

# ディザスタ・リカバリ構成における バッチ処理適用の検討

Oracle Data Guard 運用ベストプラクティス

Creation Date: October 20, 2008

Version: 1.0

**ORACLE**<sup>®</sup>

**HITACHI**  
Inspire the Next

## 1. はじめに

昨今の企業における IT システムの重要性はますます高くなっています。たとえ万一の地震などの天災によるサイト障害やハードウェアに起因するようなシステム障害が発生してしまったとしても、企業は顧客情報などのビジネス上重要なデータを保護することと、迅速なシステム復旧による継続したサービスを提供することを求められています。

このような背景から、日本オラクル株式会社と株式会社 日立製作所は、Oracle GRID Centerでの活動を通じて、BCM(Business Continuity Management)をテーマとして、日立ハードウェアとOracle Database / Oracle Fusion Middlewareの組み合わせによるプラットフォームソリューションの技術検証を実施してきました。BCMの主要素の一つであるディザスタ・リカバリについては、日立の高信頼ブレードサーバー BladeSymphonyと、Oracle Database のディザスタ・リカバリ機能であるOracle Data Guardを組み合わせた災害対策環境上で、障害時を想定した動作検証や主要機能の検証を実施済みで、その成果はホワイトペーパーとして公開されています。<sup>1</sup>

今回の検証では、プラットフォームの機能検証に留まらず、Oracle Data Guard 稼動開始後の運用現場での課題解決を意識しました。具体的には、大量データの作成や更新を行うバッチ処理に注目し、Oracle Data Guard 環境におけるバッチ処理実行の運用上の注意事項や Oracle Database 11g の新機能である転送データ圧縮機能の有効性を確認しました。その検証結果を運用ベストプラクティスとしてご紹介いたします。

---

<sup>1</sup> 「日立と Oracle が実現する BCM プラットフォームソリューション & Oracle Active Data Guard 検証報告」

[http://www.oracle.co.jp/solutions/grid\\_center/hitachi/index.html](http://www.oracle.co.jp/solutions/grid_center/hitachi/index.html)

## 2. 目次

1. はじめに.....	2
2. 目次.....	3
3. Executive Summary.....	4
4. バッチ処理について.....	6
5. 検証目的.....	7
5.1 Oracle Data Guard環境におけるバッチ処理の考慮事項.....	7
5.2 REDO圧縮機能の効果.....	7
5.3 ネットワークチューニングの効果.....	7
6. 検証構成.....	9
6.1.1 使用ハードウェア.....	9
6.1.2 使用ソフトウェア.....	10
7. 検証シナリオと着眼点.....	11
7.1 処理の流れ.....	11
7.1.1 バッチ処理.....	11
7.1.2 OLTP処理.....	12
7.1.3 検索処理.....	12
7.2 REDO圧縮の設定.....	14
7.2.1 ギャップREDO圧縮.....	14
7.2.2 非同期REDO圧縮.....	14
7.3 ネットワークチューニングの実施.....	14
8. 検証結果.....	16
8.1 Oracle Data Guard環境におけるバッチ処理の考慮事項.....	16
8.2 非同期REDO圧縮の効果.....	20
8.3 バッチ処理時のREDO転送を停止する運用方法.....	23
8.4 ネットワークチューニングの効果.....	25
9. まとめ.....	29
10. 謝辞.....	31
Appendix. OLTP処理における非同期REDO圧縮の効果.....	32

