

ORACLE®

ORACLE®

Oracle Database 12c Release 1 Recovery Manager (RMAN)

日本オラクル株式会社

ORACLE®
DATABASE 12^c



Plug into the **Cloud**.

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。

RMAN 新機能

表単位のリカバリ

- ・ 特定の表のみを指定した時間にまでリカバリ

12c

マルチセクション対応

- ・ マルチセクションでのバックアップの対応機能強化

データベースの複製

- ・ 稼働中のデータベースの複製での対応機能強化

クロス・プラットフォーム

- ・ 異なるプラットフォーム間でのリストア・リカバリ機能の強化

1. 表単位のリカバリ

"特定の表のみを指定した時間
までリカバリ"

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

ORACLE[®]

表単位のリカバリ

基本 任意の表を指定した時間までリカバリ可能 (*1)

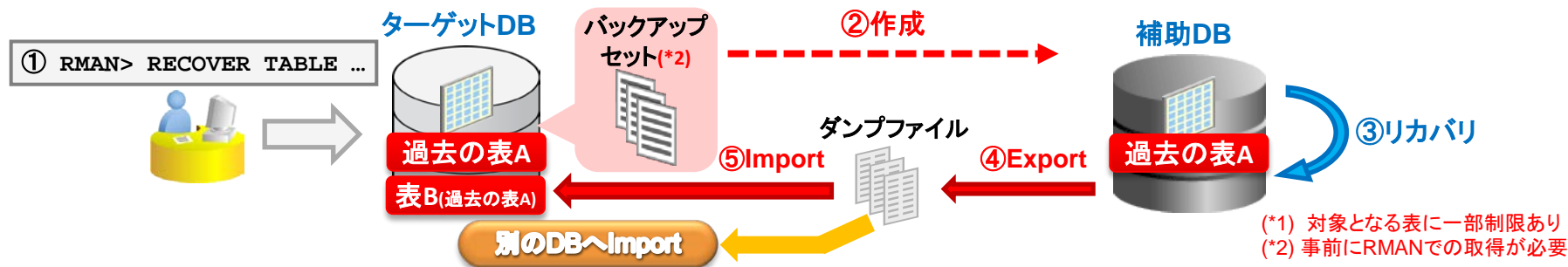
基本 表の DROP / TRUNCATE / 定義変更時でも使用可能

応用 表名・表領域名を変更してリカバリ可能

任意の時点への復旧	●
特定表に絞った復旧	●
構成変更時の復旧	●
表削除時の復旧	●

➤ 例えば、表 A を1日前の状態にリカバリし、表Bとして作成

応用 データだけを Data Pump のダンプファイル形式で出力可能



表単位のリカバリ

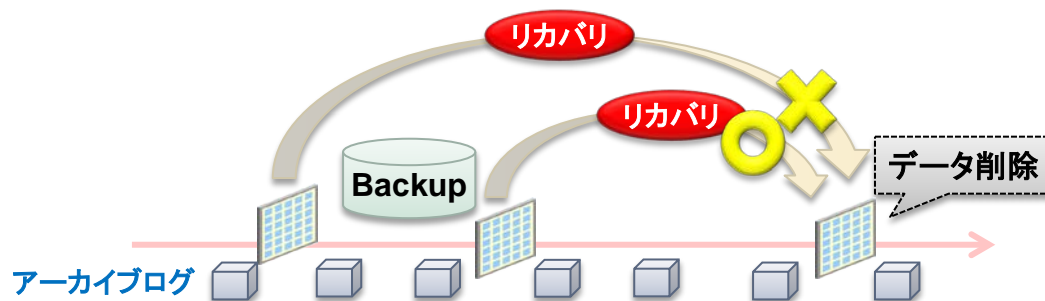
概要

□ リカバリ可能なオブジェクト

- 表
- パーティション表

□ 必要なファイル

- データベースのバックアップ (RMAN)
 - リカバリする時間より前に取得したもの
- アーカイブログファイル
 - バックアップ取得後、リカバリする時間までのもの



表単位のリカバリ

特徴

◆ ターゲットDBへの負荷が低い

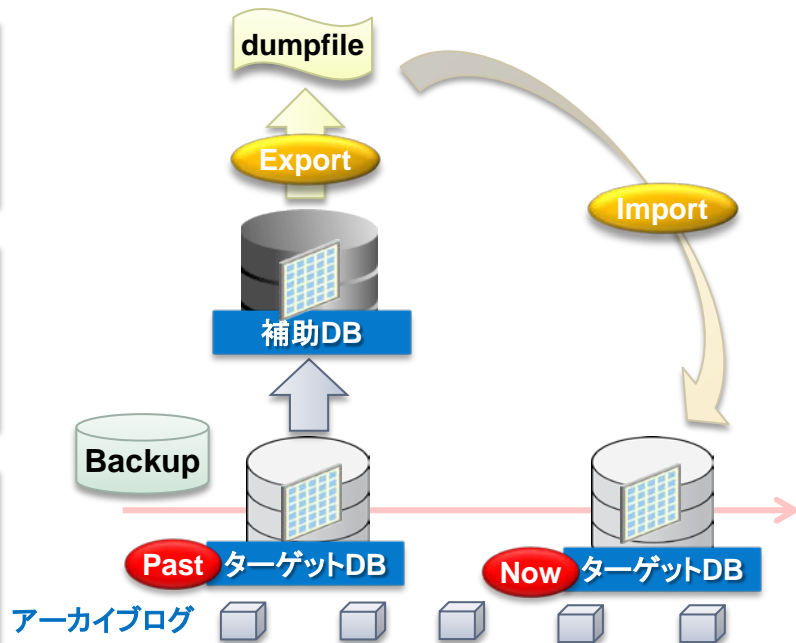
- 実行処理は Import のみ
- データ生成は補助 DB で実行

◆ 追加設定や情報取得が最小限

- アーカイブログモードの設定のみ
- RMAN のバックアップより実行可能

◆ 任意の時点のデータ生成が可能

- 表の削除・構成変更時にも復旧可能
- 論理バックアップ(Export)の定期取得の代替に

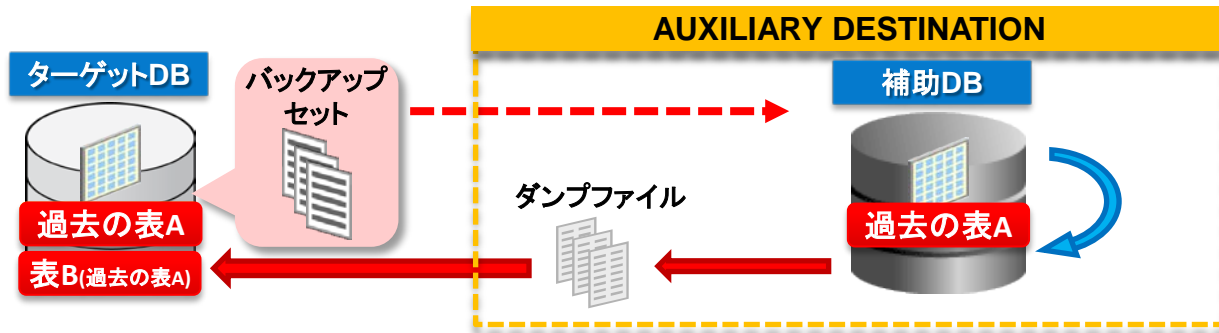


表単位のリカバリ

実行方法：接続と作業用ディレクトリの指定

- SYSDBA / SYSBACKUP 権限を持つユーザで接続
- 補助インスタンスの作成ディレクトリの指定が必要

```
RECOVER TABLE <スキーマ名>.<表名>  
UNTIL xxxx  
AUXILIARY DESTINATION '<ディレクトリ>';
```



