

※注意※本資料は2015年2月20日に実施したセミナー資料の公開版です。一部内容は今後更新される可能性がありますので、最新情報はOracle Technical Networkに掲載する資料をご参照下さい。

Oracle GoldenGate

テクニカルTips～運用・トラブルシュート編～

日本オラクル株式会社

免責事項

以下の事項は、弊社の一般的な製品の方角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は、米国オラクル・コーポレーション及びその子会社、関連会社の米国及びその他の国における登録商標または商標です。他社名又は製品名は、それぞれ各社の商標である場合があります。

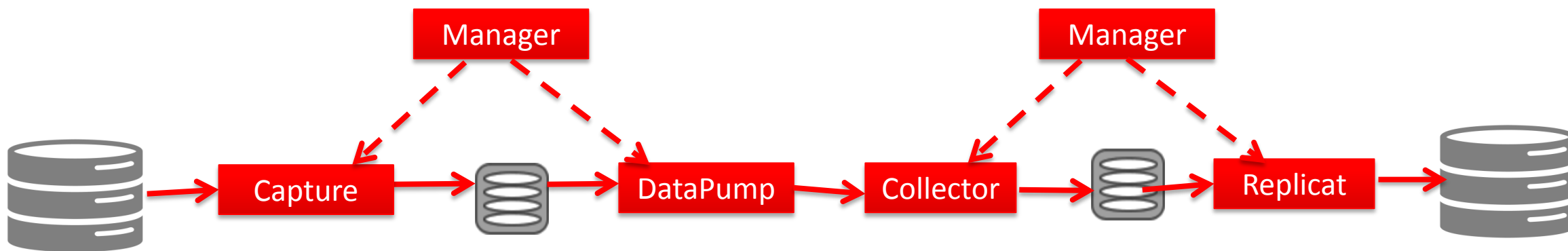
- 監視
- トラブルシュート

GoldenGateの監視

- 死活監視
- エラー監視
- 性能監視
- 整合性確認

死活監視

- GoldenGateはManagerプロセスがExtract/Replicatを監視する
 - 各サーバーのManagerを監視することで全体を監視可能
- DBサーバ自体に監視のポリシーがある場合は、継承されるケースもある
 - GoldenGateの各プロセスをps コマンドで監視など



エラー監視

- ggserr.log
 - GoldenGate操作やメッセージが出力されるログ
 - Oracle DB の alert.log に近いイメージ

```
2012-11-05 17:38:39 INFO OGG-00992 Oracle GoldenGate Capture for Oracle, cap01.prm: EXTRACT
CAP01 starting.
2012-11-05 17:38:39 INFO OGG-03035 Oracle GoldenGate Capture for Oracle, cap01.prm:
Operating system character set identified as US-ASCII. Locale: en_US_POSIX, LC_ALL:.
2012-11-05 17:38:39 ERROR OGG-00664 Oracle GoldenGate Capture for Oracle, cap01.prm: OCI
Error beginning session (status = 28001-ORA-28001: ).
2012-11-05 17:38:39 ERROR OGG-01668 Oracle GoldenGate Capture for Oracle, cap01.prm: PROCESS
ABENDING.
2012-11-05 17:39:34 INFO OGG-00987 Oracle GoldenGate Command Interpreter for Oracle: GGSCI
command (oracle): start cap01.
2012-11-05 17:39:34 INFO OGG-00963 Oracle GoldenGate Manager for Oracle, mgr.prm: Command
received from GGSCI on host red.jp.oracle.com (START EXTRACT CAP01 ).
```

- 
- INFO / WARNING / ERROR でメッセージが区分される
 - エラー監視で使用可能

エラー監視

- レポートファイル

- プロセスごとに出力されるログファイル
- プロセスごとのより詳細な挙動やエラー情報を表示
- GoldenGateインストールディレクトリ下の dirrpt ディレクトリに出力
- プロセス障害発生時にはまず参照するファイル
- 下記コマンドからも参照可能

GGSCI> view report <プロセス名>

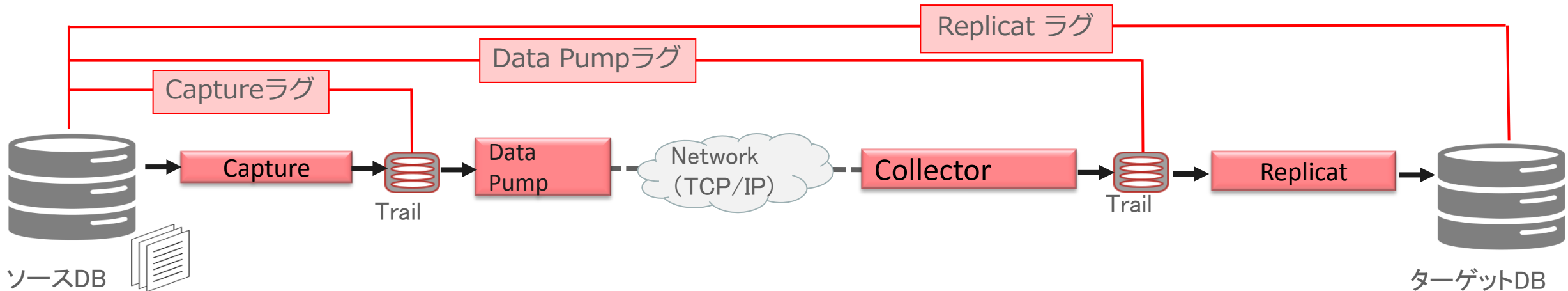
ggserr.log だけでなく、レポートファイルも確認する

性能監視

- プロセス毎のラグ
- Heartbeat Table
- 処理量

ラグ (lagコマンドで取得)

