

ORACLE®

# 高可用性ベスト・プラクティスによる データ破壊対策 完全版

日本オラクル株式会社  
テクノロジー製品事業統括本部 基盤技術本部  
Principal Engineer  
柴田 長



 #odddtky

日本オラクル、今年最大の技術トレーニング・イベント

**Oracle DBA &  
Developer Day 2013**

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さらないで下さい。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。

# 自己紹介

"しばちょう"こと柴田長(しばた つかさ)と申します。

日本オラクル株式会社  
テクノロジー製品事業統括本部 基盤技術本部  
応用グループ  
Principal Engineer  
柴田 長



Oracle Technology Networkで毎月連載中  
「しばちょう先生の試して納得！DBAへの道」

<http://www.oracle.com/technetwork/jp/database/articles/shibacho/index.html>

# Program Agenda

- Oracle Maximum Availability Architecture
- Oracle DatabaseのData Protection
  - データ破損の検出レベルを高める機能
  - 修復オペレーション
- Conclusion

***Maximum  
Availability  
Architecture***



ORACLE

# Block Corruption が及ぼす影響

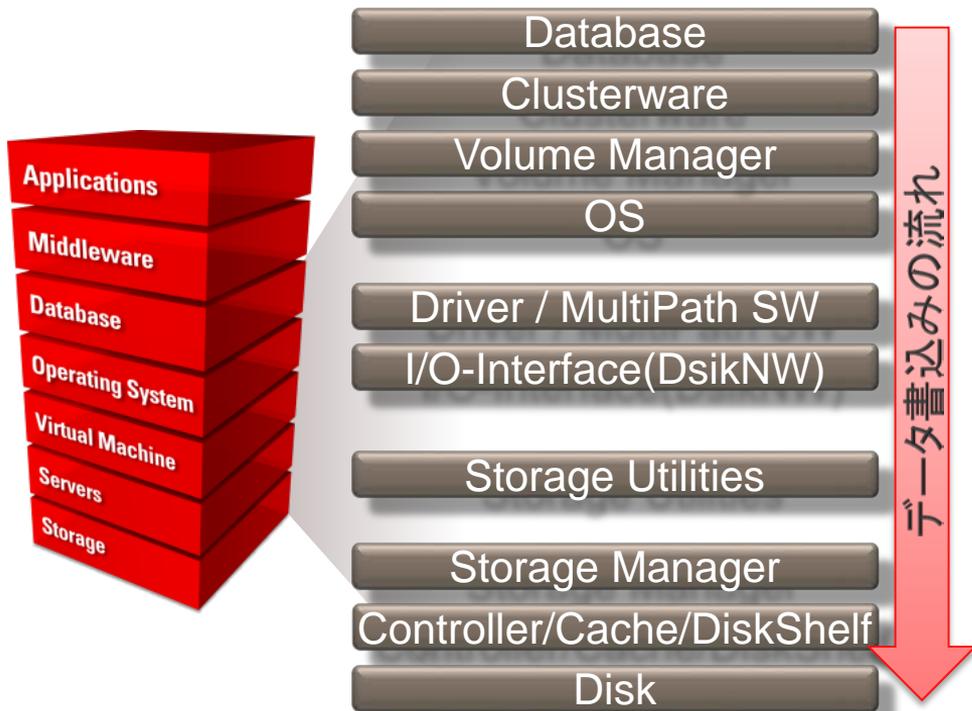
ビジネスへ直接的なダメージ



- ✓ データ損失による**業務停止**
- ✓ **長時間化**するデータ復旧
- ✓ 復旧作業の人的ミス、**二次災害**
- ✓ 原因究明の**長期化**

# データ破壊の主な要因

複雑かつ分断されたレイヤーのリスク



- ✓ 複数レイヤーを経由
- ✓ 各レイヤーがマルチベンダー提供で複雑化
- ✓ 各レイヤーにおける破損リスク
  - S/W、ファームウェアの不具合による破損
  - Diskフラッシュ時の不整合
  - 物理的部品故障、経年劣化にともなう破損

→ Database側に書き込み成功と戻ってきても、  
破損データが書き込まれていたり、そもそも何も書き込まれていない事象が発生するリスク

→ 修復時のオペレーション・ミスによる、  
業務の完全停止のリスク

# システムに求められるデータ保護の仕組み

業務の確実な継続性を高める為には？

- システム単体では防げないリスクがあることを考慮した保護の仕組み
  - 例: 人的ミスでデータを削除した場合、RAID構成では防げません
- 複製データ間での一貫性と確実性を考慮した保護の仕組み
  - 例: 部分的なバックアップによるデータ整合性の欠如
- 迅速な切り替えと確実なリカバリ・復旧(サービスの継続)を考慮した仕組み
  - 例: 被災訓練不足により、実際には切り替えられない、戻せないバックアップ

# Oracle Maximum Availability Architecture

## 高可用性テクノロジーのベスト・プラクティス

- **M**aximum **A**vailability **A**rchitecture (MAA) は、Oracleの実証済み高可用性テクノロジーと成功事例に基づいたベスト・プラクティス
- MAAの目的
  - あらゆる停止を回避、検出、修復するためのベスト・プラクティスを提供
  - 最適な高可用性アーキテクチャをシンプルに構成
    - ハードウェアやOSの影響を受けない(特定の高価な製品や技術は不要)
    - 高可用性のソリューションをすぐに提供(Oracleが事前検証済み)

# Oracle MAA の全体像

Low-Cost, Integrated, Fully Active, High ROI

## Production Site

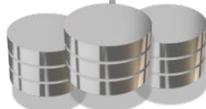
### RAC

- 拡張性
- サーバー障害対策



### Flashback

- 人的エラーの訂正



### ASM

- ボリュームマネージャー

### RMAN & Fast Recovery Area

- ディスクバックアップ

Edition-based Redefinition,  
Data Guard, GoldenGate

- メンテナンス、アップグレード、  
マイグレーションに伴う停止時間を極小化



## Active Replica

### Active Data Guard

- データ保護
- 災害対策
- クエリ・オフロード

### GoldenGate

- スタンバイの活用  
(Active-Active)
- 異機種混在環境への対応

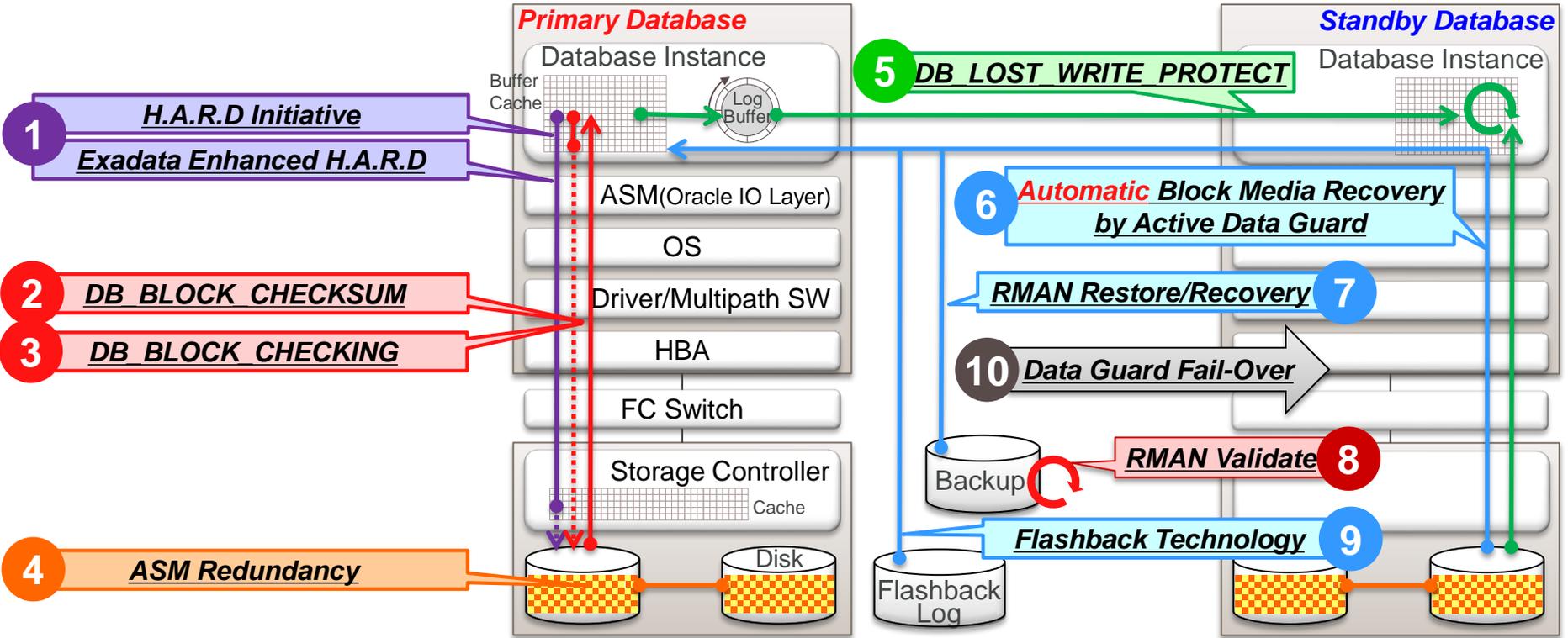


### Oracle Secure Backup

- テープバックアップ、クラウド対応

# Oracle MAA の Data Protection

Oracle Database 11g Release 2



# Oracle Database の Data Protection機能

● 検出レベルを高める機能



ORACLE

# Type of Block Corruption

*Doc ID 1088018.1 Master Note for Handling Oracle Database Corruption  
KROWN#152523 [マスターノート] Corruption(破損)*

- Data Block Corruption(Doc ID 840978.1)
  - *Physical Block Corruption*
  - *Logical Block Corruption*
- Other Corruptions
  - Control file Corruption
    - Use control file mirror & Copy
  - Redo Corruption
    - ASM Mirroring / Use multiplexed log file
  - Dictionary Corruption(Doc ID 136697.1)