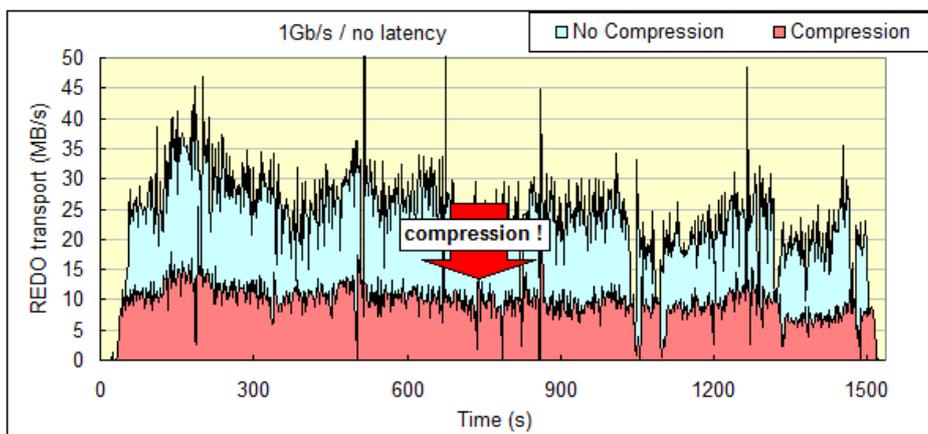
### 3. Executive Summary

本ホワイトペーパーでは、日立 BladeSymphony 上に構築された Oracle Real Application Clusters と Oracle Data Guard による災害対策環境で、大量データを作成するバッチ処理における注意点や効果的なチューニングについて検証した結果を報告します。

Oracle Data Guard では、プライマリ・データベースからスタンバイ・データベースへ REDO を転送することにより同期をとります。大量更新を行うバッチ処理では、REDO 生成量が転送ネットワーク帯域を上回る可能性があります。このような処理が長時間続く場合にまず課題となるのは、**REDO 転送が追いつかずにプライマリ - スタンバイ間の REDO のタイムラグ（転送ラグ）が大きくなる**ことです。このタイムラグは、プライマリ・データベースに障害が発生してスタンバイ・データベースへフェイルオーバーした時の消失データの表すため、**タイムラグが大きくなるとシステムで事前に定義されたリカバリ・ポイント目標(Recovery Point Objective : RPO)を満たせなくなる可能性があります。**

この課題を解決する最もシンプルな方法は、REDO 転送用に十分なネットワーク帯域を確保することです。しかし、災害対策で遠隔地にスタンバイ・データベースを配置する構成では、REDO 転送ネットワークに十分な帯域 (1Gb/s など) を確保することは多くの場合困難です。

このような場合に有効なのが、Oracle Database 11gのAdvanced Compression Optionによって提供される転送時にREDOを圧縮する機能です。グラフ 3-1は、同一のバッチ処理（データロード）中の、REDO圧縮設定の有無による転送量の比較です。**圧縮設定により、転送量が半分以下にまで減少する**ことがわかりました。

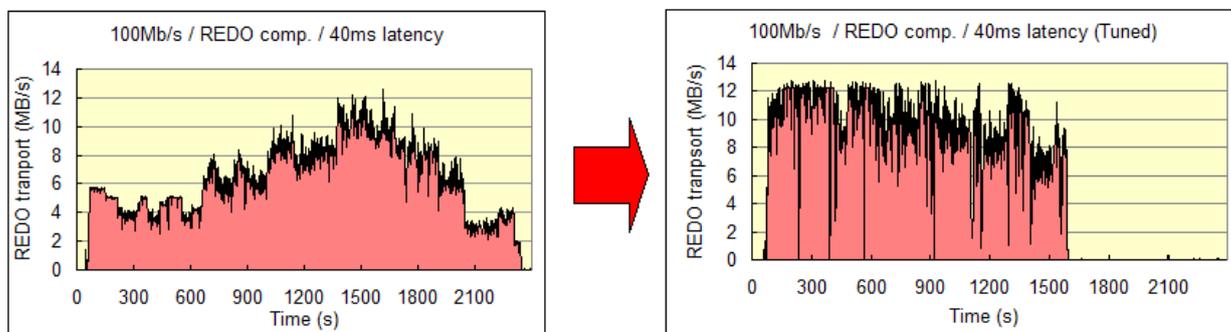


グラフ 3-1 圧縮設定の有無による REDO 転送量の比較

また今回の検証シナリオでは、**REDO圧縮によってプライマリ - スタンバイ間のタイムラグを最小限に抑え、データが適切に保護されることや、REDO圧縮設定によるバッチ処**

理性能へのオーバーヘッドはほとんどないことも確認できました。

スタンバイ・データベースを遠隔地に配置する場合は、ネットワーク遅延によるREDO転送効率の低下にも注意する必要があります。これは、転送効率が低下すると、プライマリ・データベースとスタンバイ・データベースのタイムラグが大きくなり、適切なデータ保護ができなくなる可能性があるためです。Oracle社では、ネットワーク遅延による転送効率低下に対して、Oracleのネットワーク関連パラメータを編集することによるチューニング方式をベストプラクティス<sup>2</sup>として公開しています。グラフ 3-2はバッチ処理（データロード）によって生成されたREDOの転送量の推移を示しています。左のグラフはチューニング適用前で右のグラフがチューニング適用後です。チューニングによって転送効率が大幅に改善していることがわかります。