- 最後に処理したトランザクションのコミット時間と処理完了時までの時間の差
 - どのプロセスの処理が遅れているか？
- 各プロセスが自分の処理を完了した時間
 - Extract(Capture/DataPump)であればTrailファイルの出力まで
 - Replicatであればターゲット側でのデータの適用まで



ラグ情報の蓄積

- Management Pack for Oracle GoldenGateの使用
 - オリジナルのWeb UI よりラグの履歴をグラフで確認可能
 - Oracle GoldenGate Monitor
 - Enterprise Manager Cloud Control の画面よりグラフで確認可能
 - Enterprise Manager Oracle GoldenGate System Monitoring Plug-In
- 定期的な lagコマンド実行をファイルに出力
 - Shell スクリプトへの組み込み
- Managerパラメータ設定より、レポート/ggserr.log に出力
 - LAGREPORT{MINUTS | HOURS}
 - LAGINFO{SECONDS | MINUTS | HOURS}
 - LAGCRITICAL{SECONDS | MINUTES | HOURS}

lagコマンドの注意点

- lagコマンドは各プロセス内で計算可能なラグ情報を算出
 - (プロセスが最新トランザクションを処理した時間) – (最新トランザクションがソースDBでコミットされた時間)
 - 新しいレコードを読みこまなければラグは更新されない
 - 前段のプロセスが停止
 - 業務処理が停止
- ソースDBとターゲットDBの時間差は常に測定できない
 - Heartbeat Table の使用を検討

参考:[GG] LAG コマンドを使用した LAG 情報の取得方法(KROWN:149705) (Doc ID 1753561.1)

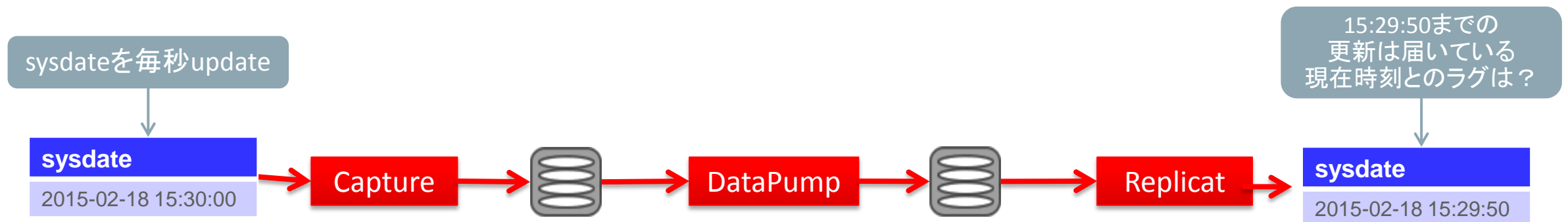
Heartbeat Table

- 原理

- ダミー表を作り、定期的にsysdateをupdate
- ダミー表をGoldenGateでレプリケーション
- ターゲット側でダミー表を参照し「いつ時点までのデータが届いているか」を確認

- GoldenGateの管理情報を含むHeartbeat Table の作成・管理も可能

- 参考: Oracle GoldenGate Best Practices: Heartbeat Table for Monitoring Lag times (Doc ID 1299679.1)



処理量

- statsコマンドによる取得
 - 各プロセスに対して実行し、表毎のSQL処理量を取得

コマンド

```
GGSCI> stats {extract | replicat} <プロセス名>
```

出力結果例(抜粋)

```
Extracting from SCOTT.TEST to SCOTT.TEST:
```

```
*** Total statistics since 2011-08-16 16:02:56 ***
```

Total inserts	100.00
Total updates	20.00
Total deletes	17.00
Total discards	0.00
Total operations	137.00