*Oracle Support へ  
お問い合わせください*

# Data Block Corruption

## Physical Block Corruptions

- 物理的にBlock内のデータが破損している状態のBlock

- 破損例

- ① Block Headerが不正
- ② Block HeaderとFooterの情報が不一致
- ③ Dataの欠落
- ④ Blockの配置場所が不正
- ⑤ ゼロ埋めされたBlock



# Data Block Corruption

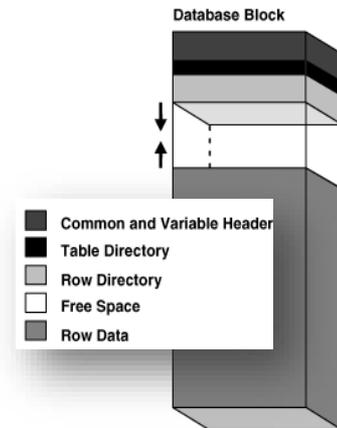
## *Logical Block Corruptions*

- Block内の構造が論理的に破損している(不正な)状態のBlock
  - 物理的(Block HeaderやFooterの情報、Checksumの計算結果)には正しい
  - 破損例
    - ① 行断片の開始位置や終了位置がブロック内におさまっていない
    - ② 行断片同士がオーバーラップしている
    - ③ 行断片をロックしているITLの番号が不正
    - ④ ITLが示すロック中の行断片の数が実際の数と一致しない
    - ⑤ Block内の空き領域のサイズが正しくない
    - ⑥ Lost Write

# Oracle Databaseに最適化された破損検知

## Data Block Format

- Oracle DatabaseのBlockは単なるビットの羅列ではなく、明確に事前定義された構造
  - Blockの構造を理解しているOracle Databaseだからこそ、Physical CorruptionやLogical Corruptionのチェックが可能
  - さらに、**Oracle Databaseの各Technologyと併用することで、検出レベルを高めることが可能**
- OS、ファイルシステムやストレージでは、命令通りにBlockをI/Oするだけであり、その中身(Blockの構造)がData Blockとして正しいか否かは判断できない。
  - Exadata Cell Storage Serverと、H.A.R.D Initiative はチェック可能



# Oracle DatabaseのData Protection

## 検出レベルを高める機能

- チェックの範囲、レベル、破損タイプを制御する初期化パラメータ
  - DB\_BLOCK\_CHECKSUM
  - DB\_BLOCK\_CHECKING
  - DB\_LOST\_WRITE\_PROTECT

} DB\_ULTRA\_SAFEパラメータでこれらを一括制御可能
- 定期的なチェック機能
  - Oracle Recovery Manager(CHECK LOGICAL句 / VALIDATE コマンド)
  - SQL> ANALYZE TABLE文 (VALIDATE STRUCTURE CASCADE オプション)

# DB\_BLOCK\_CHECKSUM

## 概要

- Data Block内のChecksumを利用したPhysical Corruption検知の仕組み
  - DiskへBlockを書き込む前
    - DBWR、およびDirect Loadを実行するサーバー・プロセスが、Checksum (Block内の全データを元に計算した数値)をBlock Headerに格納しておく。
  - DiskからBlock読み出した後
    - Blockを読み出したプロセスが、再計算したChecksumとBlock Header内に格納しておいたChecksumを比較検証する。

→ もし、Checksumが不一致の場合は、Block内のデータが変更されたことになる為、Physical Block Corruptionが発生したと判定可能

# DB\_BLOCK\_CHECKSUM

## 設定値毎の動作

Value	Default	Checksum生成の タイミング	Checksum検証の タイミング	対象表領域	Redo Block のChecksum
OFF		Disk書込み前	Disk読み込み後	SYSTEM	無し
TYPICAL	✓	Disk書込み前	Disk読み込み後	全て	有り
FULL		Disk書込み前 + Buffer上で更新後	Disk読み込み後 + Buffer上で更新前	全て	有り

FULLの場合は、UPDATEやDELETE等のDML文でBuffer Cache上のBlockを更新する前後でも生成と検証が実行される

Redo BlockのChecksumは Archiving / Instance Recovery / Roll forward時に検証される

# DB\_BLOCK\_CHECKSUM

*Buffer Cache上で更新されたData Blockに対して  
Checksumが埋め込まれるタイミング*

- 設定値による動作の違いに注意
  - DB\_BLOCK\_CHECKSUM=TYPICAL
    1. サーバー・プロセスによるBlock更新直後、Checksumはゼロ
    2. DBWRがDiskへ書き込む際に、Checksumを計算して埋め込む
      - 更新の度にChecksumを計算しないことで効率化
  - DB\_BLOCK\_CHECKSUM=FULL
    1. サーバー・プロセスによるBlock更新直後、Checksumを計算して埋め込む
    2. DBWRがDiskへ書き込む際に、Checksumを検証
      - メモリ上でのBlock Corruptionを検知することが可能

# DB\_BLOCK\_CHECKSUM

## Redo Blockに対するChecksumの生成と検証 (KROWN#155653)

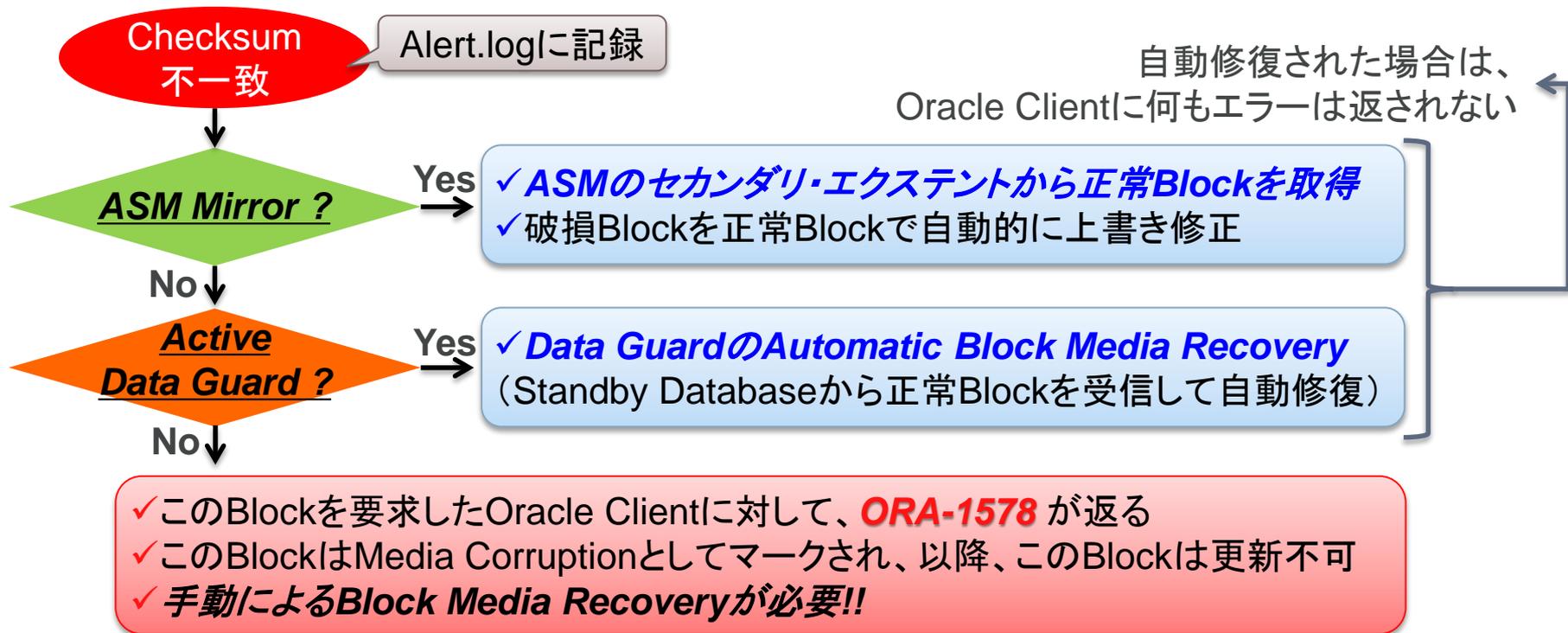
- リリース毎に挙動が異なる点に注意

Release	Checksum生成	Disk書込み前の検証
~10.2	LGWR	なし
11.1~11.2.0.1	Foreground Process	LGWR(FULLの場合のみ)
11.2.0.2~	Foreground Process	なし

- Release 11.1以降では、Redo BlockのChecksumの生成処理を、そのRedoを生成したForeground Processが担う → **LGWRの負荷が軽減**
  - ただし、Release 11.1~11.2.0.1において、FULLを設定している場合、LGWRがRedo BlockをDiskへ書込む前に、Foreground Processが生成したChecksumの整合性チェックを実行する動作になる

# DB\_BLOCK\_CHECKSUM

## 破損検出後の動作



# DB\_BLOCK\_CHECKSUM

## Automatic Block Media Recovery(ABR)で自動修復された際の参考ログ

### ■ Primary Database's Alert.log

```
Mon Aug 19 18:44:27 2013
Hex dump of (file 3, block 35) in trace file /u01/app/oracle/diag/rdbms/orcl/ORCL1/trace/ORCL1_ora_16202.trc
Corrupt block relative dba: 0x00000023 (file 3, block 35)
Fractured block found during buffer read
Data in bad block:
  type: 6 format: 2 rdba: 0x00000023
  last change scn: 0x0000.0cff8994 seq: 0x1 flg: 0x04
  spare1: 0x0 spare2: 0x0 spare3: 0x0
  consistency value in tail: 0x76150601
  check value in block header: 0xcf5
  computed block checksum: 0xffe3
Reading datafile '+DATA/orcl/datafile/tbs20.319.822054891' for corruption at rdba: 0x00000023 (file 3, block 35)
Read datafile mirror 'DATA_0002' (file 3, block 35) found same corrupt data (no logical check)
Read datafile mirror 'DATA_0000' (file 3, block 35) found same corrupt data (no logical check)
Automatic block media recovery requested for (file# 3, block# 35)
Mon Aug 19 18:44:29 2013
Automatic block media recovery successful for (file# 3, block# 35)
```

