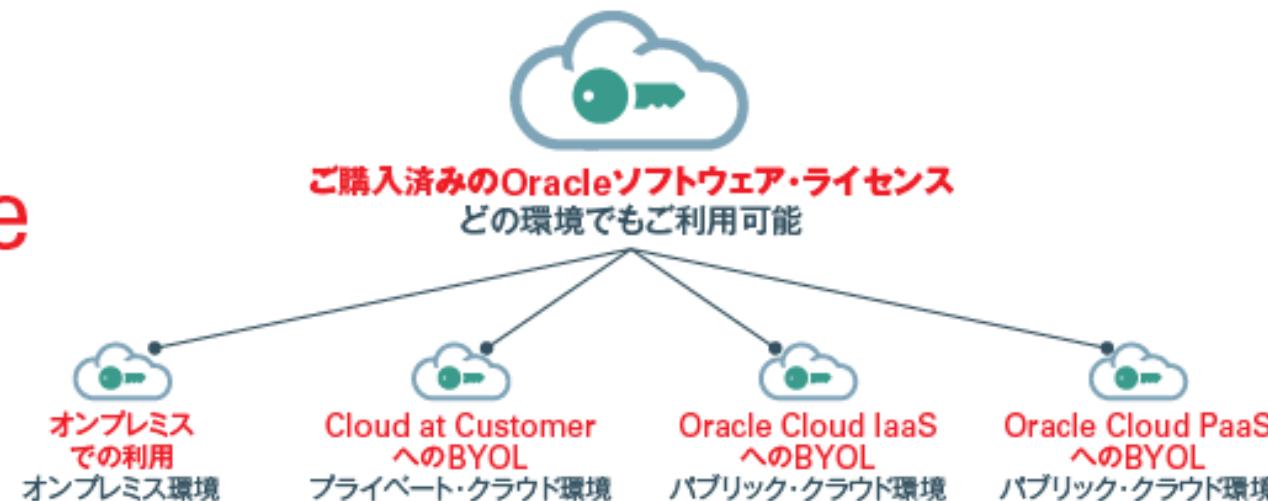
技術コラム 津島 博士の パフォーマンス 講座



もしも
みんなが
DBをクラウドで
動かしてみたら

Bring Your Own License

既存のオラクル・ライセンスを柔軟にクラウド環境で活用



300ドル分の無料トライアルでOracle Cloudを体験!



https://cloud.oracle.com/ja_JP/tryit

Oracle Cloudでは各種クラウドサービスを300ドル分無料でお試しいただけるトライアルサービスをご提供しております。無料トライアルのお申込み方法の詳細は、左のQRコード、またはURLにアクセスしてください。

Oracle Cloudのユースケース、導入事例、資料、価格などの詳細情報は、下記URLにアクセスしてください。

<http://www.oracle.com/jp/cloud/platform/overview/index.html>

～ みなさまの投稿をお待ちしております～



Twitter

#OracleTechNight

こんな時、かけこむ会社が増えています。



ビジネスプロセスを
改善したい!



今のシステムは
使いにくい!



システムコストを
下げたい!



パフォーマンスを
良くしたい!



経営分析を
したいのだが...



どんなソリューションが
あるの?



見積りはどれくらい
なんだろう?



楽に管理を
したい!

Oracle Digitalは、オラクル製品の導入をご検討いただく際の総合窓口。
電話とインターネットによるダイレクトなコミュニケーションで、どんなお問い合わせにもすばやく対応します。
もちろん、無償。どんなことでも、ご相談ください。



お問い合わせは電話またはWebフォーム

☎ 0120-155-096

受付時間 月～金 9:00-12:00 / 13:00-17:00
(祝日および年末年始休業日を除きます)

<http://www.oracle.com/jp/contact-us>

Hardware and Software Engineered to Work Together

ORACLE®