- REPORTCOUNTパラメータに取得

- 各プロセスのパラメータとして指定し、指定の間隔で、全体のSQL処理量をレポートファイルに記録

設定例

```
REPORTCOUNT EVERY 10 SECONDS, RATE
```

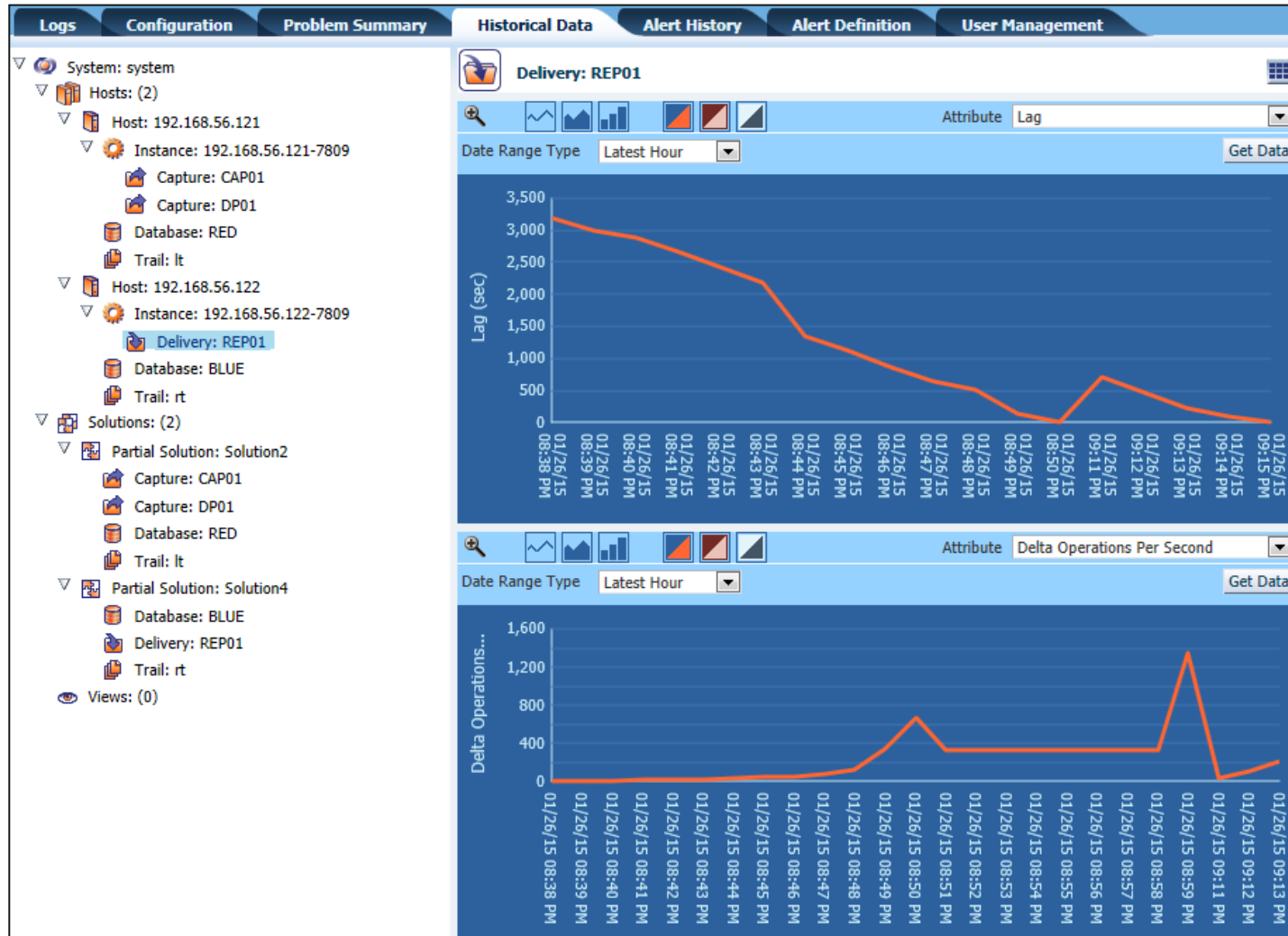
出力結果例(抜粋)

```
1029260 records processed as of 2011-09-19 17:22:02 (rate 102053,delta 102053)
1946631 records processed as of 2011-09-19 17:22:12 (rate 96917,delta 91737)
3089452 records processed as of 2011-09-19 17:22:22 (rate 102688,delta 114281)
4000338 records processed as of 2011-09-19 17:22:35 (rate 92206,delta 68493)
```

処理量の蓄積

- Management Pack for Oracle GoldenGateの使用
 - オリジナルのWeb UI より処理量の推移をグラフで確認可能
 - Oracle GoldenGate Monitor
 - Enterprise Manager Cloud Control の画面よりグラフで確認可能
 - Enterprise Manager Oracle GoldenGate System Monitoring Plug-In
- 定期的な statsコマンド実行をファイルに出力
 - Shell スクリプトへの組み込み
- 各プロセスで REPORTCOUNTパラメータを指定

(参考) Oracle GoldenGate Monitor



← ラグの推移

← 処理量の推移

(参考)スクリプトの作成

- OBEYスクリプト
 - GGSCIコマンドをスクリプトファイル化
 - GGSCIより OBEYコマンドで呼び出す
 - SQL*Plusの@のようなイメージ

作成例

```
$ cat lag_extract.oby
lag extract cap01
```

実行例

```
GGSCI> obey lag_extract.oby
```

```
GGSCI> lag extract cap01
```

```
Sending GETLAG request to EXTRACT CAP01 ...
Last record lag: 2 seconds.
At EOF, no more records to process.
```

• シェルスクリプト

- OSシェルスクリプトからGGSCIコマンドを呼び出す

作成例(定期的にOBEYスクリプトを実行)

```
$ cat lag_extract.sh
#!/bin/sh
OGG_DIR=/goldengate
cd ${OGG_DIR}

(
SHELL_DIR=/home/oracle/shell
SLEEP_COUNT=10
LOOP_COUNT=0
LOOP_MAX=10

while [ $LOOP_COUNT -lt $LOOP_MAX ]
do
echo "OBEY ${SHELL_DIR}/lag_extract.oby"
sleep ${SLEEP_COUNT}
LOOP_COUNT=`expr $LOOP_COUNT + 1`
done
) | ./ggsci
```