破損の検知

ASM Mirrorで自動修復を試行

ABRで自動修復成功

※ 上記からも確認出来るが、自動修復に成功した場合はORA-1578は記録されない

# DB\_BLOCK\_CHECKING

## 概要

- **Buffer Cache**上でBlockを変更後に、論理的な整合性をチェックすることで、Logical Corruptionを検知する
  - Checksumが正しい場合でも、論理的/意味的に不正な状態を検知可能
  - 変更後と表記した理由は、DMLによるデータ更新時以外の変更含む為
    - 例: DBWRの書き出しに伴う、Block Header情報の変更
  - Buffer Cacheを経由しないDirect Load Operationは、本チェック対象外
  - パラメータの設定値に応じて、チェックの対象やレベルを制御可能

# DB\_BLOCK\_CHECKING

## 設定値毎の動作

- 上位の設定値は下位の設定値のチェックを含む
  - 例えば、「LOW」=「OFF」のチェック + 全てのBlockのBlock Header Check

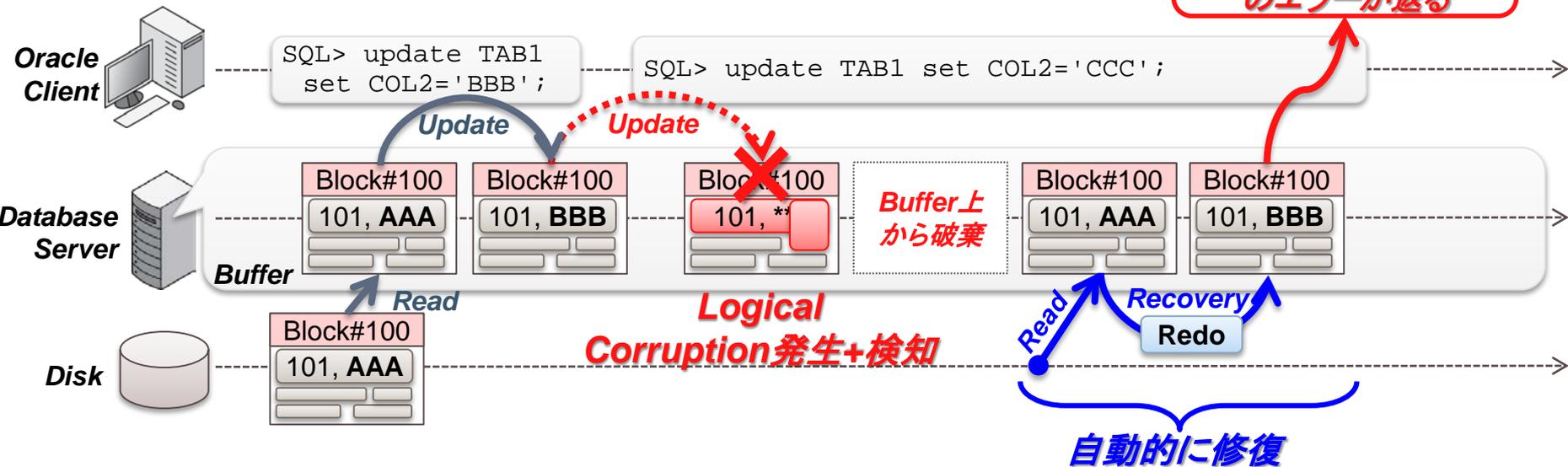
Value	Default	チェック方法	
		対象Block	種類
OFF	✓	SYSTEM表領域	Semantic Block Check
LOW		全て	Block Header Check
MEDIUM		全ての表セグメント	Semantic Block Check
FULL		全ての索引セグメント	Semantic Block Check

- 基本的には、Buffer Cache上でのBlock内容の変更後にチェックが実行されるが、Block Header Checkは、RACのインスタンス間でのBlock転送後にも実行される

# DB\_BLOCK\_CHECKING

破損検出後の動作 (Disk上のBlockが正常なケース)

Oracle Clientに  
ORA-607 & ORA-600  
のエラーが返る



- 未コミットのRedoも含めて、エラー発生直前のBlockの状態へ

→ 同一セッションにてトランザクションを継続可能

# DB\_BLOCK\_CHECKING

## 検出時の参考ログ(Disk上のBlockが正常なケース)

### ■ Oracle Client

ORA-00607: データ・ブロックの変更中に内部エラーが発生しました。

**ORA-00600:** 内部エラー・コード, 引数: [*kdBlkCheckError*], [18], [35], [6107], [], [], [], [], [], [], [], []

### ■ Alert.log

Wed Aug 28 11:32:31 2013

**Corrupt Block Found**

*TSN = 47, TSNAME = TT*

*RFN = 18, BLK = 35, RDBA = 75497507*

*OBJN = 118071, OBJD = 118071, OBJECT = TAB1, SUBOBJECT*

*SEGMENT OWNER = TRY, SEGMENT TYPE = Table Segment*

破損Blockの情報

(表領域名、オブジェクト名、Block Address等)

Wed Aug 28 11:32:35 2013

Errors in file /u01/app/oracle/diag/rdbms/orcl/ORCL1/trace/ORCL1\_ora\_3341.trc (incident=4009905):

**ORA-00600:** 内部エラー・コード, 引数: [*kdBlkCheckError*], [18], [35], [6107], [], [], [], [], [], [], [], []

Incident details in: /u01/app/oracle/diag/rdbms/orcl/ORCL1/incident/incdir\_4009905/ORCL1\_ora\_3341\_i4009905.trc

...

Block recovery from logseq 8, block 6189 to scn 224893931

Recovery of Online Redo Log: Thread 1 Group 10 Seq 8 Reading mem 0

Mem# 0: +FRA/orcl/onlinelog/group\_10.1470.802603633

**Block recovery completed** at rba 8.6190.16, scn 0.224893932

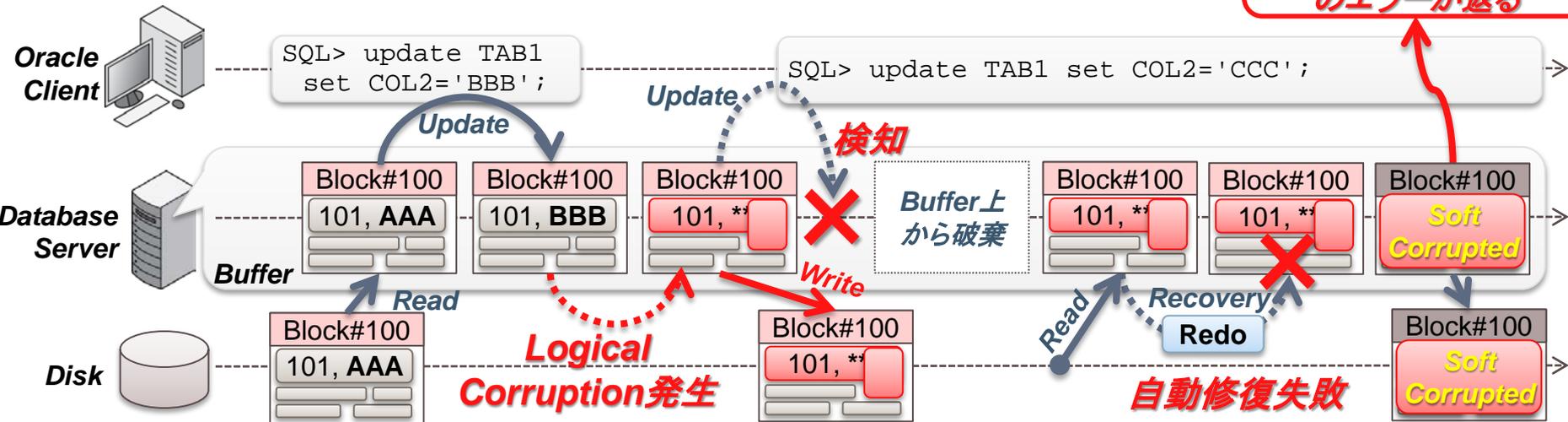
自動的にリカバリを実行している(成功時)

...