表単位のリカバリ

実行方法：リカバリ時間の指定

□ リカバリ時間の指定

➤ SCN

```
UNTIL SCN <SCN番号>
```

➤ 時間

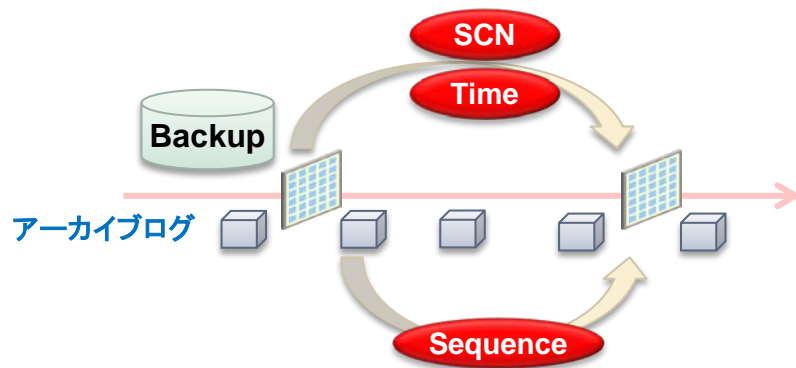
```
UNTIL TIME '<時間>'
```

➤ シーケンス番号

```
UNTIL SEQUENCE <シーケンス番号>
```

実行例

```
RECOVER TABLE SCOTT.EMP  
UNTIL TIME '2013-02-01 12:00:00'  
AUXILIARY DESTINATION '/tmp/rman';
```



表単位のリカバリ

実行方法：REMAP

□ REMAP による名前の変更

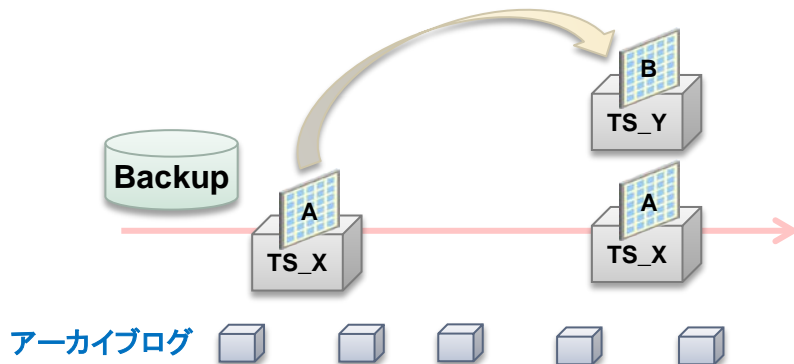
- リカバリ後の表名の指定
- リカバリ後の配置表領域の指定

実行例

```
RECOVER TABLE SCOTT.EMP  
UNTIL SEQUENCE 200  
AUXILIARY DESTINATION '/tmp/rman'  
REMAP TABLE SCOTT.EMP:NEW_EMP  
REMAP TABLESPACE USERS:NEW_USERS;
```

```
REMAP TABLE <スキーマ名>.<現在の表名>:<変更後の表名>
```

```
REMAP TABLESPACE <現在の表領域名>:<変更後の表領域名>
```



表単位のリカバリ

実行方法：Exportダンプファイルの生成

□ Data Pump のダンプファイルの生成

- ダンプファイルのみ生成
- ダンプファイルの出力ディレクトリの指定
- ダンプファイル名の指定

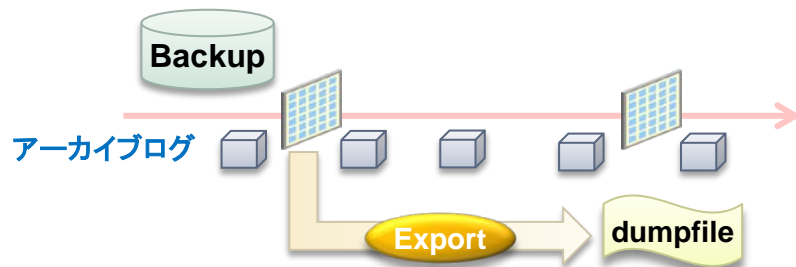
実行例

```
RECOVER TABLE SCOTT.EMP  
UNTIL SCN 1863847  
AUXILIARY DESTINATION '/tmp/rman'  
DATAPUMP DESTINATION '/tmp/datapump'  
DUMP FILE 'emp_export.dmp'  
NOTABLEIMPORT;
```

```
NOTABLEIMPORT
```

```
DATAPUMP DESTINATION '<ディレクトリ名>'
```

```
DUMP FILE '<ファイル名>'
```



表単位のリカバリ

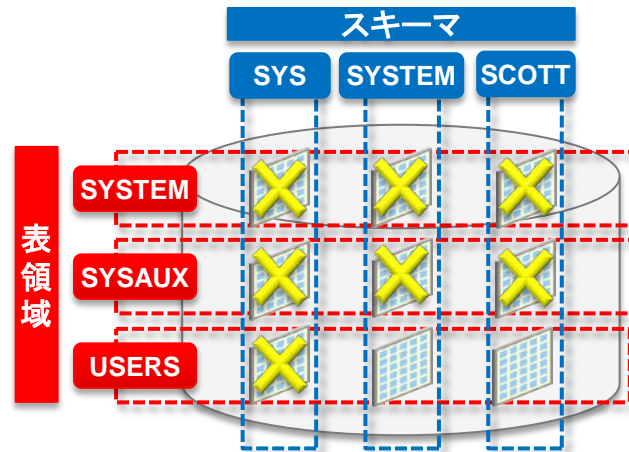
制限・注意点

□ 制限

- リカバリ対象とならないケース
 - SYSスキーマの表
 - SYSTEM / SYSAUX 表領域上の表

□ 注意点

- REMAP TABLE オプション未使用時は、事前に表の削除が必要
- REMAP TABLE オプション使用時は、制約・索引は作成されない
- 表名が RMAN の予約語の場合、クォーテーションが必要
 - 表名で指定できる予約語と、RMAN の予約語が異なる
 - 例えば、「TEST」「STANDBY」は表名として使用可能だが、RMAN では予約語



2. マルチセクション対応

"マルチセクションでのバックアップ
の対応機能を強化"

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

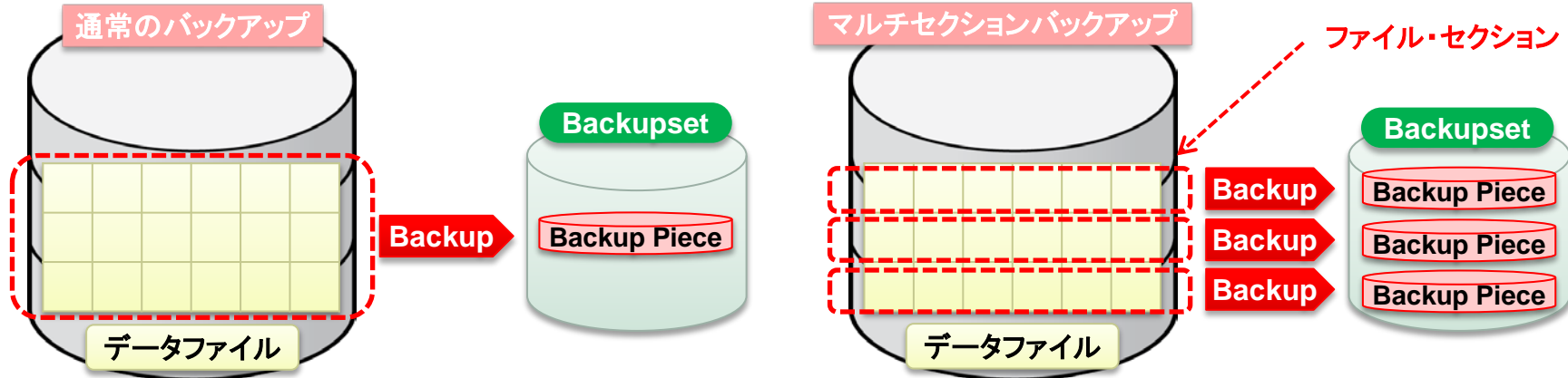
ORACLE[®]

マルチセクション対応

概要

□ マルチセクション・バックアップ

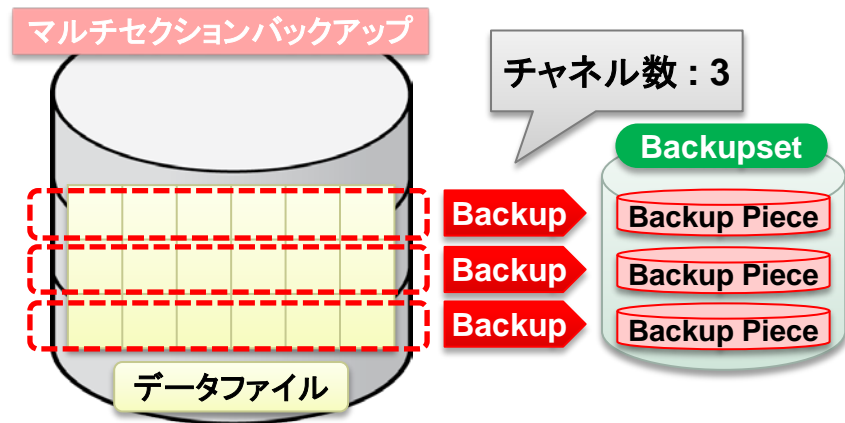
- 指定したサイズ(セクション・サイズ)毎にファイル・セクションとして分割
- ファイル・セクション単位にバックアップを取得
- 11gR1 より使用可能



マルチセクション対応

メリット

- マルチセクションバックアップのメリット
 - チャンネルを複数用意することによる平行処理が可能
 - データファイルのサイズが大きい場合に高速化が期待できる



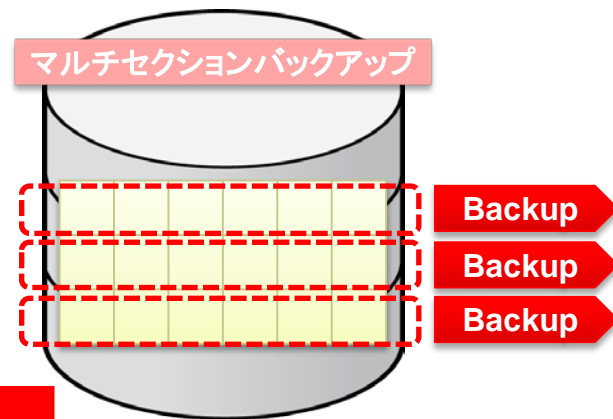
実行例

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 3 ;  
  