整合性確認

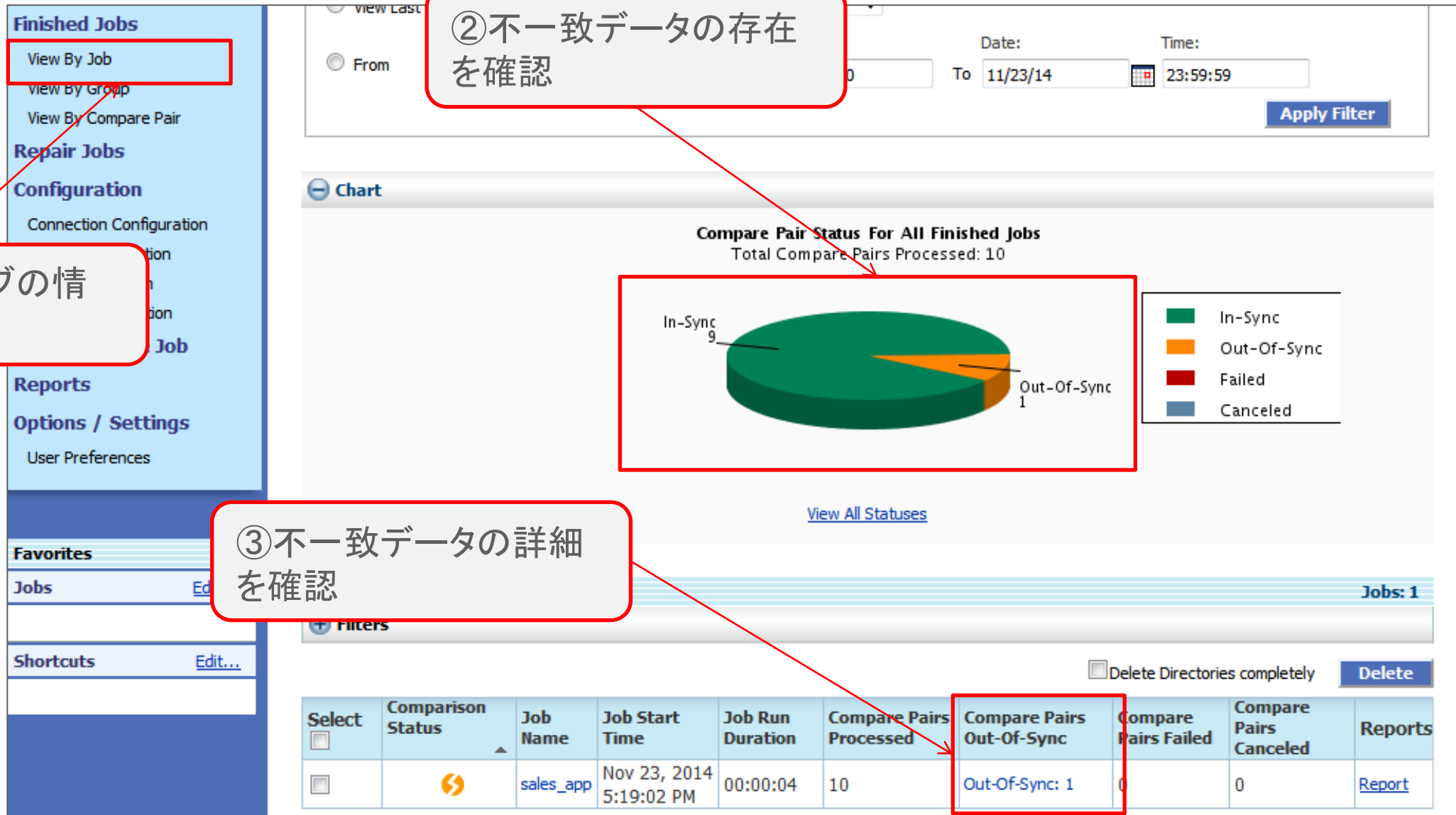
- GoldenGateによるレプリケーション時、ソースDB/ターゲットDBは常にOpen状態
- オペミス/接続ミスなどによる意図しない更新によるデータ不整合
GoldenGateではハンドリング不可能
- Oracle GoldenGate Veridataを使用して、
データ伝播対象表の中身のレベルで確認
 - どの表のどのデータが不一致かをレポート可能

(参考)Oracle GoldenGate Veridata

① 終了したジョブの情報
を表示


② 不一致データの存在
を確認


③ 不一致データの詳細
を確認




(参考)Veriataによる不一致データの修復

1 Repair Groups

☒  ☒ sales_app Compare Run Start Time: Nov 23, 2014 5:19:02 PM, Duration: 00:00:04(Compare pairs - Initial Out-Of-Sync: 1, Repaired: 0, Repair failed: 0)


☒  ☒ sales_app Compare Run Start Time: Nov 23, 2014 5:19:02 PM, Duration: 00:00:04(Compare pairs - Initial Out-Of-Sync: 1, Repaired: 0, Repair failed: 0)

☒  ☒ DEMO_SAMPLE=DEMO_SAMPLE Compare Run Start Time: Nov 23, 2014 5:19:04 PM, Duration: 00:00:01(Rows compared: 1, Initial Out-Of-Sync: 1, Repaired: 0, Repair failed: 0)

	Rows	Inserts	Updates	Deletes	Bytes
Initial Out-Of-Sync	1	0	0	1	
Repaired	0	0	0	0	
Failed Repair	0	0	0	0	
Not Repaired	1	0	0	1	

Out-Of-Sync Rows Details

Any Repair Status Any Operation All Columns **Apply Filter**

Select	Status	Operation	*COL 1 (NUMBER)	COL 2 (VARCHAR2)
<input checked="" type="checkbox"/>		Delete	1	demo

⏮ ⏪ ⏩ ⏭

Run Repair

① クリックして展開

② 不一致レコードの詳細を確認

③ 修復を実行

- 監視
- **トラブルシューティング**
 - 遅延
 - 異常停止
 - その他

遅延

- どのプロセスが遅延しているのかを特定する
- 確認ステップ例
 1. 業務テーブルまたはHeartbeat Tableからレプリケーションの遅れを確認
 - ターゲットDBに最新データが届いていない
 2. 各プロセスのラグを確認
 - どのプロセスが遅れているのか？
 3. 処理量の推移を確認
 - 処理量が増えている？

遅延ケース(1): Replicatスループットの限界

- 傾向

- Replicatのラグが大きい
- ReplicatプロセスのCPU使用率が高い(100%とは限らない)
- AWRなど見てもターゲットDBのボトルネックは見られない(ほとんどDB CPU待ちなど)

- 対処方法

- BATCHSQLが未設定であれば試す
- Replicatの多重化検討
 - Replicatごとの担当テーブルを割り振り
 - Trailファイルからテーブル毎の更新を分析し、分割方法を検討
 - RANGE関数の使用して、Replicatごとの担当をレコードで分割
 - Integrated Replicat への構成変更