# DB\_BLOCK\_CHECKING

破損検出後の動作 (Disk上のBlockも破損しているケース)

Oracle Clientに  
ORA-607 & ORA-600  
のエラーが返る



- 自動修復に失敗していても、成功した際と同じエラーしか戻らない点に注意
  - この後、再度Blockへアクセスした場合、**ORA-1578** 又は **ORA-600**が発生  
→ **手動による Block Media Recoveryが必要 (ABRは発動しない)**

# DB\_BLOCK\_CHECKING

## 検出時の参考ログ(Disk上のBlockも破損しているケース)

### ■ Oracle Client

```
ORA-00607: データ・ブロックの変更中に内部エラーが発生しました。  
ORA-00600: 内部エラー・コード, 引数: [kdBlkCheckError],[18],[35],[6107],[],[],[],[],[],[],[],[],[],[
```

### ■ Alert.log

```
Wed Aug 28 13:38:32 2013  
Corrupt Block Found  
    TSN = 47, TSNAME = TT  
    ...(省略)  
ORA-00600: 内部エラー・コード, 引数: [kdBlkCheckError],[18],[35],[6107],[],[],[],[],[],[],[],[],[  
Block recovery from logseq 9, block 29085 to scn 224911344  
Block recovery completed at rba 9.29086.16, scn 0.224911345  
Corrupt Block Found  
    TSN = 47, TSNAME = TT  
    RFN = 18, BLK = 35, RDBA = 75497507  
Block recovery from logseq 9, block 29085 to scn 224911352  
    OBJN = 118071, OBJD = 118071, OBJECT = TT, SUBOBJECT =  
    SEGMENT OWNER = TRY, SEGMENT TYPE = Table Segment  
Recovery of Online Redo Log: Thread 1 Group 7 Seq 9 Reading mem 0  
    Mem# 0: +FRA/orcl/onlinelog/group_7.1540.802603355  
Block recovery stopped at EOT rba 9.29092.16  
Block recovery completed at rba 9.29092.16, scn 0.224911351
```

自動的にリカバリを試行しているが、  
Diskから読み込んだリカバリ対象のBlockが  
破損していると検知し、リカバリが停止している  
(リカバリ失敗)  
→ Blockには、Soft Corruptedとマークされる

# Soft Corrupt

*Physical Corruption / Logical Corruptionと同列の単語ではない*

- 破損が検出された (Oracleが破損と認識) Blockに付けられるマーク
  - このマークが付けられたBlockへアクセスした場合、ORA-1578が発生する

```
SQL> select * from TAB1 where COL1=100 ;
```

```
*
```

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18、ブロック番号35)

ORA-01110: データファイル18: '+DATA/orcl/datafile/tt.316.824637849'

```
SQL> select * from V$DATABASE_BLOCK_CORRUPTION;
```

FILE#	BLOCK#	BLOCKS	CORRUPTION_CHANGE#	CORRUPTIO
18	35	1	1.0696E+14	CORRUPT

# DB\_LOST\_WRITE\_PROTECT

## Lost Writeとは？

- ストレージ装置からBlockの書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
  - Data 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

# DB\_LOST\_WRITE\_PROTECT

## 概要

- Data Guard (Physical Standby Database) でLost Writeを検出する仕組み
  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)**

Blockを一意に特定する為の情報

Block更新時に増加するバージョン的な情報
  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

## 設定値毎の動作

- Primary DatabaseとStandby Databaseの両方での設定が必要

Value	Default	Lost Write検出対象	Primary Database	Standby Database
NONE	✓	N/A	N/A	N/A
TYPICAL		Read / Write表領域	DiskからBuffer Cacheへ	Redo適用処理の一環で、
FULL		Read / Write表領域 Read Only表領域	Blockを読込み時、 検証用Redoを生成	MRPが対象Blockと Redo内のSCNを比較

- Primary 又は Standby Databaseの **いずれかの設定がNONEの場合は無効**
  - Primaryが検証用Redoを生成しないので、Standbyで検証できない
  - Primaryで検証用Redoを生成しても、Standbyで検証しない

# DB\_LOST\_WRITE\_PROTECT

*Lost Writeの発生と検出のタイミングは一致しない*

- Lost Writeを検出するタイミングは？
  - Primary Database側でLost Writeが発生しているBlockをDiskからBuffer上へ読み込んだ際に生成されたRedoを、Standby DatabaseのMRPが検証した時
- ***Lost Write発生時(Diskへの書込みが欠落した瞬間)ではなく、Lost WriteしているBlockをDiskから読んだタイミング***
- 特定BlockにおいてLost Writeが発生していても、そのBlockを使用しない限り
  - 検索結果や更新トランザクションも正常
  - 不正データが他のBlockへ伝染することは無い

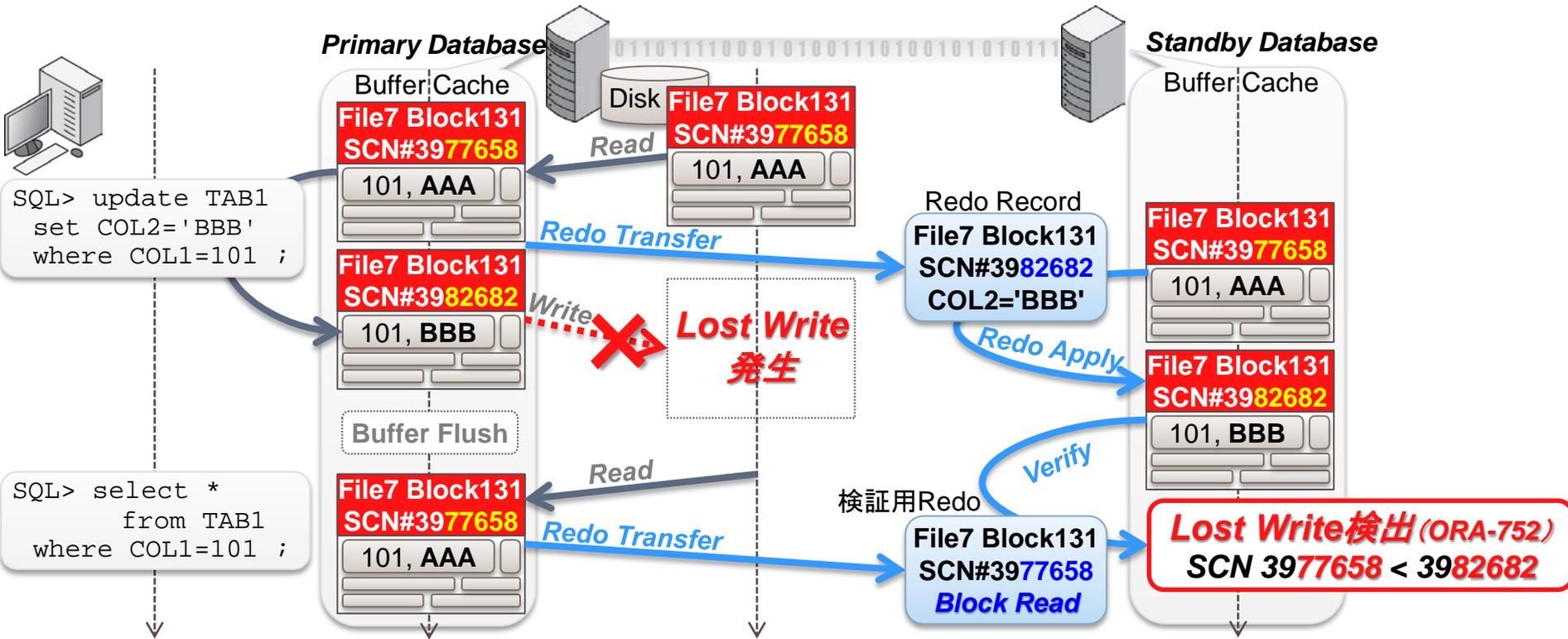
# DB\_LOST\_WRITE\_PROTECT

## 動作の補足

- 検証用Redoの生成はBuffer CacheへのBlock読み込み時に限定
  - Buffer Cacheを経由しない**Direct Path Read**では、**検証用Redoは生成しない**
- 非Data Guard環境でも、TYPICAL以上の設定で検証用Redoを生成
  - **メディア・リカバリ(ロール・フォワード)時にLost Writeを検証可能**
- **Standby Databaseで発生したLost Writeも検出可能**
  - Primary側で検証用Redo生成 + Standby側で検証、という役割は同じ
  - このケースに限定されるが、**ASM MirrorやABRによる自動修復が可能**(後述)

# DB\_LOST\_WRITE\_PROTECT

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



# DB\_LOST\_WRITE\_PROTECT

## Lost Write 検出後の動作 (Primary側でLost Writeしたケース)

- Primary側で発生したLost WriteをStandby側で検出する
  - Standby DatabaseのAlert.logに**ORA-752**の発生を記録
  - Standby Databaseのデータを保護する為に、MRPプロセスを自動停止
    - 以降のRedoの適用を中止し、不正データによるデータ汚染を広めない
  - PR<sub>xx</sub>プロセスのTrace File (xxx\_xxx\_prxx\_xxx.trc)にBlockの詳細を記録
    - Primary側で対象Blockにアクセスした際に生成したRedoレコードのDump
    - Standby側で保持している対象BlockのDump
      - これらの情報から、いつ以降にLost Writeが発生したかを確認可能

# DB\_LOST\_WRITE\_PROTECT

## Lost Write検出時のStandby DatabaseのAlert.log出力の一部抜粋 (Primary側でLost Writeしたケース)

```
Wed Oct 23 19:18:08 2013
Hex dump of (file 7, block 131) in trace file /u01/app/oracle/diag/rdbms/orcls/orcls1/trace/orcls1_pr02_1401.trc
Reading datafile '+DATA/orcls/datafile/lw.281.829593935' for corruption at rdba: 0x01c00083 (file 7, block 131)
Read datafile mirror 'DATA_0004' (file 7, block 131) found same corrupt data (logically corrupt)
Read datafile mirror 'DATA_0006' (file 7, block 131) found same corrupt data (logically corrupt)
STANDBY REDO APPLICATION HAS DETECTED THAT THE PRIMARY DATABASE
LOST A DISK WRITE OF BLOCK 131, FILE 7
NO REDO AT OR AFTER SCN 3987667 CAN BE USED FOR RECOVERY.
Recovery of Online Redo Log: Thread 1 Group 7 Seq 103 Reading mem 0
Slave exiting with ORA-752 exception
Errors in file /u01/app/oracle/diag/rdbms/orcls/orcls1/trace/orcls1_pr02_1401.trc:
ORA-00752: リカバリによりデータ・ブロックの書き込みの欠落が検出されました
ORA-10567: Redo is inconsistent with data block (file# 7, block# 131, file offset is 1073152 bytes)
ORA-10564: tablespace LW
ORA-01110: データファイル7: '+DATA/orcls/datafile/lw.281.829593935'
ORA-10561: block type 'TRANSACTION MANAGED DATA BLOCK', data object# 87637
Wed Oct 23 19:18:12 2013
Recovery Slave PR02 previously exited with exception 752
Wed Oct 23 19:18:12 2013
MRP0: Background Media Recovery terminated with error 448
Errors in file /u01/app/oracle/diag/rdbms/orcls/orcls1/trace/orcls1_pr00_1395.trc:
ORA-00448: バックグラウンド・プロセスが正常終了しました。
MRP0: Background Media Recovery process shutdown (orcls1)
```