グラフ 3-2 ネットワークチューニングによる効果

以上の結果より、日立 BladeSymphony 上に構築された Oracle Data Guard において、バッチ処理時の考慮事項について確認したうえで、REDO 圧縮機能とネットワーク・パラメータの編集によるチューニングの有効性を証明できました。

<sup>2</sup> Oracle Maximum Availability Architecture (MAA)

- Data Guard REDO 転送とネットワークのベスト・プラクティス -

<http://www.oracle.com/technology/global/jp/products/availability/htdocs/maa.html>

## 4. バッチ処理について

バッチ処理とは、一般的にコンピュータで一連のプログラム群（ジョブ）を順に実行していくような処理を指します。例えば給与計算や売上データ集計などの定期的に必要な処理をコンピュータリソースに空きがある夜間や週末などの時間にバッチ処理で行うなどの適用方法が考えられます。

バッチ処理の一般的な要件としては、以下のようなものが挙げられます。

1. 規定時間（バッチウィンドウ）内に全ての処理を終了する  
例：発注管理システム
  - バッチ処理時間：5時間(24:00 - 5:00)
  - 処理内容：売上計算 => 在庫計算 => 発注書作成
  - 通常業務開始時刻：6:00
2. エラー発生時の対応が確立されている  
例：バッチサーバーのクラスタリング、ジョブツールによるリトライ設定
3. 多数のジョブを順序良くスケジューリングできている  
例：ジョブツールによるスケジューリングと監視
4. マシン・リソースを効率的に使用できている

このように、バッチ処理はシステムに求められる業務要件を元に設計されます。バッチ処理におけるデータベースの役割に注目すると、バッチ処理では大量データの挿入 / 更新 / 集計 / 検索などを長時間に渡って行う処理と考えることができます。そのため、バッチ処理においてデータベースに求められる要件としては、バッチウィンドウ内に処理を完了するための性能や高可用性といったものが挙げられます。

## 5. 検証目的

本検証では、Oracle Data Guard 環境におけるバッチ処理に注目し、以下の項目を確認することを目的としました。

### 5.1 Oracle Data Guard環境におけるバッチ処理の考慮事項

Oracle Data Guard (DG) では、本番データベース (プライマリ・データベース) からスタンバイ・データベースに対して Oracle Database の更新ログである REDO を転送します。スタンバイ・データベースは受信した REDO を適用しながらプライマリ・データベースと同期をとります。

バッチ処理によってプライマリ・データベースに対して大量の更新が発生すると、大量の REDO が生成されるため、スタンバイ・データベースへの REDO 転送量も増加します。一方、災害対策を想定した Oracle Data Guard 環境では、スタンバイ・データベースを遠隔地に配置する機会が多いため、REDO 転送用に十分なネットワーク帯域を確保することがコスト面から困難なケースが考えられます。このような環境では、結果的にバッチ処理時にはネットワーク転送効率を上回る REDO が長時間生成され、データ保護やバッチ処理性能において問題となる可能性があります。本検証では、この「ネットワーク転送効率を上回る REDO を生成するバッチ処理」を発生させ、Oracle Data Guard 環境固有のバッチ処理への影響を確認しました。

### 5.2 REDO圧縮機能の効果

REDO 圧縮は Oracle Database 11g の Advanced Compression Option によって提供される機能です。REDO 圧縮を有効にすると、Oracle Data Guard の REDO 転送時に REDO を圧縮します。これによってネットワークの使用帯域を低減させることや、転送時間を短縮させることができます。一方で、圧縮処理による CPU リソースの消費も懸念されます。本検証では、大量の REDO が生成されるバッチ処理を使用して、REDO 圧縮の効果や処理性能への影響を確認しました。

### 5.3 ネットワークチューニングの効果

遠隔地に配置されたスタンバイ・データベースへの REDO 転送では、ネットワ

ーク帯域だけでなくネットワーク遅延によって Round Trip Time (RTT)が長くなることにも考慮する必要があります。これは、ネットワーク遅延によって転送効率が下がり、ネットワーク帯域を使いきれなくなるためです。本検証では、遠隔地への REDO 転送時に想定される遅延があるネットワーク環境において転送効率を向上させるためのチューニングについて検討し、その効果を確認しました。