BACKUP  
SECTION SIZE <サイズ>  
DATABASE ;
```


マルチセクション対応

11g での制限と 12c での改良点

- 11g での制限
 - 全体バックアップのみ対応
- 12c での改良点
 - 増分バックアップ、イメージ・コピーにも対応



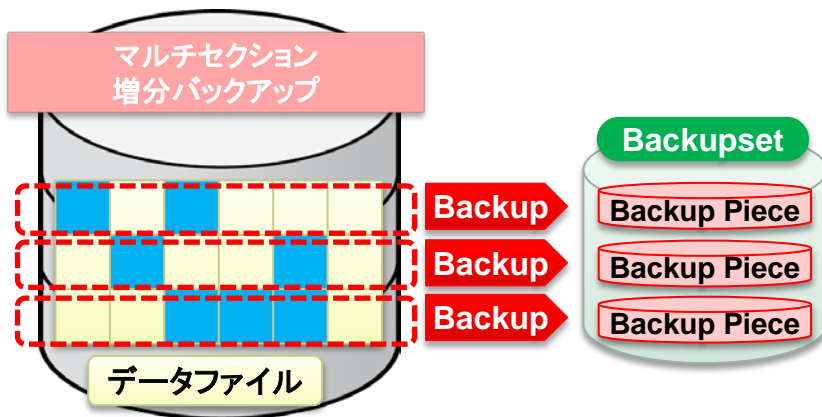
	全体バックアップ	増分バックアップ	イメージ・コピー
11gR1/R2	●	×	×
12c	●	●	●

マルチセクション対応

増分バックアップ

□ マルチセクション・増分バックアップ

- セクション内の更新部分のみをそれぞれバックアップ



実行例

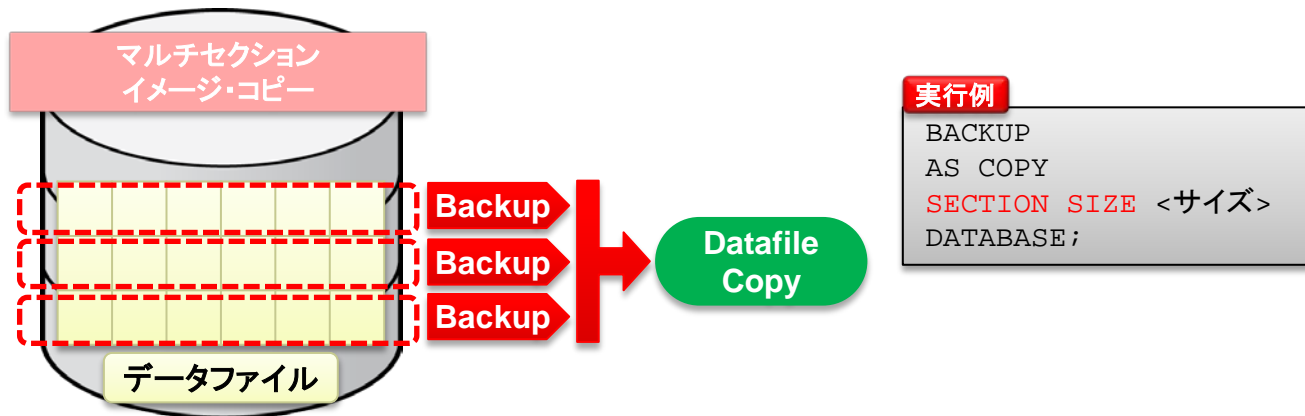
```
BACKUP  
INCREMENTAL LEVEL 1  
SECTION SIZE <サイズ>  
DATABASE ;
```

マルチセクション対応

イメージ・コピー

□ マルチセクション・イメージ・コピー

- セクションに分割してバックアップ後、一つのデータファイル・コピーに集約



3. データベースの複製

"稼働中のデータベース複製での
対応機能の強化"

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

ORACLE[®]

アクティブなデータベースの複製

アクティブなデータベースの複製

□ 稼働中 DB の複製

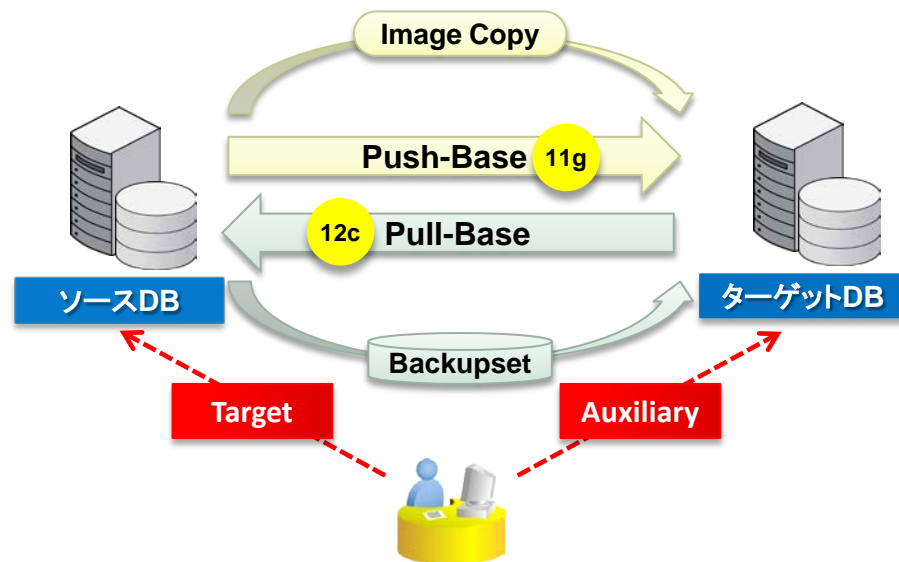
- 稼働中の DB (ソース)のバックアップを取得し、複製 DB (ターゲット)を作成
- 2種類の方法

Push-Base (11g~)

- ソース DB がターゲット DB へイメージ・コピーを転送

Pull-Base (12c~)

- ターゲット DB がソース DB のバックアップセットを取得



アクティブなデータベースの複製

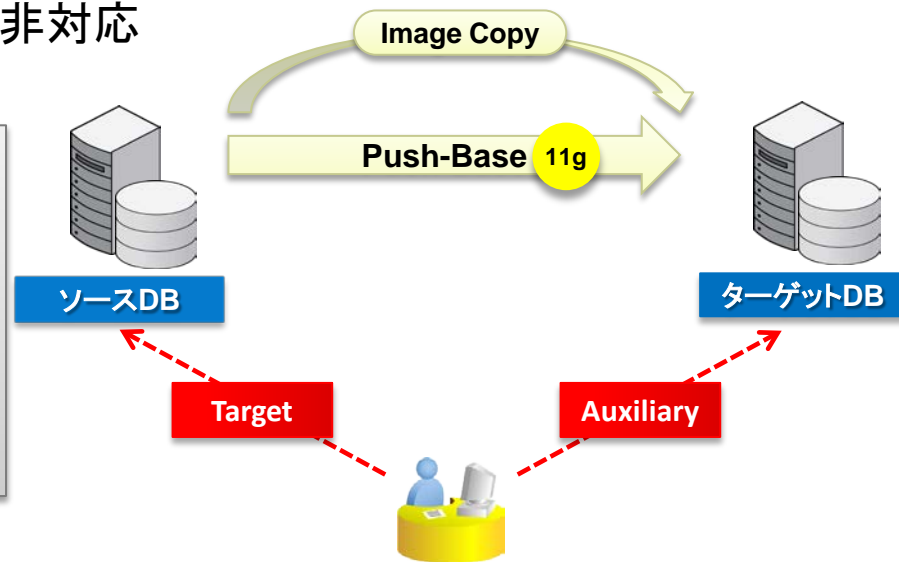
Push-Based Active Duplication

□ Push-Based Active Duplication (11g~)

- ソース DB が イメージ・コピー をターゲット DB に転送
- 圧縮、暗号化、マルチセクションは非対応

実行例

```
CONNECT TARGET "rman/rman@source as SYSBACKUP"  
CONNECT AUXILIARY "sys/sys@target as SYSDBA"  
  
run {  
  ALLOCATE CHANNEL c1 DEVICE TYPE disk;  
  ALLOCATE CHANNEL c2 DEVICE TYPE disk;  
  ALLOCATE AUXILIARY CHANNEL c3 DEVICE TYPE disk;  
  DUPLICATE TARGET DATABASE TO target  
  FROM ACTIVE DATABASE;  
}
```



アクティブなデータベースの複製

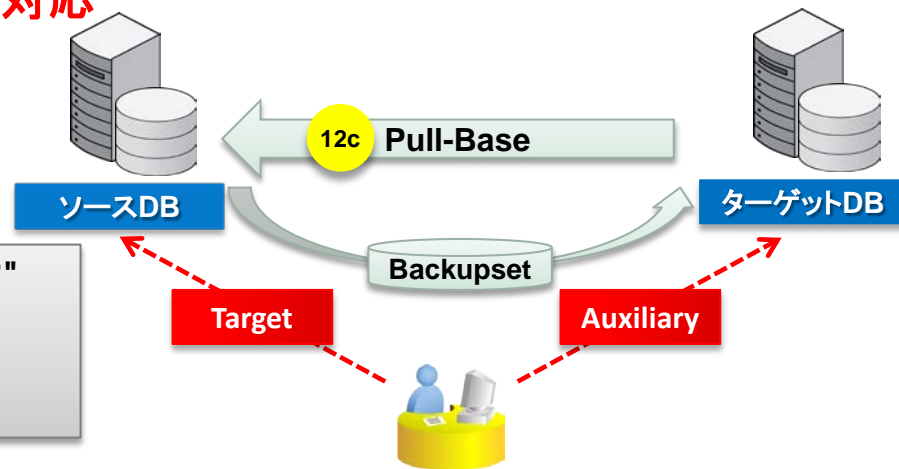
Pull-Based Active Duplication

□ Pull-Based Active Duplication (12c~)

- バックアップ・セットをターゲットDBに転送
- ターゲット DB がソース DB に接続しファイルを取得
- 圧縮、暗号化、マルチセクションに対応
- 未使用ブロックの圧縮が可能
- 12c の推奨手順

実行例