# DB\_LOST\_WRITE\_PROTECT

## 続) PR<sub>xx</sub>のTrace File出力例 (Primary側でLost Writeしたケース)

Block Dump

```
Hex dump of (file 7, block 131) ← Lost Writeが検出されたBlock
Dump of memory from 0x00000000F03B0000 to 0x00000000F03B2000
0F03B0000 0000A206 01C00083 003CC55A 04010000 [.....Z.<.....]
...
```

Main Message

```
STANDBY REDO APPLICATION HAS DETECTED THAT THE PRIMARY DATABASE
LOST A DISK WRITE OF BLOCK 131, FILE 7
The block read on the primary had SCN 3977658 (0x0000.003cb1ba) seq 1 (0x01)
while expected to have SCN 3982682 (0x0000.003cc55a) seq 1 (0x01)
The block was read at SCN 3987667 (0x0000.003cd8d3), BRN:
CHANGE #1 TYP:2 CLS:6 AFN:7 DBA:0x01c00083 OBJ:87637 SCN:0x0000.003cb1ba SEQ:1 OP:23.2 ENC:0 RBL:1
...
```

Redo Dump

```
REDO RECORD - Thread:1 RBA: 0x000067.00000128.0010 LEN: 0x0034 VLD: 0x10
SCN: 0x0000.003cd8d3 SUBSCN: 1 10/23/2013 19:18:16
(LWN RBA: 0x000067.00000126.0010 LEN: 0003 NST: 0001 SCN: 0x0000.003cd8cf)
CHANGE #1 TYP:2 CLS:6 AFN:7 DBA:0x01c00083 OBJ:87637 SCN:0x0000.003cb1ba SEQ:1 OP:23.2 ENC:0 RBL:1
Block Read - afn: 7 rdba: 0x01c00083 BFT:(1024,29360259) non-BFT:(7,131)
scn: 0x0000.003cb1ba seq: 0x01
flags: 0x00000006 ( dlog ckval )
```

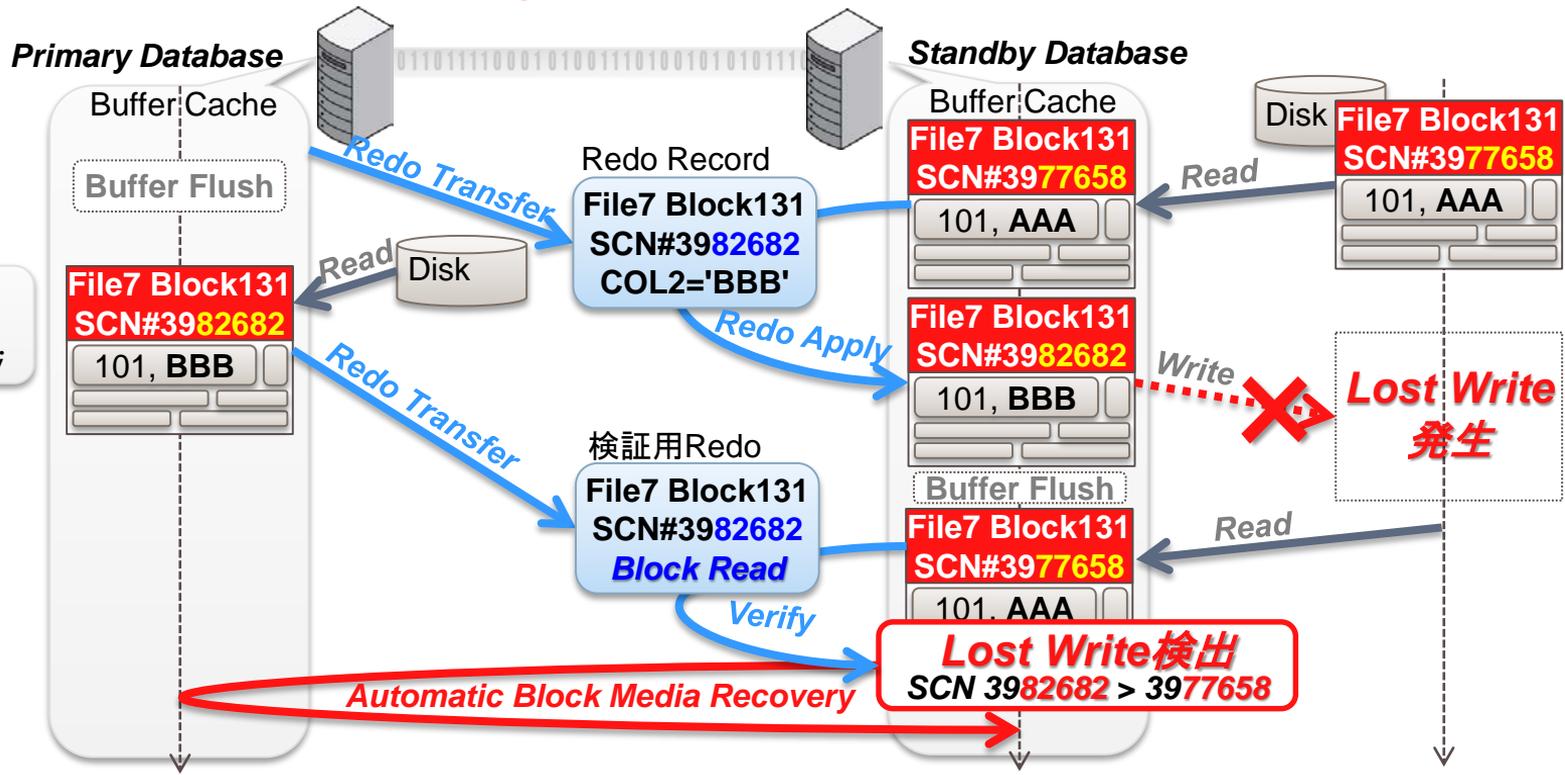
**3977658 (①) < 3982682 (②) より、Primary側でLost Writeが発生したと判定可能**  
**3982682 (②)の更新分でLost Writeが発生し、3987667 (③)のタイミングでBlockへアクセス(検出)**

# DB\_LOST\_WRITE\_PROTECT

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



```
SQL> select *  
      from TAB1  
     where COL1=101 ;
```



# DB\_LOST\_WRITE\_PROTECT

## Lost Write検出後の動作 (Standby側でLost Writeしたケース)

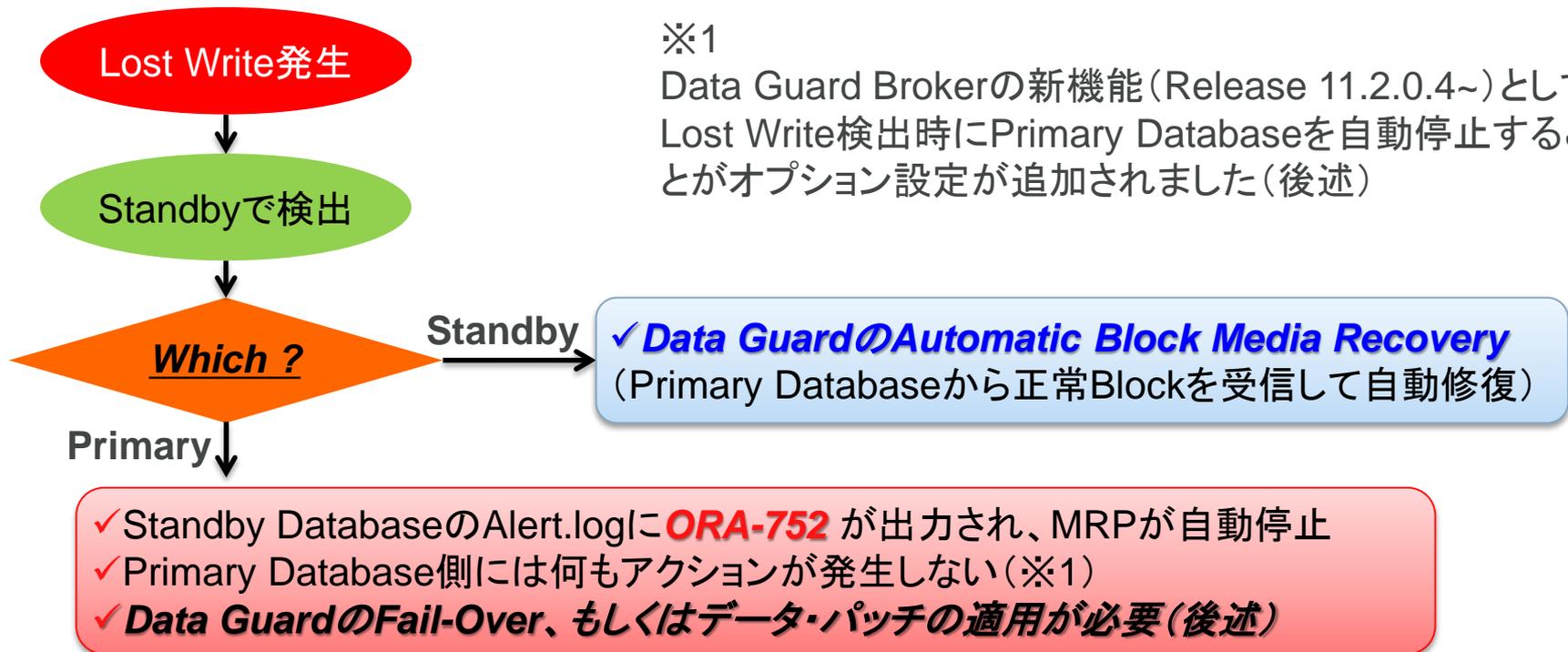
- Lost Write検出は、Standby Database側
  - Primary側で発生したLost Writeとは異なり、自動修復される
    - Primary Database側では、**最新の正常Block**を保持しているため
- **Data GuardのAutomatic Block Media Recoveryで自動修復**

```
### Standby's Alert Log
Fri Oct 25 11:33:11 2013
Automatic block media recovery requested for (file# 7, block# 132)
Automatic block media recovery successful for (file# 7, block# 132)
```