## 6. 検証構成

図 6-1が検証システム構成です。負荷クライアントマシンからデータベースサーバーへの接続はパブリックネットワークを使用して行われます。パブリックネットワーク帯域は1Gb/s です。

スタンバイ・データベースはフィジカル・スタンバイを構成しています。プライマリ・データベースからスタンバイ・データベースへの REDO 転送用に独立したネットワークを構成しています。また、REDO 転送にはネットワークシミュレータを仲介させることにより、帯域や遅延を制御可能にしています。

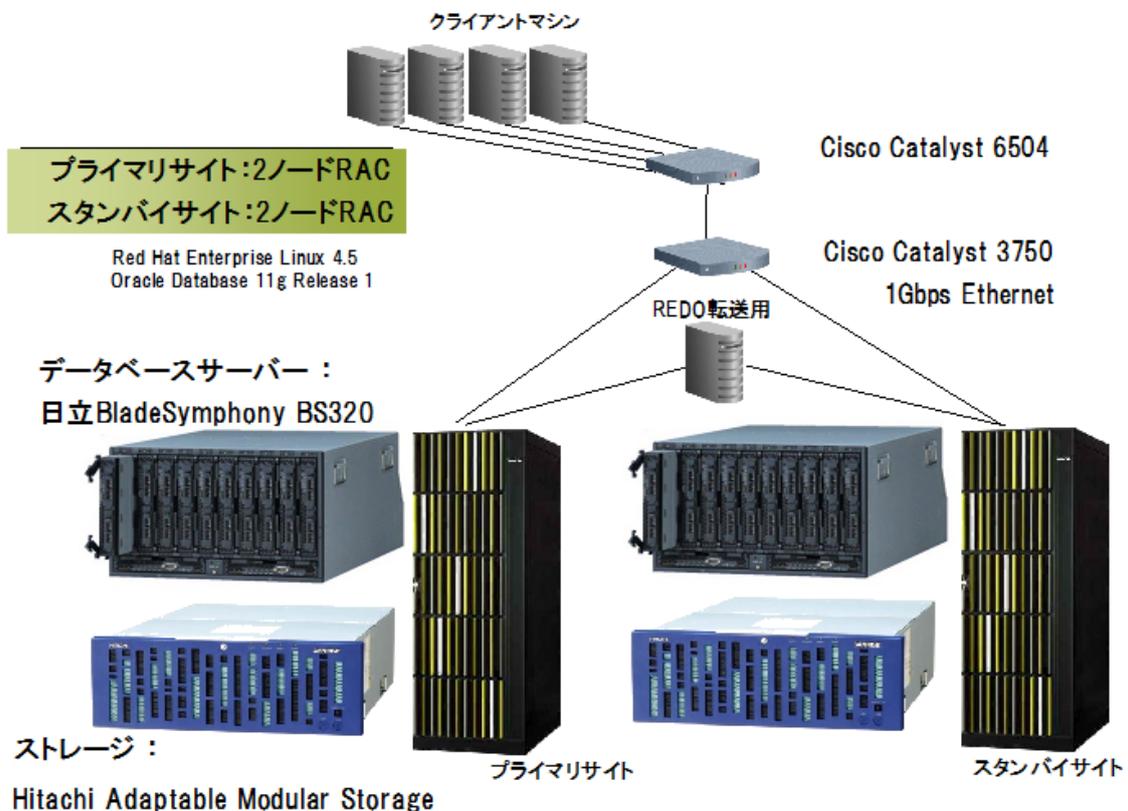


図 6-1 検証システム構成

### 6.1.1 使用ハードウェア

- データベースサーバー

機種	日立 BladeSymphony BS320 × 計4ブレード
CPU	デュアルコア インテル <sup>(R)</sup> Xeon <sup>(R)</sup> プロセッサ 3GHz 2ソケット/ブレード
メモリ	8GB

- クライアントマシン

機種	インテルホワイトボックス 計3台
CPU	デュアルコア インテル <sup>®</sup> Xeon <sup>®</sup> プロセッサ 2.66GHz 2ソケット/サーバー
メモリ	4GB

- 回線シミュレータマシン

機種	日立 BladeSymphony BS320 × 1ブレード
CPU	デュアルコア インテル <sup>®</sup> Xeon <sup>®</sup> プロセッサ 3GHz 2ソケット/ブレード
メモリ	8GB