```
CONNECT TARGET "rman/rman@source as SYSBACKUP"  
CONNECT AUXILIARY "sys/sys@target as SYSDBA"  
  
DUPLICATE TARGET DATABASE TO target  
FROM ACTIVE DATABASE;
```



アクティブなデータベースの複製

Push-Base と Pull-Base

□ Pull-Base が使用されるケース

➤ アロケートしたチャンネル数が以下の場合

"Targetのチャンネル数" <= "Auxiliaryのチャンネル数"

※ 自動割り当ての場合は Pull-Base

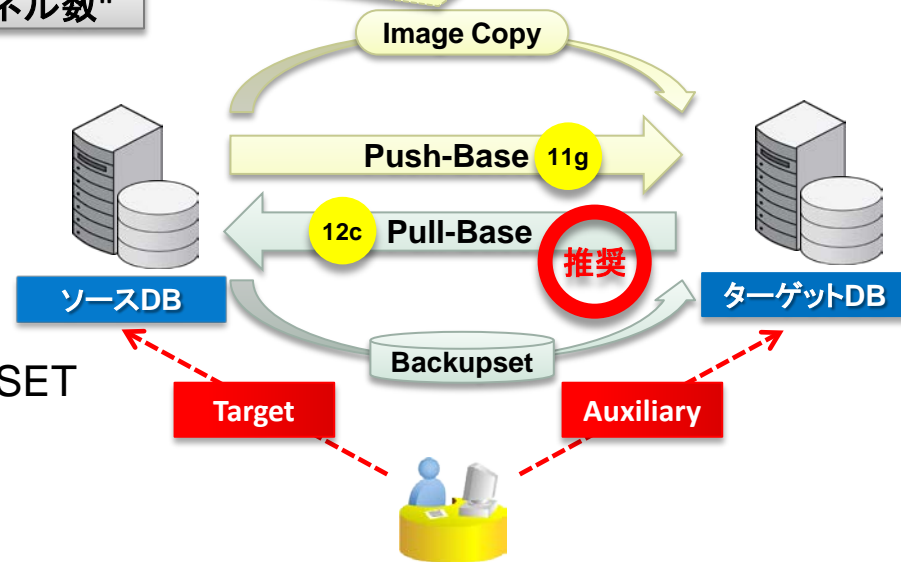
➤ 以下が指定されている場合 (*)

- SECTION SIZE
- USING BACKUPSET
- USING COMPRESSED BACKUPSET

◆ Pull-Base の使用を推奨

以下をすべて満たす場合のみ

- チャンネルを明示的に割り当てている
- "Targetのチャンネル数" > "Auxiliaryのチャンネル数"
- (*) のオプションが未指定



アクティブなデータベースの複製

Push-Base と Pull-Base

□ Pull-Base と Push-Base の比較

	Push-Base	Pull-Base
対象リリース	11g, 12c	12c
バックアップ形式	イメージコピー	バックアップセット
圧縮	×	●
暗号化	×	●
マルチセクション	×	●

4. クロス・プラットフォーム

"異なるプラットフォーム間での
リストア・リカバリ機能の強化"

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

ORACLE[®]

クロス・プラットフォームでのリストア・リカバリ

バックアップ・セットを用いたクロス・プラットフォームのデータ転送

□ プラットフォームを跨ぐデータベース、表領域の移動

➤ 移動方法

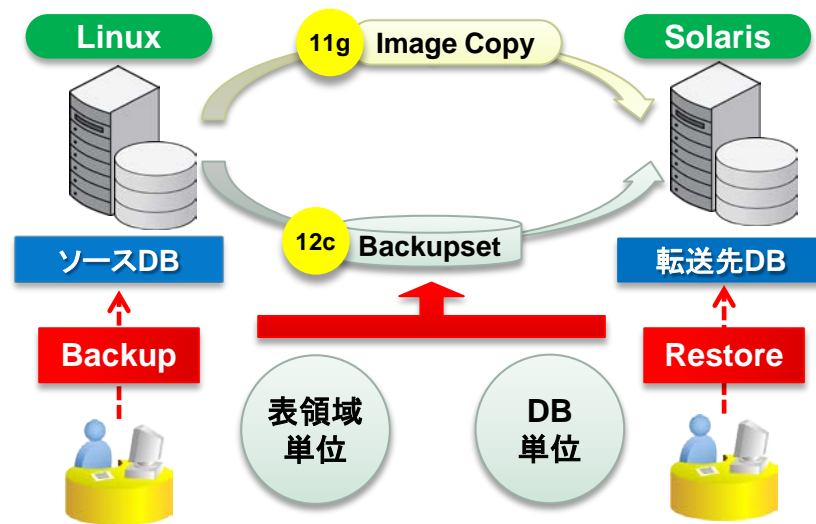
イメージ・コピー (11g~)

- ソース DB のイメージ・コピーを転送先 DB でリストア

バックアップ・セット (12c~)

- ソース DB のバックアップ・セットを転送先 DB でリストア

- 圧縮バックアップが可能
- 未使用ブロックの圧縮が可能



クロス・プラットフォームでのリストア・リカバリ

プラットフォームを跨ぐ場合の考慮点

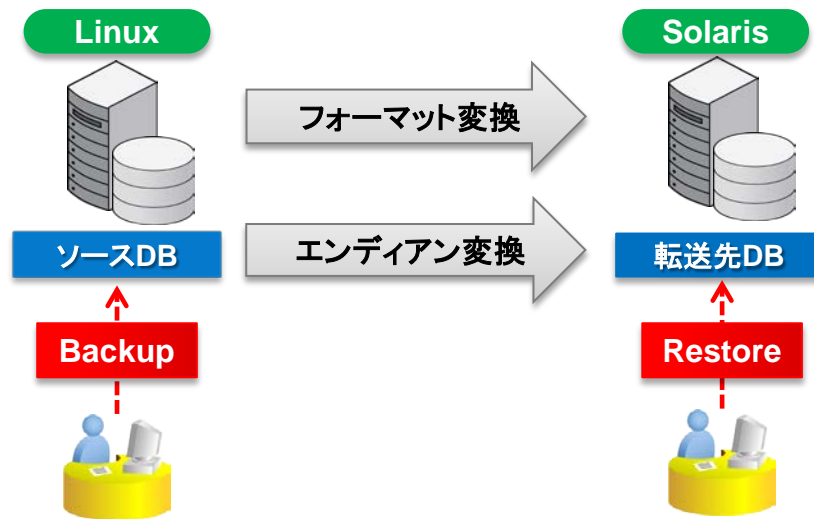
□ プラットフォームを跨ぐ場合の考慮点

プラットフォームへの対応

- プラットフォームに対応したフォーマット変換

エンディアンの変換

- エンディアンが異なる場合 (Big と Little) に変換が必要
- データベース単位では不可
- 表領域単位では可



クロス・プラットフォームでのリストア・リカバリ

プラットフォームの違いへの対応：2つの方法

ソース DB での対応

□ バックアップ時に対応

- TO PLATFORM 句を使用
- エンディアン変換もソース DB で実行
- バックアップ時に転送先のプラットフォームを指定

□ コマンド例

```
BACKUP  
TO PLATFORM 'Linux x86 64-bit'  
FORMAT '/tmp/backup.bk'  
DATABASE;
```

転送先 DB での対応

□ リストア時に対応

- FROM PLATFORM 句を使用
- エンディアン変換も転送先 DB で実行
- リストア時にソースのプラットフォームを指定

□ コマンド例

```
RESTORE  
FROM PLATFORM 'Linux x86 64-bit'  
FOREIGN DATABASE TO NEW  
FROM BACKUPSET '/tmp/backup.bk';
```

- 変換処理の負荷をどちらで負担するか
- どちらの処理時間の短縮を優先するか

クロス・プラットフォームでのリストア・リカバリ

プラットフォームの違いへの対応：プラットフォームの情報

□ 対応するプラットフォームの確認

- V\$TRANSPORTABLE_PLATFORM ビューより確認
- TO / FROM PLATFORM 句で PLATFORM_NAME 列の値を指定

```
SQL> SELECT PLATFORM_ID, PLATFORM_NAME, ENDIAN_FORMAT  
FROM V$TRANSPORTABLE_PLATFORM;
```

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
2	Solaris[tm] OE (64-bit)	Big
4	HP-UX IA (64-bit)	Little
6	AIX-Based Systems (64-bit)	Big
12	Microsoft Windows x86 64-bit	Little
13	Linux x86 64-bit	Little
17	Solaris Operating System (x86-64)	Little

```
BACKUP  
TO PLATFORM 'Solaris[tm] OE (64-bit)'  
FORMAT '/tmp/backup.bk'  
DATABASE;
```