- 自動修復されれば、ORA-752は発生しない (Alert.logに出力なし)
  - MRPプロセスは正常に稼働し続ける

# DB\_LOST\_WRITE\_PROTECT

## Lost Write検出後の動作のまとめ



# DB\_ULTRA\_SAFE Parameter

検出レベルを高める3つのパラメータを一括設定

- DB\_ULTRA\_SAFEパラメータの値を変更することで、3つのパラメータ値を一括で変更することが可能
  - 本パラメータのデフォルト値はFALSE
  - 各パラメータを明示的に個別設定している場合、個別設定の値を優先

	DB_BLOCK_CHECKSUM	DB_BLOCK_CHECKING	DB_LOST_WRITE_PROTECT
OFF	TYPICAL	OFF	NONE
DATA_ONLY	FULL	MEDIUM	TYPICAL
DATA_AND_INDEX	FULL	FULL	TYPICAL

索引のLogical Corruptionを検出するか否かの違い

# Oracle DatabaseのData Protection

## 定期的な破損チェックを提供する機能

- Oracle Recovery Manager (RMAN)
  - VALIDATEコマンド
    - Data File、Backup File(Image Copy / Backup Piece)の破損チェック
  - CHECK LOGICAL句
    - Physical Corruptionのチェックに追加して、Logical Corruptionをチェック
    - BACKUP / VALIDATE / RECOVER コマンドとの併用可能
- ANALYZE TABLE <TableName> VALIDATE STRUCTURE CASCADE ;
  - 表と索引Blockの間の不整合を検出可能

# RMAN> Validate Check Logical

## 参考出カログ

```
RMAN> validate check logical datafile 18 ;

Starting validate at 28-AUG-13
using target database control file instead of recovery catalog
...
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation
input datafile file number=00018 name=+DATA/orcl/datafile/tt.316.824637849
channel ORA_DISK_1: validation complete, elapsed time: 00:00:01
List of Datafiles
=====
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
18  OK      1                277          320          224873829
   File Name: +DATA/orcl/datafile/tt.316.824637849
   Block Type Blocks Failing Blocks Processed
   -----
   Data          0             4
   Index         0             1
   Other         0             38
Finished validate at 28-AUG-13
```

# SQL> ANALYZE TABLE VALIDATE STRUCTURE CASCADE ;

*KROWN#68739 参考出カログ*

```
SQL> ANALYZE TABLE TAB1 VALIDATE STRUCTURE CASCADE ;
```

\*

1行でエラーが発生しました。

ORA-01499: 表または索引の相互参照エラーです。トレース・ファイルを調べてください。

- Physical Block Corruptionが発生していれば、ORA-1578が発生
- 上記の例のようにORA-1499が発生した場合は、表と索引の間で不整合が発生している状態
  - どちらのBlockで障害が発生しているのかを切り分ける必要有り
    - 例: FULLヒント、もしくはNO\_INDEXヒントを使用して全表検索を実行

# Oracle Database の Data Protection



修復オペレーション



ORACLE

# Block Corruption が及ぼす影響

ビジネスへ直接的なダメージ

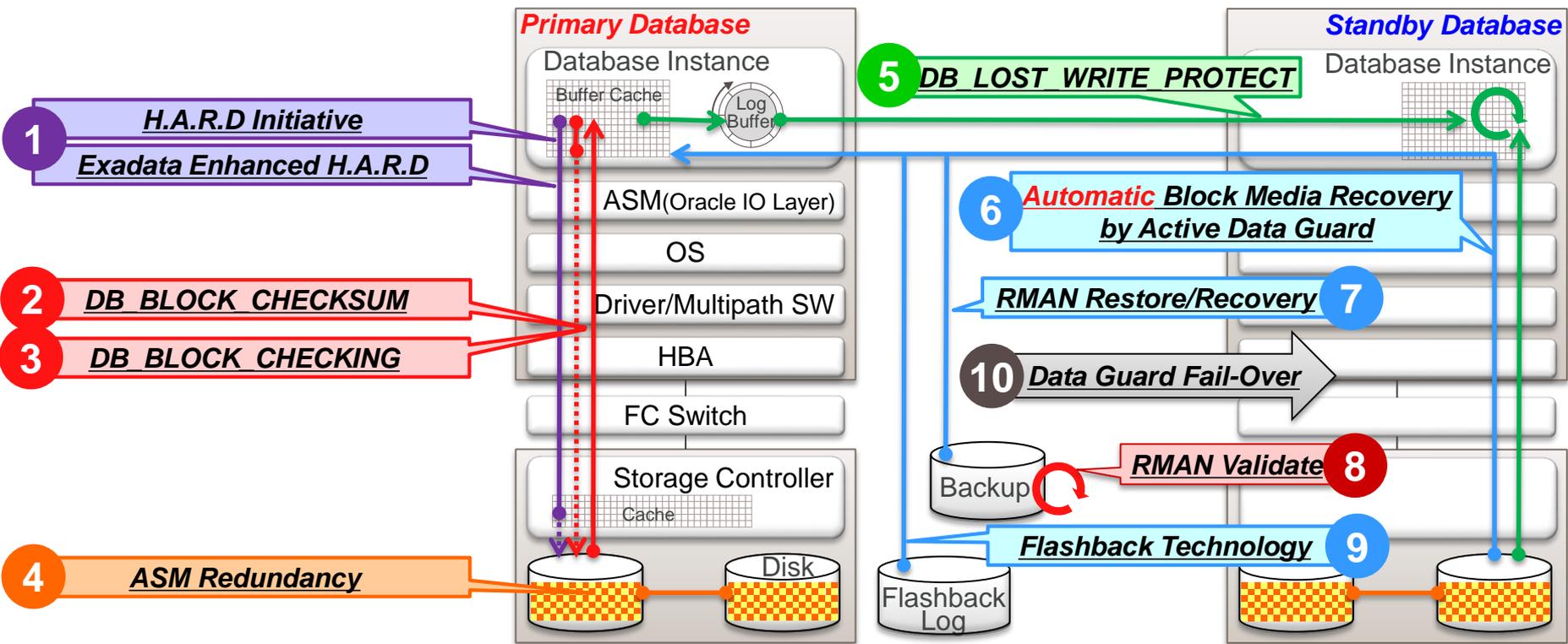


- ✓ データ損失による業務停止
- ✓ 長時間化するデータ復旧
- ✓ 復旧作業の人的ミス、二次災害
- ✓ 原因究明の長期化

→ 迅速な障害の認識と正確な修復が必須

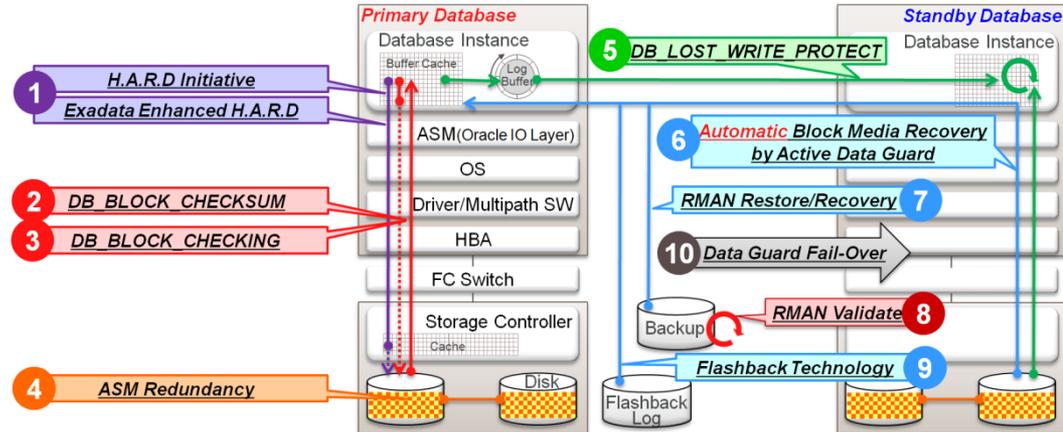
# Oracle MAAのデータ保護機能一覧

Oracle Database 11g Release 2



# Oracle MAA

## 主要な修復機能



#	Product / Function	Auto or Manual ?
4	Oracle ASM / Mirroring	<i>Auto</i>
6	Oracle Active DG / Auto Block Media Recovery	<i>Auto</i>
7	Oracle RMAN / Media Recovery	<i>Manual</i>
9	Flashback Technology (Flashback Database)	<i>Manual</i>
10	Oracle DG / Fail-Over	<i>Auto or Manual*</i>

\* Data Guard Brokerを構成したDG環境では、Fast Start FailoverでF/Oの自動化が可能  
 さらに、Release 11.2.0.4以降、Primary Lost Write Actionプロパティが追加され、  
 Lost Write検出時にPrimary側を自動的にABORTすることが可能