Oracle GoldenGate - Using the GoldenGate Logdump Utility to Manually Load Balance across Multiple Processes (Doc ID 1301300.1)

遅延ケース(2) : ターゲット表でFull Scan 多発

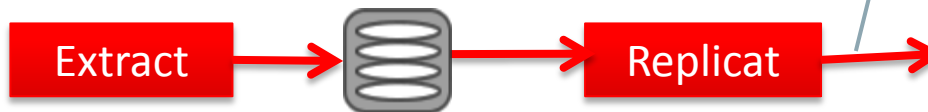
- 傾向

- Replicatのラグが大きい
- 特定の表への update / delete 処理負荷・時間がかかっている

- 原因

- 主キーの無いソース表への一括 update または delete 分
 - 例: UPDATE TAB1 SET COL3='Finished';

COL1	COL2	COL3
1	A	On going ⇒ Finished
2	B	On going ⇒ Finished
3	C	On going ⇒ Finished



ターゲットは1件ずつ更新
索引が効かないため
都度Full Scan

COL1	COL2	COL3
1	A	On going ⇒ Finished
2	B	On going ⇒ Finished
3	C	On going ⇒ Finished

- 対策

- ターゲット表に(一時的でも)索引を作成し 論理キー(KEYCOL)として指定

遅延ケース(3) : ネットワーク転送が遅い

- 傾向
 - DataPumpのラグが大きい
- 原因
 - 帯域不足
 - Round Trip Time(RTT)が大きい(遠隔地転送など)ことによるスループット低下
- 対処方法
 - 帯域不足に対しては、圧縮転送機能の使用を検討
 - RMTHOST <hostname>, MGRPORT <port number>, COMPRESS
 - RTTの大きい環境に対してはTCPソケットバッファサイズを大きくする
 - 例: 平均RTT 80ms (0.08 sec) 帯域幅 100Mbpsの場合
 $0.08 \times 100 \div 8$ (バイト換算) = 1MB = 1000000
RMTHOST <hostname>, MGRPORT <port number>, TCPBUFSIZE 1000000

遅延ケース(4):NASへのTrailファイルの書き込みが遅い

- 傾向

- Data Pump または Capture のラグが大きい

- 原因

- NFSマウントオプション(共通)

- Oracle GoldenGate Best Practice: NFS Mount options for use with GoldenGate (Doc ID 1232303.1)
- Linuxの場合: rw,bg,hard,nointr,rsize=32768,wsiz=32768,tcp,**actimeo=0,noac**,vers=3,timeo=600 (actimeo=0, noac が低速化の原因になりやすい傾向)

- I/Oサイズ(Data Pumpのみ)