```
RESTORE  
FROM PLATFORM 'Linux x86 64-bit'  
FOREIGN DATABASE TO NEW  
FROM BACKUPSET '/tmp/backup.bk';
```

※ 一部の結果を抜粋したものです

クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるバックアップ

- バックアップの取得単位

1. データベース単位

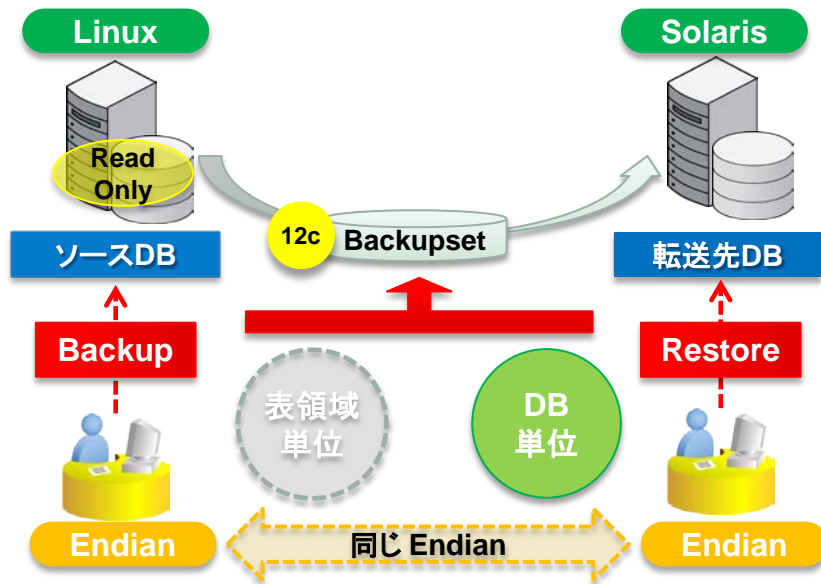
- データベースを一括してバックアップ
- データベース全体を READ ONLY に変更
- 同一エンディアンのみ対応

2. 表領域単位

- 表領域毎にバックアップを取得
- 対象の表領域を READ ONLY に変更
- **エンディアン変換が可能**

クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるデータベース単位のバックアップ・リストア



DB全体のバックアップを取得し
転送先にリストア

◆ 要件

- ソース、転送先が同一エンディアン
- バックアップ時にはソースDBをREAD ONLYでOPENする
- 使用できないオプション有り (*)

(*) 後のスライドを参照

ORACLE

クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるデータベース単位のバックアップ・リストア(1)

□ 変換を**転送先**で実行する場合の手順(1/2)

➤ (例) データベースの移行

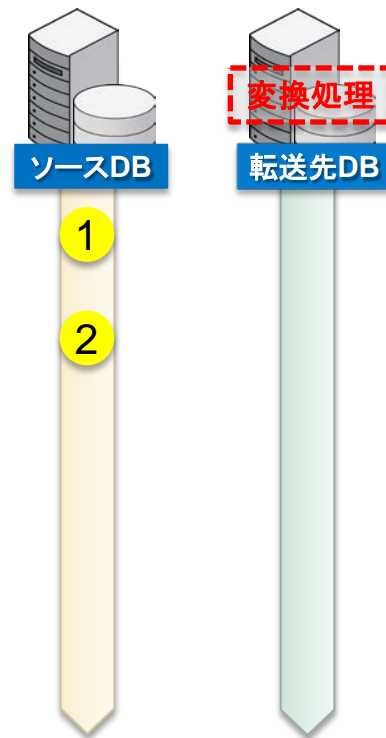
1 DB を READ ONLY で OPEN

```
SHUTDOWN IMMEDIATE  
STARTUP OPEN READ ONLY;
```

2 バックアップを取得

```
BACKUP  
FOR TRANSPORT  
FORMAT '/tmp/backup.bk'  
DATABASE;
```

バックアップのファイル名



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるデータベース単位のバックアップ・リストア(1)

□ 変換を**転送先**で実行する場合の手順(2/2)

➤ (例) データベースの移行

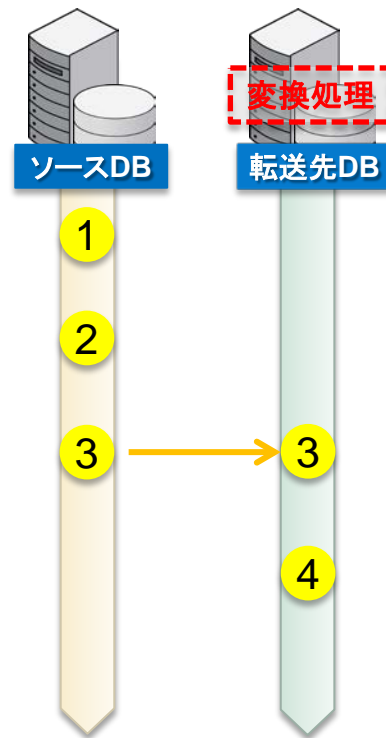
- 3 バックアップセットを転送先にコピー
 - FTP, SCP 等で物理ファイルをコピー
 - ファイル名は FORMAT 句で指定したもの
 - /tmp/backup.bk

4 バックアップをリストア

ソースのプラットフォーム名

```
RESTORE
FROM PLATFORM 'Solaris Operating System (x86-64)'
FOREIGN DATABASE TO NEW
FROM BACKUPSET '/tmp/backup.bk';
```

手順3でコピーしたファイル



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるデータベース単位のバックアップ・リストア(2)

□ 変換をソースで実行する場合の手順(1/2)

➤ (例) データベースの移行

1 DB を READ ONLY で OPEN

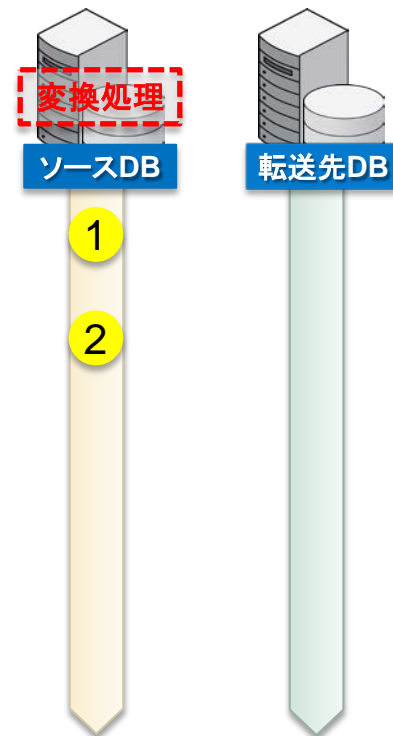
```
SHUTDOWN IMMEDIATE  
STARTUP OPEN READ ONLY;
```

2 バックアップを取得

```
BACKUP  
TO PLATFORM 'Linux x86 64-bit'  
FORMAT '/tmp/backup.bk'  
DATABASE;
```

転送先のプラットフォーム名

バックアップのファイル名



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるデータベース単位のバックアップ・リストア(2)

□ 変換をソースで実行する場合の手順(2/2)

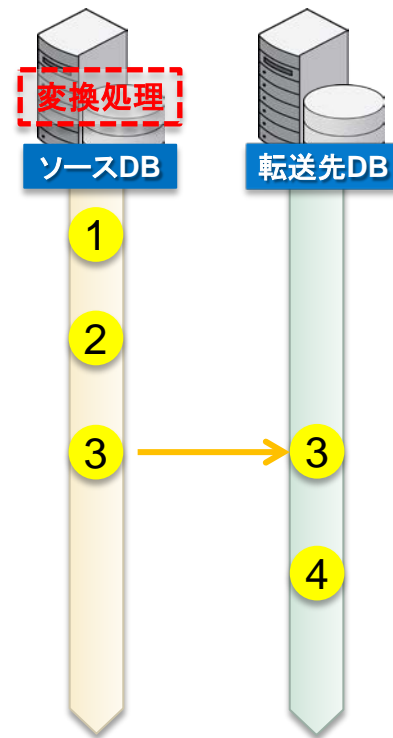
➤ (例) データベースの移行

- 3 バックアップセットを転送先にコピー
 - FTP, SCP 等で物理ファイルをコピー
 - ファイル名は FORMAT 句で指定したもの
 - /tmp/backup.bk

- 4 バックアップをリストア