- ストレージ

機種	Hitachi Adaptable Modular Storage (AMS)
ハードディスク	144GB × 28HDD (+2HDD スペア)
RAID グループ構成	RAID5(2D+1P) × 8 (Oracle データベース・・・プライマリ/スタンバイでそれぞれ4RAID グループを使用)

### 6.1.2 使用ソフトウェア

- データベースサーバー

OS	Red Hat Enterprise Linux 4.5
Oracle	Oracle Database 11g Release 1 (11.1.0.6) Enterprise Edition Oracle Real Application Clusters Oracle Active Data Guard Oracle Advanced Compression Oracle Partitioning

- クライアントマシン

OS	Red Hat Enterprise Linux 4 Update 3
Oracle	Oracle Client 10g Release 2 (10.2)

- 回線シミュレータマシン

OS	Red Hat Enterprise Linux 4.5
ネットワーク 制御機能	netem tc-tbf

## 7. 検証シナリオと着眼点

今回の検証で想定したシナリオと技術的に確認すべきポイントや必要な設定について記述します。

### 7.1 処理の流れ

Oracle Data Guard構成に対して、バッチ処理 / OLTP処理 / 検索処理(レポート生成処理等)を行うケースを想定します。プライマリ・データベースとスタンバイ・データベースで行われる処理のイメージを図 7-1に示します。

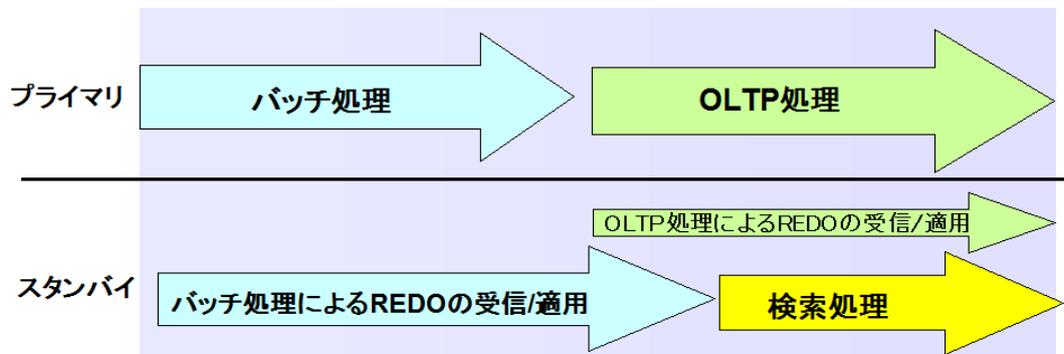


図 7-1 処理イメージ

以降、各処理について説明します。

#### 7.1.1 バッチ処理

バッチ処理では、Oracle Databaseに分析用データを作成するため、データロード / 統計情報取得 / 索引作成を行います。処理内容の表 7-1にまとめます。

処理順	処理名	処理内容
1	データロード	複数の表に対して合計約 30GB のデータをフラット・ファイルから Oracle Database へロードする。外部表機能を使用したパラレル・ダイレクト・ロード・インサートによって実施
2	統計情報取得	分析時に Oracle Database が最適なデータアクセスパスを選択するため、データがロードされた表の物理記憶特性の統計を

		収集する。DBMS_STATS パッケージによって実施
3	索引作成	データがロードされた任意の表に索引を作成する。索引作成はパラレル処理によって統計情報の自動収集を有効にした状態で実施

**表 7-1 バッチ処理内容**

また、バッチ運用方法として以下の2つのケースを想定しました。

**(1) バッチ処理中の REDO 転送を有効にする**

このケースは、Oracle Data Guard の通常の運用方法です。本検証では、REDO 転送を有効にする全てのパターンで非同期転送を設定しました。バッチ処理による更新量によっては、ネットワーク帯域を上回る REDO が転送される可能性があります。この場合、プライマリ-スタンバイ間のタイムラグが大きくなることによって最新の更新データがスタンバイで保護されない状態が起こり得ます。

**(2) バッチ処理中の REDO 転送を停止する**

バッチ処理中の REDO 転送を停止するため、Oracle Data Guard 構成によるバッチ処理性能への影響を考慮する必要はありません。REDO 転送が停止になっている間は、プライマリ-スタンバイ間のタイムラグはリニアに開いていきます。バッチ処理終了後に REDO 転送を再開すると、バッチ処理による REDO はギャップとして自動解決され、タイムラグも短縮します。