- TCPバッファサイズに依存してCollectorからのI/Oサイズが小さくなり、NFSマウントオプションの影響を受けやすい

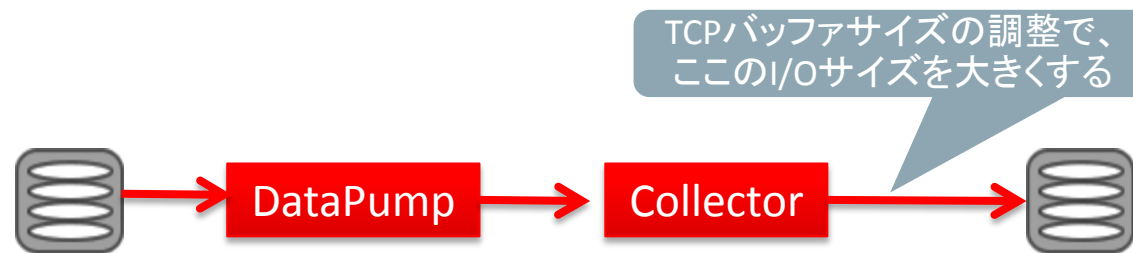
- 対処方法

- 出力先の変更

- TCPバッファサイズ増加によってI/Oサイズを大きくする

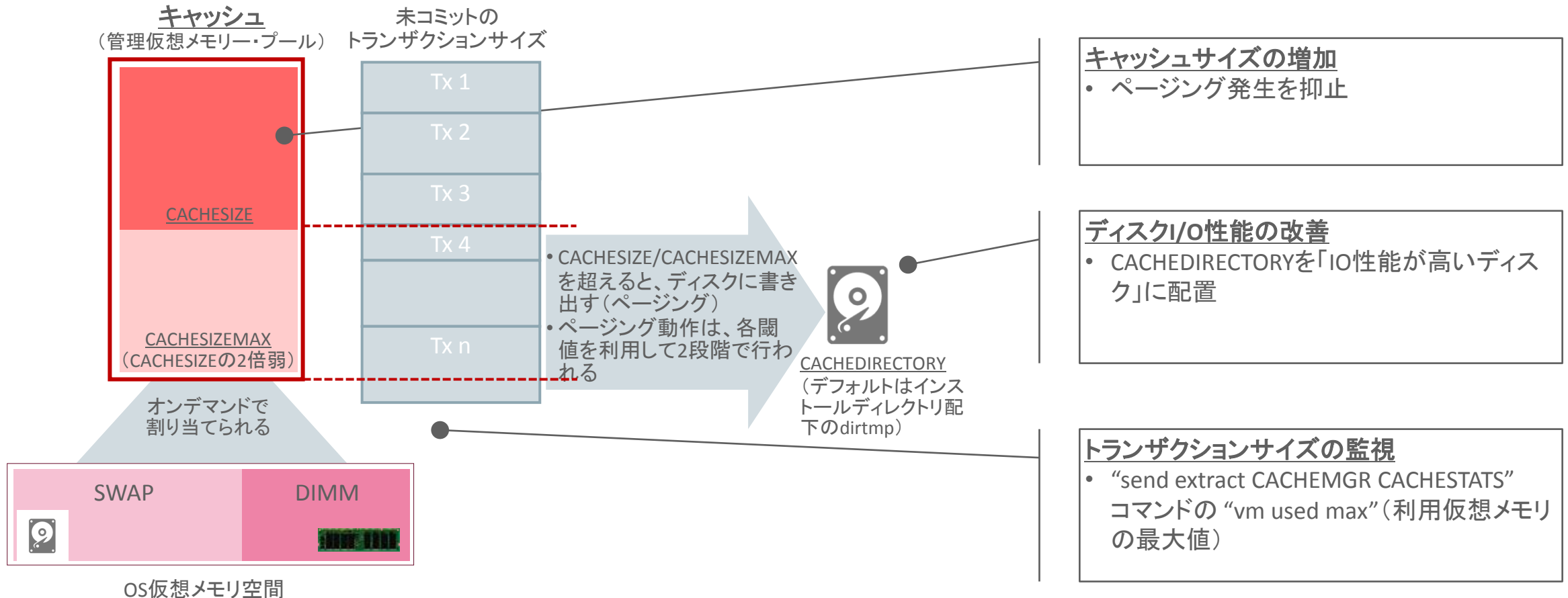
- 前スライドと同様の設定方法

- actimeo=0, noac を外す対応は、性能向上に有効な可能性があるが、障害リスクが高まるため **推奨されない方法**。
問題が発生した場合、マウントオプションを元に戻すガイドをすることが考えられます。