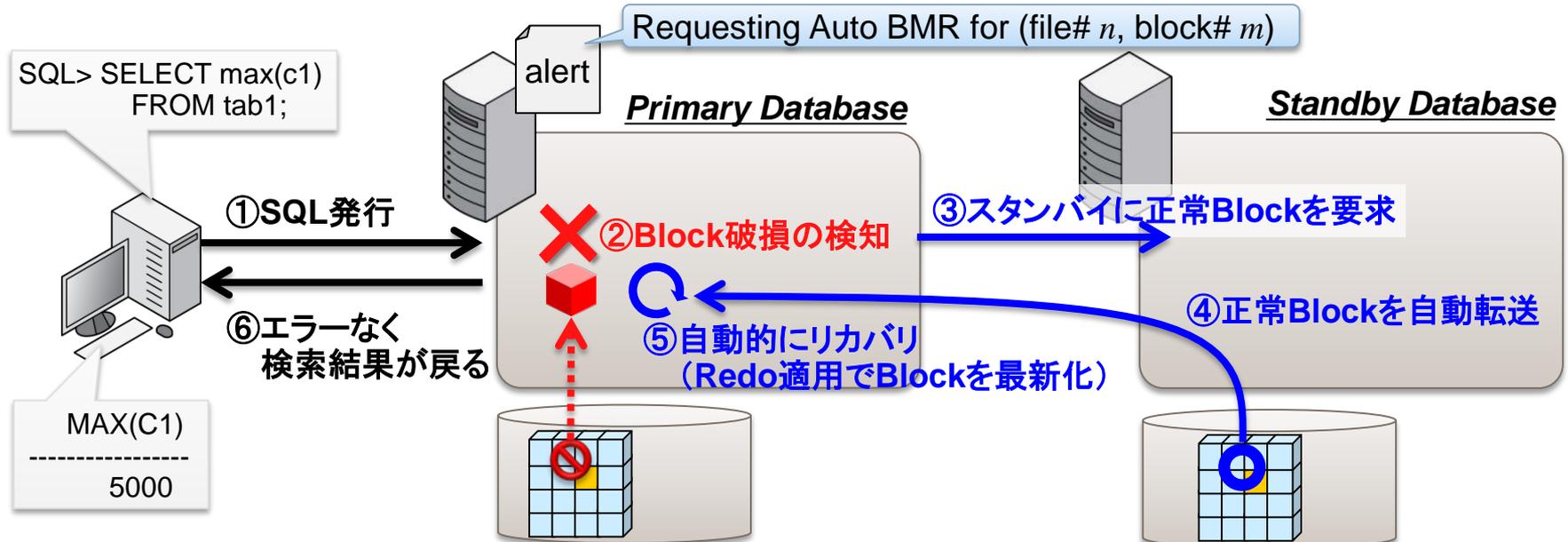
# Oracle ASM / Mirroring

*Oracle Clientに透過的、かつ自動的にBlockを修復*

- Normal(二重化) 及び High Redundancy(三重化)の構成が必須
- 破損検知時にMirrorデータを参照して自動修復
  - Oracle Clientに対して透過的(ORAエラーは戻らない)
  - Primary側で発生するLost Writeに関しては対象外
    - Standby側でLost Writeを検証する際は、Mirrorも使用して誤検知を防止
- Redo Blockの破損時にもMirrorデータを参照
  - Normal/High RedundancyのASM Diskgroup上にRedo Log fileを配置
    - Doc ID 1274318.1

# Automatic Block Media Recovery

Active Data Guardによる透過的なBlock修復



Oracle Clientに対して透過的に、破損ブロックを自動修復

# Automatic Block Media Recovery

## 動作の追加情報

- Standby側でBlock破損を検出した場合は、**逆向きABR**で自動修復
  - 対象のBlock破損
    - Standby側でのPhysical Block Corruption
      - DB\_BLOCK\_CHECKSUMの機能で検出したもの
      - **Soft Corruptとマーク済みのBlockへアクセスした場合は動作しない**  
(あくまで、マークする前に修復を試みるのがABR)
    - Standby側でLost WriteしたBlock
      - **Primary側でLost WriteしたBlockは対象外**  
(不正Blockを更新した際に生成された不正なRedoでリカバリ不可の為)

# Automatic Block Media Recovery

## Block破損の検知とABRで自動修復された際の参考ログ

### ■ Primary Database's Alert.log

```
Mon Aug 19 18:44:27 2013
Hex dump of (file 3, block 35) in trace file /u01/app/oracle/diag/rdbms/orcl/ORCL1/trace/ORCL1_ora_16202.trc
Corrupt block relative dba: 0x00000023 (file 3, block 35)
Fractured block found during buffer read
Data in bad block:
  type: 6 format: 2 rdba: 0x00000023
  last change scn: 0x0000.0cff8994 seq: 0x1 flg: 0x04
  spare1: 0x0 spare2: 0x0 spare3: 0x0
  consistency value in tail: 0x76150601
  check value in block header: 0xcf5
  computed block checksum: 0xffe3
Reading datafile '+DATA/orcl/datafile/tbs20.319.822054891' for corruption at rdba: 0x00000023 (file 3, block 35)
Read datafile mirror 'DATA_0002' (file 3, block 35) found same corrupt data (no logical check)
Read datafile mirror 'DATA_0000' (file 3, block 35) found same corrupt data (no logical check)
Automatic block media recovery requested for (file# 3, block# 35)
Mon Aug 19 18:44:29 2013
Automatic block media recovery successful for (file# 3, block# 35)
```

破損の検知

ASM Mirrorで自動修復を試行

ABRで自動修復成功

※ 上記からも確認出来るが、自動修復に成功した場合はORA-1578は記録されない

# Oracle Recovery Manager

## RECOVERコマンドによるBlock単位での修復

- 修復対象Blockが格納されているData Fileはオンライン状態で実行可能
- Lost WriteによるBlock破損は対象外

```
RMAN> RECOVER DATAFILE n BLOCK m,... ;
```

- 特定のBlockを指定して修復するコマンド
  - 基本的には、V\$DATABASE\_BLOCK\_CORRUPTIONビューで表示されるBlock(破損検出済みBlock)を指定する(実行時にも破損チェックされる)

```
RMAN> RECOVER CORRUPTION LIST ;
```

- V\$DATABASE\_BLOCK\_CORRUPTIONビューで表示されている全Blockを一括で修復するコマンド

# Oracle Recovery Manager

## Block単位でのRecoveryのベースとなる正常Blockは？

- ブロック単位のRecoveryとは言えども、Restore可能な正常Blockが必要
  - 正常Blockの検索先の優先順位は次の通り
    1. *Active Data GuardのPhysical Standby Database*
    2. *Flashback Log (Recovery実行中のDatabase内)*
    3. *RMAN Image Backup (Recovery実行中のDatabase内)*
  - 上記のいずれかで正常BlockがRestoreされれば、自動的にリカバリで最新化
  - もし、Block単位で修復不可の場合、データ・ファイル単位でのRecoveryが必須

# Oracle Enterprise Manager Cloud Control 12<sup>c</sup>

## *Data Recovery Advisor*による修復コマンドの自動生成



- Database環境の状況に応じた最適なRecoveryを容易に実現
  - Block単位でのRecoveryが可能か否かまで考慮された修復コマンド
  - **Oracle Enterprise Manager**で簡単な操作でRecovery実行まで可能
- 実行例
  - 次のDatabase環境において、Physical Block CorruptionをEM(**Data Recovery Advisor**)を使用して修復する手順をご紹介します
    - Active Data Guard環境無し
    - Flashback Log有り(但し、保存の期限切れ状態)
    - RMAN Image Copy Backup無し

# Oracle Enterprise Manager Cloud Control 12<sup>c</sup>

## Incident Managerによる障害の早期把握



✓ 障害発生時にメールによる通知を行う設定も可能

重大度	サマリー	ステータス	最終更新	タイプ	カテゴリ
重大	チェッカ実行によって1の新規永続データ障害が見つかりました。	新規	2013/10/31 9:53:13 JST	インシデント	フォルト
問題	問題: ORA 1578	新規	2013/10/31 9:53:13 JST	問題	フォルト 診断
問題	時間/行番号: Thu Oct 31 09:46:39 2013/08/04でOracleのデータブロック破損が/u01/app/oracle/diag/rdbms/orcl/c	新規	2013/10/31 9:53:13 JST	インシデント	フォルト 診断

問題の詳細

ID	113102
問題キー	ORA 1578
ターゲット	ord2 (データベース: インスタンス)
インシデントの数	1
第1インシデント	2013/10/31 9:53:13 JST
最終インシデント	2013/10/31 9:53:13 JST
パッケージ済	いいえ
サービス・リクエスト#	-
バグ#	-

ガイドされた解決

診断  
サポート・ワークベンチ: 問題の詳細

アクション  
サポート・ワークベンチ: パッケージ診断

# Oracle Enterprise Manager Cloud Control 12<sup>c</sup>

## Data Recovery Advisorの実行

解決

SQL修復アドバイザ

データリカバリアドバイザ

障害の表示および管理

結果セットに表示されるデータをフィルタ処理するには、ドロップダウンの値を選択し、オプションで障害の説明と影響を入力してください。

障害の説明	影響	優先度	ステータス	検出時間
		CRITICALまたはHIGH	OPEN	すべて

アドバイザ

選択	障害の説明	影響	優先度	ステータス
<input type="checkbox"/>	▼ データ障害			
<input checked="" type="checkbox"/>	▶ データファイル5: '+DATA/orcl/datafile/tt.259.825875913'には破損したブロックが1つ以上含まれています	表領域TT内の一部のオブジェクトが使用できない可能性があります	HIGH	OPEN

ヒント すべてのCRITICAL障害は、「アドバイザ」の前に選択する必要があります。すべてのCRITICAL障害は、「優先度を高く設定」または「優先度を低く設定」