**7.1.2 OLTP処理**

OLTP 処理はバッチ処理終了後に開始します。これは「夜間バッチ終了後に通常業務を再開する」といった要件を想定しています。OLTP 処理では、オンライン・ショッピング・サイトを想定したワークロードを実行します。

バッチ処理中に REDO 転送を停止していた場合や、REDO の転送が追いつかない場合は、OLTP 処理開始後に OLTP 処理の REDO と未転送のバッチ処理の REDO が同時に転送されます。

**7.1.3 検索処理**

検索処理は、バッチ処理によって作成されたデータに対して行います。検索処

理をスタンバイ・データベース側で行うことによって、災害対策サイトのサーバーリソースを有効活用します。また、これによって OLTP 処理と検索処理の競合も回避できるため、結果的に OLTP 処理スループット向上や検索処理のレスポンス高速化も期待できます。また、検索処理を実行している間も、スタンバイ・データベースはプライマリ・データベースから送信された REDO の適用処理を継続します。フィジカル・スタンバイ・データベースのこのような活用方法は Oracle Database 11g から新たに提供された Oracle Active Data Guard Option によって実現します。

今回の検証シナリオにおいて注意すべき点は、検索処理はプライマリ・データベースで行われたバッチ処理による REDO がスタンバイ・データベースに適用された時点で開始できるということです。つまり、OLTP 処理と検索処理を基準として、2 つのバッチウィンドウが存在することになります (図 7-2)。プライマリ・データベースからスタンバイ・データベースへの REDO 転送がバッチ処理による REDO 生成に追いついていけば、バッチ処理終了からスタンバイでの REDO 適用完了までにはほとんど時間がかからないため、バッチウィンドウの検討は容易になります。

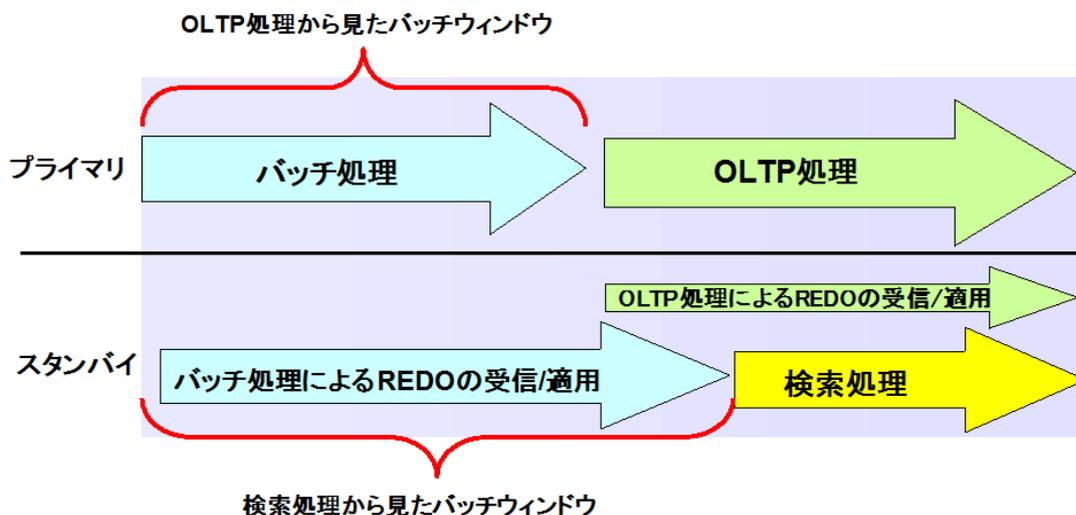


図 7-2 バッチウィンドウ

本検証では、特にこの 2 つのバッチウィンドウ内の処理に注目し、検証を実施しました。

## 7.2 REDO圧縮の設定

REDO 圧縮は Oracle Database 11g の新オプションである Oracle Advanced Compression Option によって提供される機能です。REDO 圧縮によってプライマリ・データベースからスタンバイ・データベースへ転送される REDO が圧縮されます。REDO 圧縮には以下の 2 種類の設定があります。