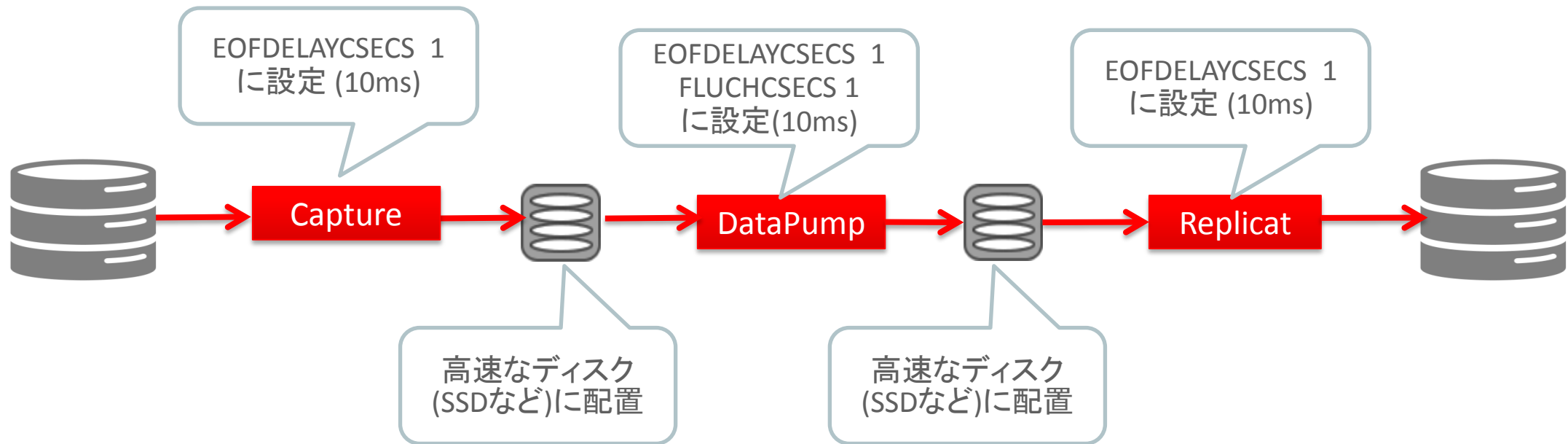
遅延ケース(5):ラージトランザクションのキャプチャ

- CaptureがラージトランザクションをTrailファイルに書き出すまでに時間がかかる



遅延ケース(6): ラグの極小化

- 各プロセスの処理間隔を最小にする
- Trailファイルを高速なディスクに配置
- その他のボトルネックが無いか確認



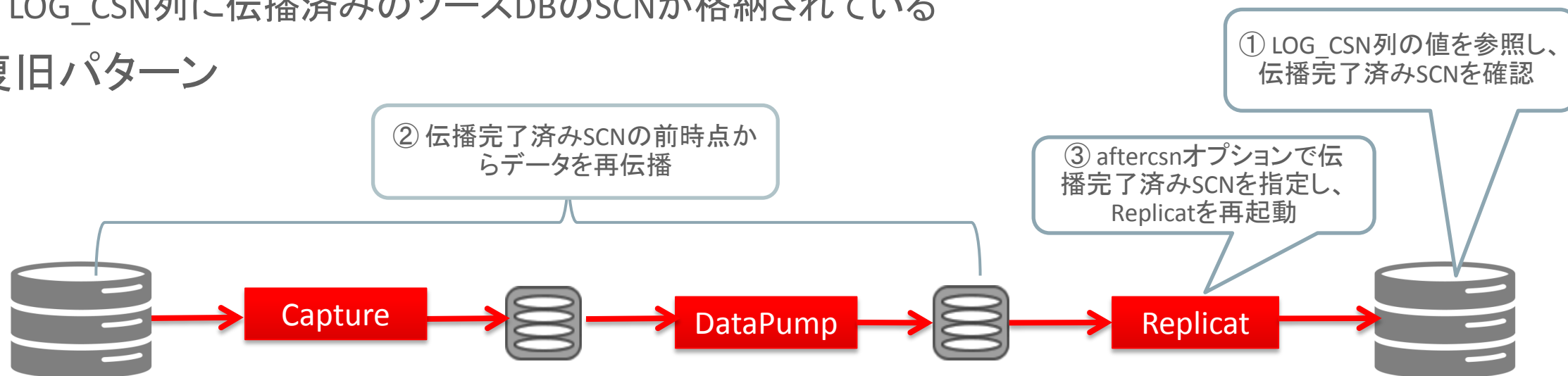
異常停止

障害の種類	関連コンポーネント	復旧方法
プロセス障害	Extract (Capture)	Managerによる自動再起動
	Extract (Data Pump) / Collector	Managerによる自動再起動
	Replicat	Managerによる自動再起動
	Manager	手動再起動 (Oracle Clusterwareにより自動化)
ファイル障害 (オペミス / OS, Disk障害 による破損等)	Trail ファイル / Checkpointファイル	過去時点からのTrailファイル再生成し、再同期 または、初期ロードから再構成
ネットワーク障害	Extract (Data Pump) / Collector	ネットワーク復旧後にプロセスを再起動
サーバー障害	Extract (Capture), Extract (Data Pump) / Collector, Replicat , Manager	サーバー復旧後にプロセスを再起動 (Extract / ReplicatはManager起動時に自動起動可能)
データ同期障害 (オペミスによるターゲット表への誤った更新等)	Extract (Capture)	過去時点からのTrailファイル再生成し、再同期 または、初期ロードから再構成
	Extract (Data Pump) / Collector	
	Replicat	手動での修復

主な障害は自動復旧が可能

データ再同期による復旧

- プロセス再起動だけでは、レプリケーション再開できないケース
 - Trailファイルの破損：伝播するデータがなくなる
 - チェックポイントファイルの破損：処理開始ポイントが分からない
 - ターゲットDBのPoint-in-time Recovery：GoldenGateのチェックポイント情報とずれる
- ターゲットDBのチェックポイント・テーブルを使用
 - LOG_CSN列に伝播済みのソースDBのSCNが格納されている
- 復旧パターン



データ再同期による復旧

- GLOBALSファイルに以下のパラメータを入れることで、Replicatは常にチェックポイントテーブルを参照しながら起動

```
_csnfiltering on
```

- Replicat起動時にチェックポイントテーブルを参照するよう明示的に指定することも可能

```
GGSCI> start replicat <name> filterduptransactions
```

- これらの機能を使用することで、チェックポイントテーブルの参照が不要になる
- Integrated Replicatはチェックポイントテーブルは持たずに、内部的に適用状況を管理

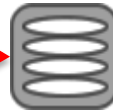
データ障害ケース(1):Not Nullのない 一意キー / 一意索引

- GoldenGateのadd trandataコマンドは、表にUKのみが存在する場合、UK列のみをサプリメンタル・ロギング対象に指定
- UK列にNot Nullがない場合、UK=null列が複数可能
- ソースのUK=nullの行が更新されると、ターゲットはUK=nullの行全てを更新します
- 回避策: Null含め一意となる列の組み合わせで、ソース側のサプリメンタル・ロギングを指定**

UPDATE TEST01 SET TYPE='US'
WHERE NAME='B' ;

ID(UK列)	NAME	TYPE
1	A	JP
	B	JP ⇒ US
3	C	JP
	D	JP

Extract



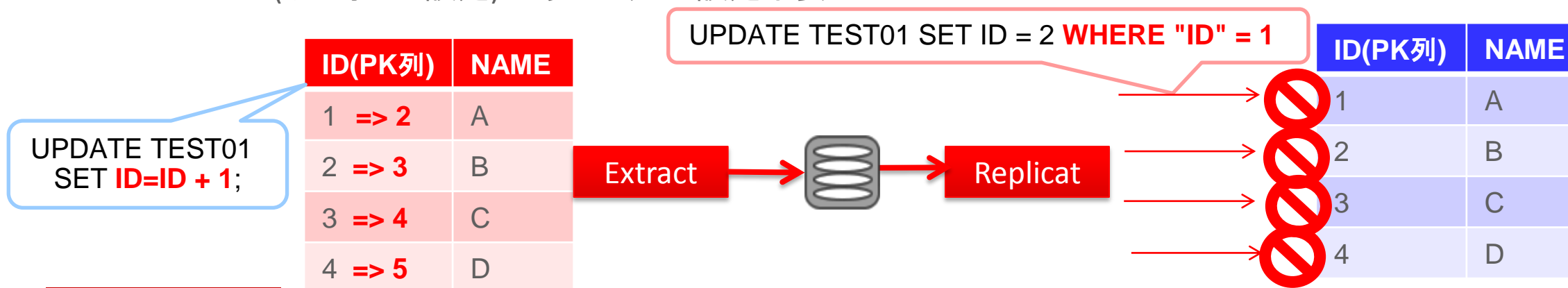
Replicat

UPDATE "TEST01 SET TYPE = 'JP' **WHERE**
"ID" is NULL

ID(UK列)	NAME	TYPE
1	A	JP
	B	JP ⇒ US
3	C	JP
	D	JP ⇒ US

データ障害ケース(2):主キー一括更新

- ソース側が単一のSQLに更新でも、ターゲット側は1レコードずつ処理
- $PK = PK + 1$ のように1レコード単位で実行するとエラーになる一括更新は注意が必要
- 対応策
 - ターゲットDBが11.2.0.2未満:ターゲット表の設定変更が必要。ターゲット表の 主キー制約設定 deferrable initially immediate の設定が必要。この設定はデフォルトではなく、設定変更には対象列の制約および索引の再作成が必要。これによりindexがuniqueでなくなるため、業務処理への影響確認が必要。また、ReplicatにHANDLETPKUPDATEパラメータの設定が必要。
 - ターゲットDBが11.2.0.2以降:ターゲット表の設定変更は基本的に不要。主キー制約設定 not deferrable initially immediate (デフォルト設定)であれば追加設定不要