```
RESTORE  
FOREIGN DATABASE TO NEW  
FROM BACKUPSET '/tmp/backup.bk';
```

手順3でコピーしたファイル



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるバックアップ

- バックアップの取得単位

1. データベース単位

- データベースを一括してバックアップ
- データベース全体を READ ONLY に変更
- 同一エンディアンのみ対応

2. 表領域単位

- 表領域毎にバックアップを取得
- 対象の表領域を READ ONLY に変更
- エンディアン変換が可能

2.1. 一貫性

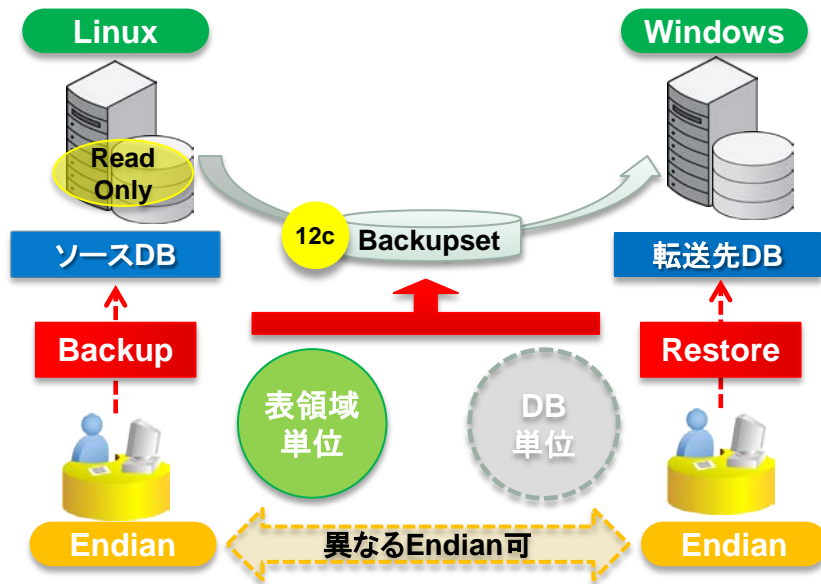
- 表領域の全体バックアップを使用
- READ ONLY はバックアップ取得完了まで

2.2. 非一貫性

- 差分増分バックアップを使用
- READ ONLY は最後のバックアップ取得時のみ

クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア



◆ 要件

- エンディアンの変換可能
- 一貫性・非一貫性のいずれかの方法でバックアップ取得が可能
- Data Pump を使用 (内部的に)
- 使用できないオプション有り (*)

表領域毎のバックアップを取得し
転送先にリストア

(*) 後のスライドを参照

ORACLE

クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるバックアップ

- バックアップの取得単位

1. データベース単位

- データベースを一括してバックアップ
- データベース全体を READ ONLY に変更
- 同一エンディアンのみ対応

2. 表領域単位

- 表領域毎にバックアップを取得
- 対象の表領域を READ ONLY に変更
- エンディアン変換が可能

2.1. 一貫性

- 表領域の全体バックアップを使用
- READ ONLY はバックアップ取得完了まで

2.2. 非一貫性

- 差分増分バックアップを使用
- READ ONLY は最後のバックアップ取得時のみ

クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：一貫性(2)

□ 変換をソースで実行する場合の手順(1/2)

➤ (例) TESTTBS 表領域の移行

1 表領域を READ ONLY に変更

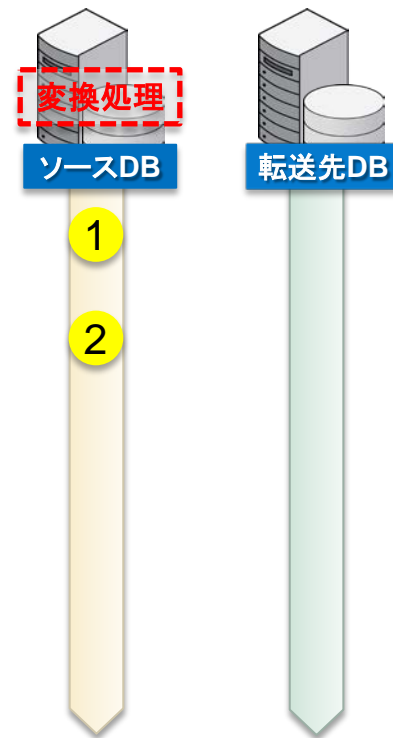
```
ALTER TABLESPACE testtbs READ ONLY;
```

2 バックアップを取得、DUMPFILe を作成

```
BACKUP  
TO PLATFORM 'Linux x86 64-bit'  
FORMAT '/tmp/backup.bk'  
DATAPUMP FORMAT '/tmp/dumpfile.bk'  
TABLESPACE testtbs;
```

転送先のプラットフォーム名

DUMPFILe のファイル名



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：一貫性(2)

□ 変換をソースで実行する場合の手順(2/2)

➤ (例)TESTTBS 表領域の移行

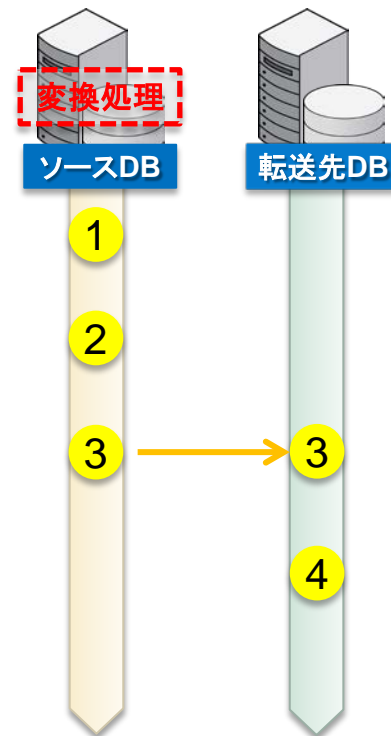
3 バックアップセット、DUMPFIL を転送先にコピー

- FTP, SCP 等で物理ファイルをコピー
- ファイル名は FORMAT 句で指定したもの
➤ /tmp/backup.bk、/tmp/dumpfile.bk

4 バックアップをリストア

```
RESTORE
FOREIGN TABLESPACE testtbs TO NEW
FROM BACKUPSET '/tmp/backup.bk'
DUMP FILE FROM BACKUPSET '/tmp/dumpfile.bk';
```

DUMPFIL のファイル名



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによるバックアップ

- バックアップの取得単位

1. データベース単位

- データベースを一括してバックアップ
- データベース全体を READ ONLY に変更
- 同一エンディアンのみ対応

2. 表領域単位

- 表領域毎にバックアップを取得
- 対象の表領域を READ ONLY に変更
- エンディアン変換が可能

2.1. 一貫性

- 表領域の全体バックアップを使用
- READ ONLY はバックアップ取得完了まで

2.2. 非一貫性

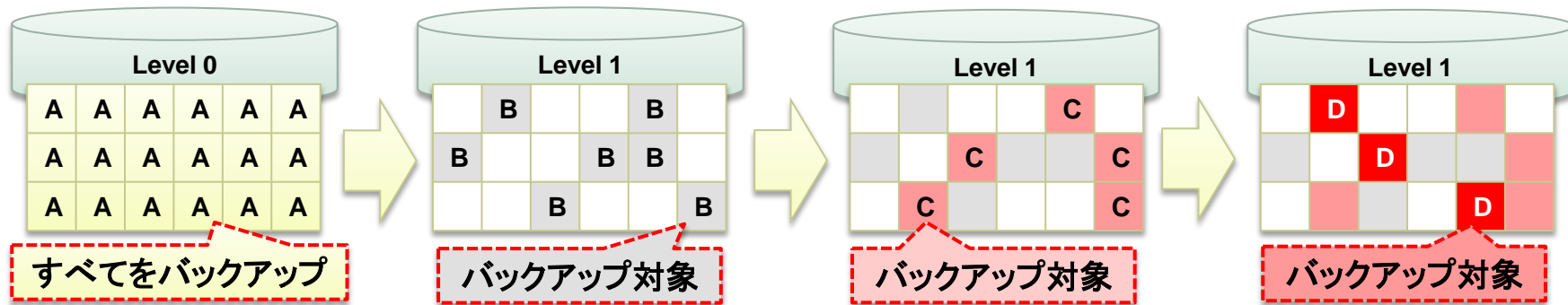
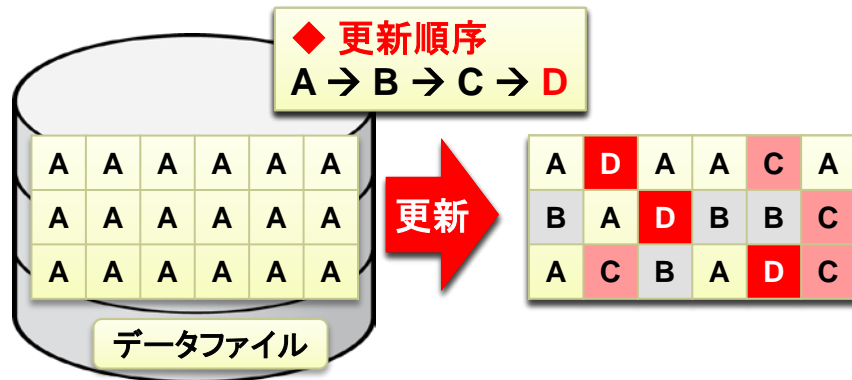
- 差分増分バックアップを使用
- READ ONLY は最後のバックアップ取得時のみ

クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性

□ 差分増分バックアップの使用

- Level 0 のバックアップを取得
- Level 1のバックアップ(差分)を取得
… 繰り返し …
- Level 1のバックアップ(差分)を取得



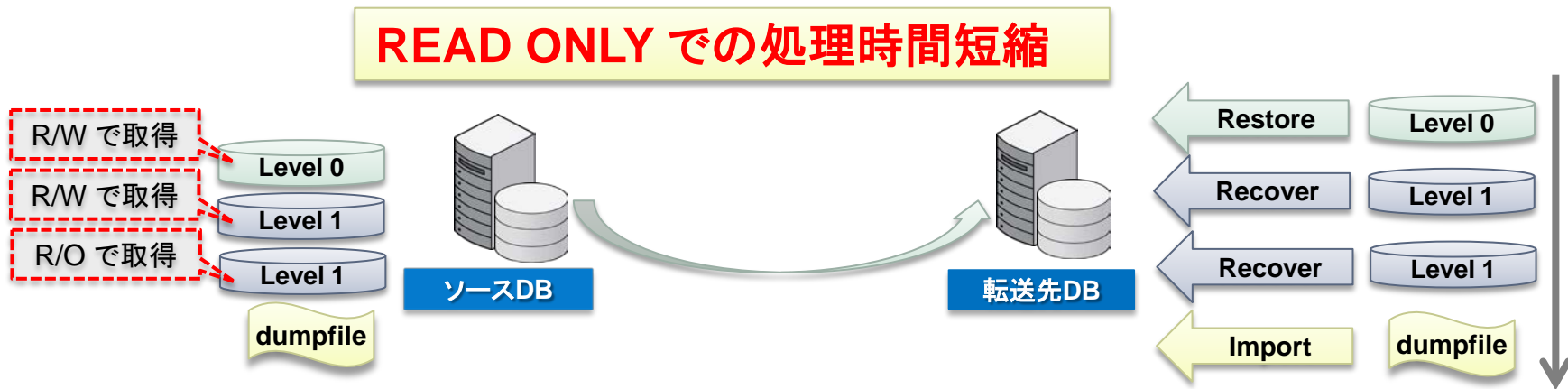
クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性

□ 非一貫性バックアップの利用

➤ 差分増分バックアップを使用

- ベースのバックアップ及びその差分のバックアップは READ WRITE で取得
- 最後の差分のバックアップのみ READ ONLY で取得



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性

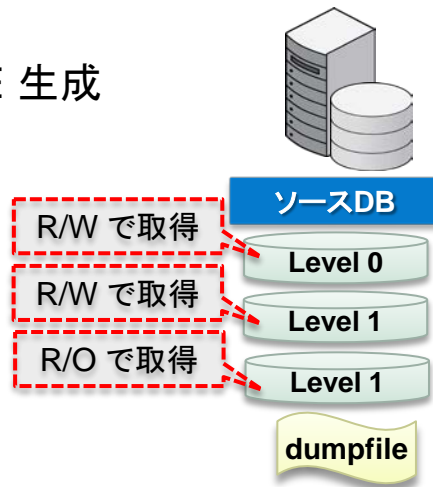
□ 非一貫性バックアップの利用

➤ ソースDB

1. ベースのバックアップを READ WRITE で取得
2. 更新ブロックのバックアップを READ WRITE で取得
3. 最後のバックアップを READ ONLY で取得 + DUMPFIL 生成

➤ 転送先DB

4. ベースのバックアップをリストア
5. 更新ブロックのリカバリ
6. 最後のバックアップでのリカバリ
7. DUMPFIL の IMPORT



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性

□ 非一貫性バックアップ使用時の考慮点

- バックアップのリストア・リカバリだけではなく、Data Pump による IMPORT の実行が必要
 - IMPORT する DUMPFILE は、ソース DB 側で RMAN より取得
- バックアップ時に ALLOW INCONSISTENT 句の指定が必要

クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

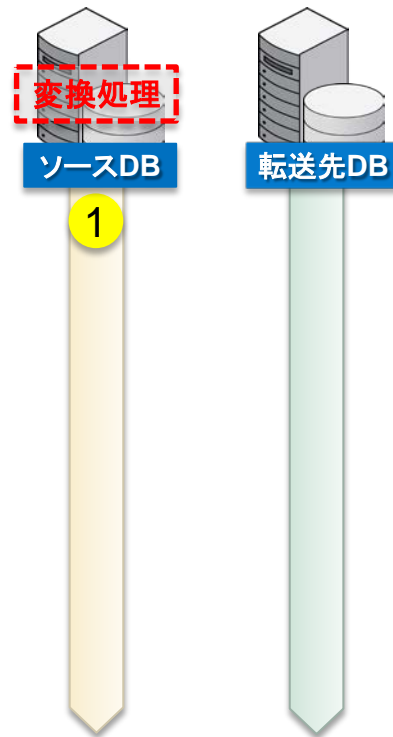
□ 変換をソースで実行する場合の手順(1/9)

➤ (例) TESTTBS 表領域の移行

1 ベースのバックアップ(Level 0)を取得