# Oracle Enterprise Manager Cloud Control 12<sup>c</sup>

## Data Recovery Advisorの実行結果

- Database環境の状況(3ページ前)を考慮し、Block単位でのRecoveryが不可能と判断している

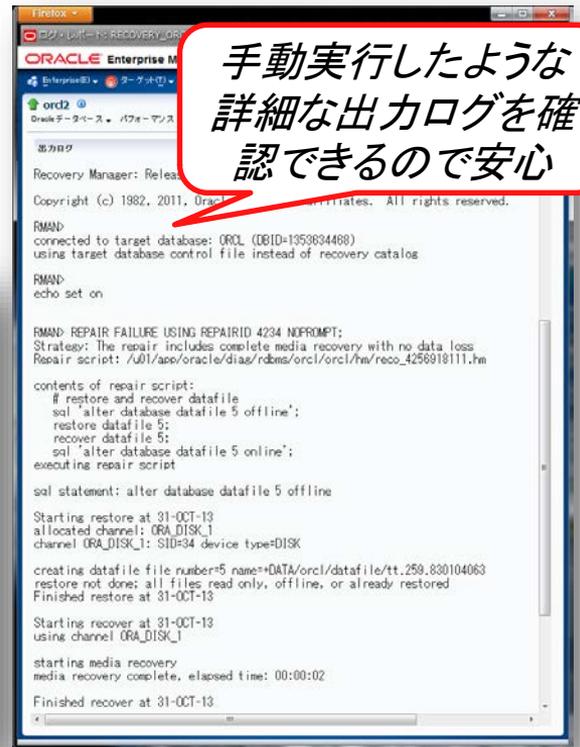


### RMANスクリプト

```
# restore and recover datafile
sql 'alter database datafile 5 offline';
restore datafile 5;
recover datafile 5;
sql 'alter database datafile 5 online';
```

# Oracle Enterprise Manager Cloud Control 12<sup>c</sup>

Recovery Jobの発行 (Advisorの結果スクリプトを実行)



# Flashback Technology

## *Flashback Database (Flashback Log)の活用例*

- Flashback Databaseコマンド
  - 人的ミス(データ削除や不適切な更新処理の実行等)からの迅速な復旧
  - Primary側でLost Write発生した為に、Data GuardでFail-Over実行後、旧Primaryを新Standby環境として迅速に復旧させる場合
- Flashback Log
  - 上記のFlashback Databaseコマンド実行時
  - 前項で紹介した、RMANによるBlock単位でのMedia Recovery実行時

※ Flashback Logは、「SQL> alter database flashback on ;」  
又は、保証付きRestore Pointを作成することで生成を開始

# Oracle Data Guard

## *Physical Standby DatabaseへのFail-Over*

- Oracle Data GuardのPhysical Standbyは完全な複製Database
  - Primary Databaseで生成されたRedoを同じようにStandby側で適用
    - Primary側のBlock更新処理は、Redoを先に生成してBlockへ適用している
- **必ずFail-Overすることが可能なDatabase環境**
- 特に、Primary Database側でLost Writeが発生した場合に有効
  - 正常データを使用した業務の迅速な再開
  - 業務影響なく、Lost Write発生要因の調査が可能
    - 原因不明な状況で、PrimaryのH/Wを使用し続けることのリスクを考慮

# Oracle Data Guard

## Lost Write検出後のFail-Over時に考慮すること(1)

- Lost Write検出からFail-Overまでに実行されたトランザクションは失う
  - Primary側はStandby側でLost Writeを検出しても稼働を続けている
    - 業務トランザクションにより、新たな変更が発生している状況
    - *正しい変更もあるが、Lost WriteしたBlockを使用した不正な変更も混在*
  - Standby側はデータ汚染を防ぐ為、検出以降のRedo適用は停止している
    - *如何に迅速に、Primary側を停止してFail-Overするかが重要*
    - *Release 11.2.0.4以降、Data Guard BrokerのPrimary Lost Write Actionプロパティが追加*
    - *Lost Write検出時にPrimary側を自動的にABORTすることが可能*

# Oracle Data Guard

## Lost Write検出後のFail-Over時に考慮すること(2)

- Fail-Over直後は、Data Guard構成が崩れている状態
  - 旧Primaryと新Primary(元Standby)は別の道を進んでいる為
    - *旧Primary Databaseを新Standby Databaseへ作成し直す必要有り*
- 作成し直す方法

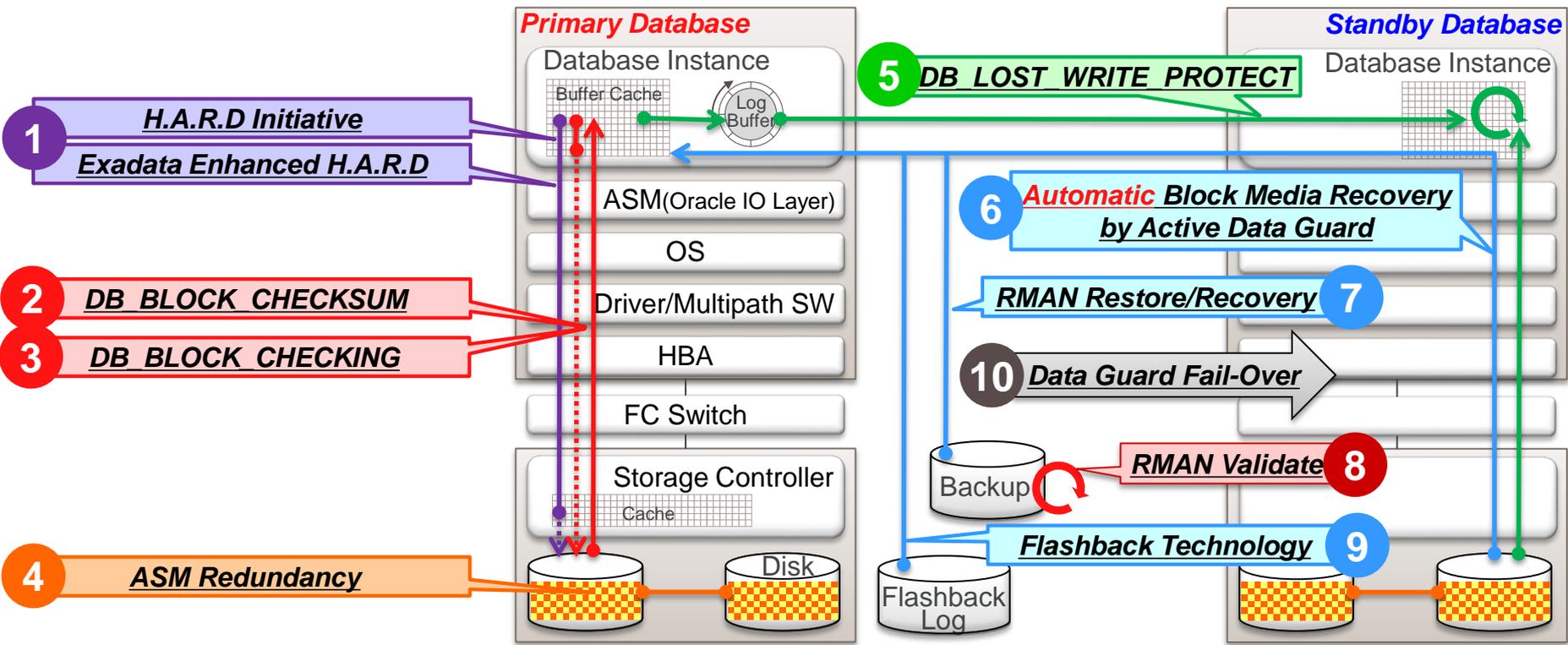
#	方法	新Standbyに対する性能影響	所要時間
1	新Standbyを使用した複製(Duplicateコマンド)	有り	数時間~数日
2	旧Primary上のBackupを使用した不完全リカバリ	無し	数時間
3	旧PrimaryをFlashback Database	無し	数分

# **Oracle's Data Protection** **Conclusion**



# Data Protection

## Oracle Database 11g Release 2



# Data Protection

## 3つの初期化パラメータの検知動作と修復方法のサマリ

Parameter	主な検知対象	主な検知のタイミング	Oracle Clientへ戻る主なエラー	Alert.logに記録されるORAエラー	主な修復方法 (自動は太文字)
<b>DB_BLOCK_CHECKSUM</b>	Physical Corruption	<ul style="list-style-type: none"><li>•DiskからのBlock読込後</li><li>•メモリ上での更新前 (FULL設定のみ)</li></ul>	<ul style="list-style-type: none"><li>•無し(自動修復時)</li><li>•ORA-1578</li></ul>	<ul style="list-style-type: none"><li>•無し(自動修復時)</li><li>•ORA-1578</li></ul>	<ul style="list-style-type: none"><li>•<b>ASM / Mirroring</b></li><li>•<b>Active Data Guard / ABR</b></li><li>•RMAN Block Media Recovery<ul style="list-style-type: none"><li>•Active Data Guard</li><li>•Flashback Technology</li><li>•RMAN Backup</li></ul></li></ul>
<b>DB_BLOCK_CHECKING</b>	Logical Corruption	<ul style="list-style-type: none"><li>•メモリ上でのBlock変更後</li><li>•Direct操作は対象外</li></ul>	<ul style="list-style-type: none"><li>•ORA-600</li><li>•ORA-607</li></ul>	<ul style="list-style-type: none"><li>•ORA-600 [kdblkCheckError]</li></ul>	<ul style="list-style-type: none"><li>•RMAN Block Media Recovery<ul style="list-style-type: none"><li>•Active Data Guard</li><li>•Flashback Technology</li><li>•RMAN Backup</li></ul></li></ul>
<b>DB_LOST_WRITE_PROTECT</b>	Lost Write	<ul style="list-style-type: none"><li>•DiskからのBlock読込後 (タイムラグ有)</li><li>•Direct操作は対象外</li></ul>	<ul style="list-style-type: none"><li>•無し (非同期で検知)</li></ul>	<ul style="list-style-type: none"><li>•Standby側で検知時にORA-752 (Primary側は無し)</li></ul>	<ul style="list-style-type: none"><li>•Data Guard / Fail-Over (<b>Broker設定で自動化可能</b>)</li></ul>

# Conclusion

## Data Protection



Oracle Database はデータを保護する包括的な高可用性ソリューションを提供します。

- *DB\_ULTRA\_SAFE Parameter*
- *Oracle Data Guard*
- *Oracle Recovery Manager*
- *Oracle Enterprise Manager*

**Hardware and Software**

**ORACLE®**

**Engineered to Work Together**

ORACLE®