データ障害ケース(3):初期ロード時のロングトランザクション

- 初期ロード後にcommitしたはずのトランザクションが伝播できていない？
 - Captureは構成後に開始したトランザクションのみをキャプチャ



Tx 1 : GoldenGateで伝搬

Tx 2 : 初期ロードで伝搬

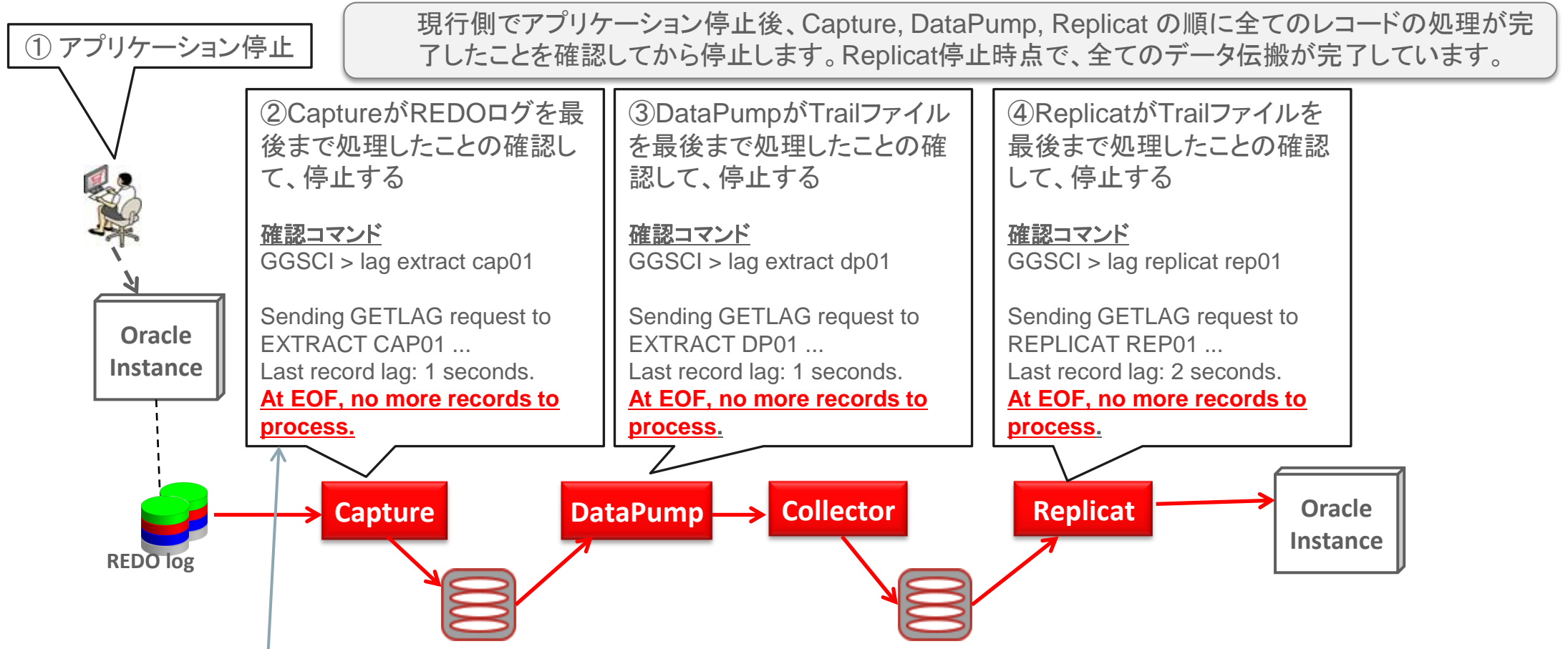
Tx 3 : GoldenGateでも初期ロードでも伝搬されない

- 対処方法
 - Trailファイル / アーカイブ・ログから分析
- 対策
 - Capture構成前にv\$transaction(gv\$transaction)等でオープントランザクションを確認

データ障害ケース(4)：未対応の処理

- 伝播漏れが発覚した場合は、GoldenGateが対応していない処理によるデータではないかを確認
 - Classic Captureで RAC環境へのXA
 - Classic CaptureでRAC環境へのパラレルDML
 - Nologging のオペレーション

その他Tips (1) : 移行先/DRサイトへの切り替え



注意: アプリケーションが停止しても、内部処理によりREDOが生成されEOFに到達しない場合があります、infoコマンドよりCaptureプロセスのチェックポイント情報を合わせて参照して、処理の完了を確認します。
参考: [GG] LAG コマンドを使用した LAG 情報の取得方法(KROWN:149705) (Doc ID 1753561.1)

その他Tips (2) : Integrated Capture / Replicat

- パフォーマンス・チューニング・障害解析に別途情報が必要
 - 独自のヘルスチェックレポート
 - Oracle DBにGoldenGate用のビューを実装(V\$GOLDENGATE_XXX, V\$GG_XXX など)
 - Oracle 12c では、AWRレポート内にGoldenGateの情報も表示
- 参考ドキュメント
 - GoldenGate Integrated Capture and Integrated Replicat Healthcheck Script (Doc ID 1448324.1)
 - Best Practices and Performance Tuning Recommendations for Oracle GoldenGate Integrated Replicat (Doc ID 1953623.1)
 - Oracle GoldenGate Performance Data Gathering (Doc ID 1488668.1)
 - GoldenGate Integration with AWR (Doc ID 1596493.1)

Hardware and Software Engineered to Work Together

ORACLE®