```
BACKUP  
TO PLATFORM 'Linux x86 64-bit'  
ALLOW INCONSISTENT  
INCREMENTAL LEVEL 0  
TABLESPACE testtbs  
FORMAT '/tmp/backup0.bk';
```

バックアップのファイル名



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

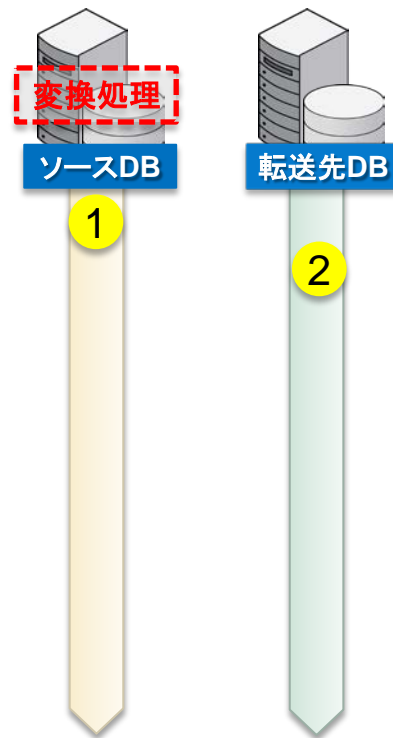
□ 変換をソースで実行する場合の手順(2/9)

➤ (例) TESTTBS 表領域の移行

2 取得したベースのバックアップ(Level 0)のリストア

```
RESTORE  
FOREIGN TABLESPACE testtbs  
FORMAT '/home/oradata/remote/testtbs.dbf'  
FROM BACKUPSET '/tmp/backup0.bk';
```

手順1で取得したバックアップ



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

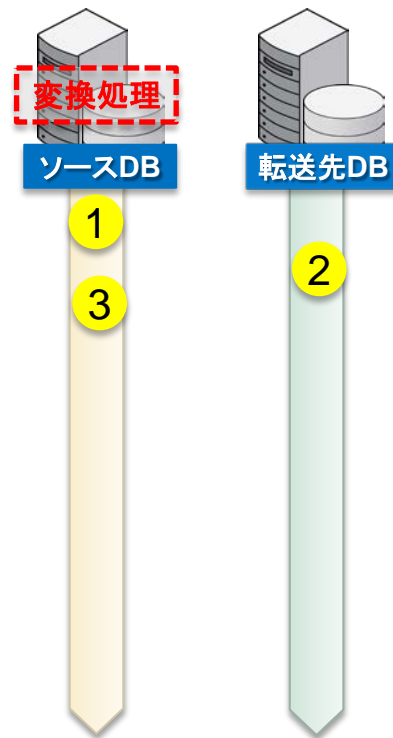
□ 変換をソースで実行する場合の手順(3/9)

➤ (例) TESTTBS 表領域の移行

3 差分のバックアップ(Level 1)の取得

```
BACKUP  
TO PLATFORM 'Linux x86 64-bit'  
ALLOW INCONSISTENT  
INCREMENTAL LEVEL 1  
TABLESPACE testtbs  
FORMAT '/tmp/backup1.bk';
```

バックアップのファイル名



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

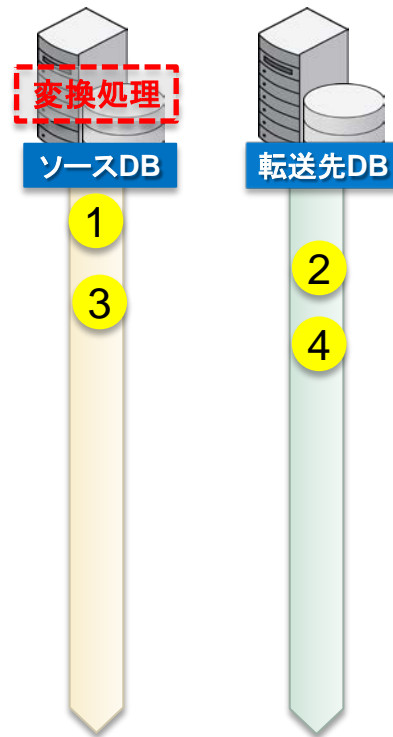
□ 変換をソースで実行する場合の手順(4/9)

➤ (例) TESTTBS 表領域の移行

4 取得した差分バックアップ(Level 1)のリストア・リカバリ

```
RECOVER  
FOREIGN DATAFILECOPY '/home/oradata/remote/testtbs.dbf'  
FROM BACKUPSET '/tmp/backup1.bk';
```

手順3で取得したバックアップ



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

□ 変換をソースで実行する場合の手順(5/9)

➤ (例) TESTTBS 表領域の移行

5 表領域を READ ONLY に変更

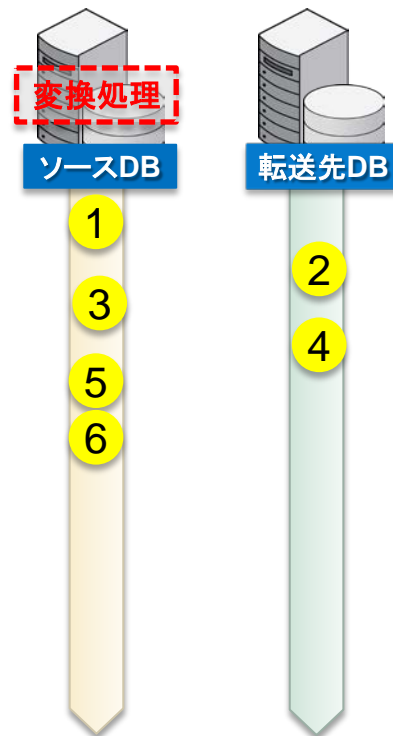
```
ALTER TABLESPACE testtbs READ ONLY;
```

6 一貫性バックアップ(Level 1)と DUMPFILE の取得

```
BACKUP  
TO PLATFORM 'Linux x86_64-bit'  
INCREMENTAL LEVEL 1  
TABLESPACE testtbs  
FORMAT '/tmp/backup2.bk'  
DATAPUMP FORMAT '/tmp/backup_dp.bk';
```

バックアップのファイル名

DUMPFILE のバックアップファイル名



クロス・プラットフォームでのリストア・リカバリ

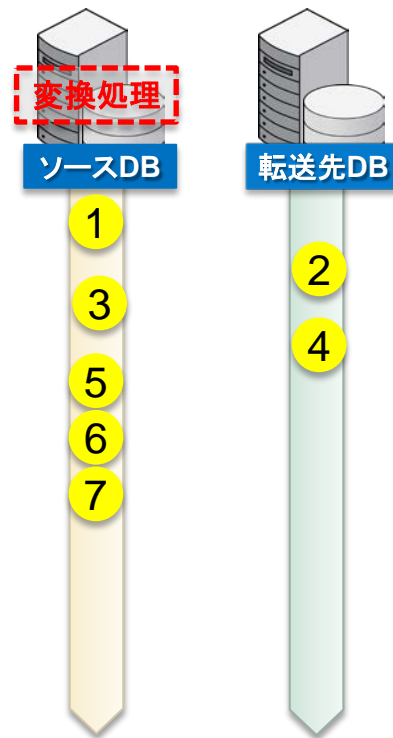
バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

□ 変換をソースで実行する場合の手順(6/9)

➤ (例) TESTTBS 表領域の移行

7 表領域を READ WRITE に変更

```
ALTER TABLESPACE testtbs READ WRITE;
```



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

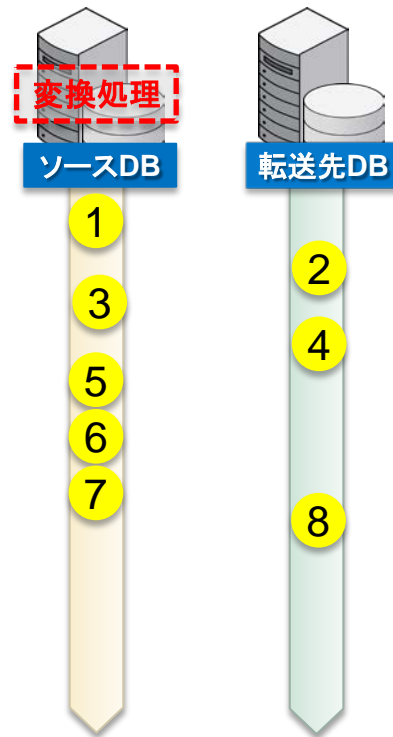
□ 変換をソースで実行する場合の手順(7/9)

➤ (例) TESTTBS 表領域の移行

8 取得した一貫性バックアップ(Level 1)のリストア・リカバリ

```
RECOVER  
FOREIGN DATAFILECOPY '/home/oradata/remote/testtbs.dbf'  
FROM BACKUPSET '/tmp/backup2.bk';
```

手順6で取得したバックアップ



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

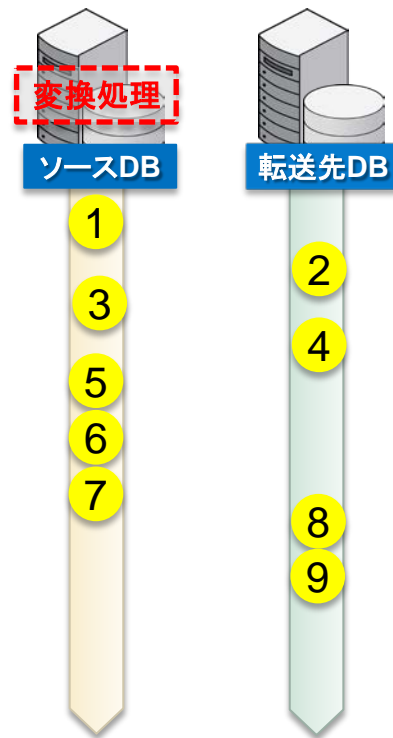
□ 変換をソースで実行する場合の手順(8/9)

➤ (例) TESTTBS 表領域の移行

9 DUMPFIL FILE のリストア

```
RESTORE
DUMP FILE 'datapump.dmp'
DATAPUMP DESTINATION '/tmp'
FROM BACKUPSET '/tmp/backup_dp.bk';
```

手順6で取得した DUMPFIL FILE のバックアップ



クロス・プラットフォームでのリストア・リカバリ

バックアップセットによる表領域単位のバックアップ・リストア：非一貫性(1)

□ 変換をソースで実行する場合の手順(9/9)

➤ (例) TESTTBS 表領域の移行

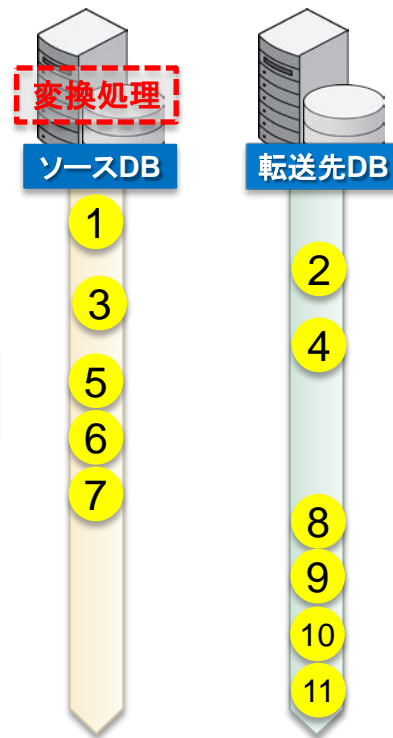
10 Data Pump での IMPORT

```
CREATE OR REPLACE DIRECTORY rman_dir as '/tmp';
```

```
$ impdp '/as sysdba' directory=rman_dir dumpfile=datapump.dmp -  
transport_datafiles='/home/oradata/remote/testtbs.dbf'
```

11 表領域を READ WRITE に変更

```
ALTER TABLESPACE testtbs READ WRITE;
```



クロス・プラットフォームでのリストア・リカバリ

バックアップ時に使用できないオプション

オプション名	概要
COMULATIVE	累積増分バックアップで使用
forRecoveryOfSpec	増分更新バックアップで使用
keepOption	バックアップの保存方針の指定
notBackedUpSpec	バックアップ未取得のファイルだけを対象とするバックアップ
PROXY	プロキシ・コピー機能の使用
SECTION SIZE	マルチセクション・バックアップの使用
TAG	バックアップセットへのタグ名の指定
VALIDATE	ファイルの検査

※ 詳細はマニュアル「Oracle Databaseバックアップおよびリカバリ・リファレンス 12cリリース1(12.1)」を参照

クロス・プラットフォームでのリストア・リカバリ

データベース単位と表領域単位

□ バックアップセットによる方式の比較

	データベース単位	表領域単位
バックアップ取得単位	データベースの一括バックアップ	表領域毎のバックアップ
エンディアン変換	×	●
非一貫性での取得	×	●
READ ONLY の時間	データベースのバックアップ取得中	一貫性：表領域のバックアップ取得中 非一貫性：最後の差分増分バックアップ取得中
ステップ数	少ない	多い
圧縮	圧縮バックアップ可 未使用ブロックの圧縮可	圧縮バックアップ可 未使用ブロックの圧縮可

5. SQLインタフェース

"RMAN からの SQL 文の実行"

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

ORACLE[®]

SQLインターフェース

RMANからのSQL文の実行

□ SQLインターフェースの改善

➤ 'SQL'の接頭語なしでSQL文の実行が可能に

11g RMAN> sql 'alter tablespace users read only';



12c RMAN> alter tablespace users read only;

➤ DESC コマンドの実行も可能に

12c RMAN> desc scott.emp

ALTER	ANALYZE	AUDIT
ASSOCIATE STATISTICS	CALL	COMMENT
COMMIT	CREATE	DELETE
DISASSOCIATE STATISTICS	DROP	EXPLAIN PLAN
INSERT	FLASHBACK	GRANT
LOCK	MERGE	NOAUDIT
PURGE	RENAME	REVOKE
ROLLBACK	SAVEPOINT	SELECT
SET	TRUNCATE	UPDATE

SQLインターフェース

注意点

□ 注意点

➤ SELECT 結果の表示フォーマットの変更は限定的

- 「SET」での出力変更は「NUMWIDTH」「LONG」のみ



```
RMAN> SET LONG 100
```



```
RMAN> SET NUMWIDTH 20
```



```
RMAN> SET LINESIZE 200
```

```
RMAN> select * from scott.emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450
7839	KING	PRESIDENT		1981-11-17 00:00:00	5000
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	950
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300

12行が選択されました

Hardware and Software

ORACLE®

Engineered to Work Together

ORACLE®

ORACLE®