### 7.2.1 ギャップREDO圧縮

Oracle Data Guard が検知した REDO のギャップ(一時的なネットワーク障害や、本検証シナリオのように REDO 転送を一時的に停止していた間にプライマリで生成された REDO) を自動解決する際、REDO を圧縮します。圧縮されるのはギャップとして認識された REDO のみです。ギャップ REDO 圧縮を有効にするには、プライマリ・データベースで REDO 転送の宛先や転送方法を指定する初期化パラメータ `log_archive_dest_n` の `compression` 属性を `TRUE` に指定します。

### 7.2.2 非同期REDO圧縮

ギャップ解決時に加え、通常時に非同期転送によって転送される REDO も圧縮されます。非同期転送を行うには Oracle Data Guard の保護モードを最大パフォーマンスモードに設定する必要があります。

---

---

非同期 REDO 圧縮の設定方法は、MetaLink note: 729551.1 をご参照ください。

---

---

REDO 圧縮はプライマリ側で REDO 転送時に動作し、スタンバイで受信後に伸張されます。導入時にはこの 2 つの動作による CPU 使用率へのオーバーヘッドを考慮する必要があります。

## 7.3 ネットワークチューニングの実施

ディザスタ・リカバリ等の目的で遠隔地に配置されたサーバー間の通信では、遅延によってネットワーク Round Trip Time (RTT) が長くなり、転送効率が低下する可能性があります。これは、Oracle Data Guard の REDO 転送においても同様です。Oracle Database 11g では、非同期の REDO 転送においてネットワーク遅延による転

送効率への影響が低減されるよう機能改善がされていますが、本検証では、さらにネットワークチューニングを実施し「RTTが長い場合のREDO転送効率」を改善させる効果があるかを確認します。今回の検証で実施したチューニングはOracle Database関連ファイル内のパラメータ編集のみで、OSレベルの変更は行っていません。編集を行ったファイルと設定パラメータについて表 7-2に示します。

設定パラメータ	説明	設定ファイル
SDU (Session Data Unit)	データ転送時にバッファに格納する単位 (バイト)。チューニング時には 32767 を指定	tnsnames.ora listener.ora sqlnet.ora
SEND_BUF_SIZE RECV_BUF_SIZE	TCP ソケットのバッファ・サイズ (バイト)。チューニング時には帯域幅遅延積 (BDP) の 3 倍を指定。帯域幅遅延積はネットワーク帯域幅 (バイト) と RTT (秒) の積によって算出	tnsnames.ora listener.ora sqlnet.ora

**表 7-2 ネットワークチューニング**

---

各設定値の決定と設定方法は、Oracle 社が公開している以下のホワイトペーパーを参考にしています。

「Data Guard REDO 転送とネットワークのベストプラクティス」

<http://www.oracle.com/technology/global/jp/products/availability/htdocs/maa.html>

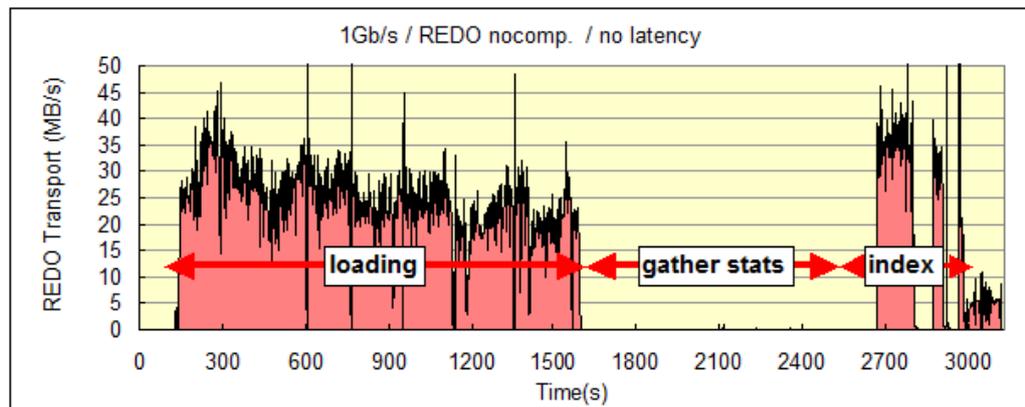
---

## 8. 検証結果

今回の検証結果と考察について示します。

### 8.1 Oracle Data Guard環境におけるバッチ処理の考慮事項

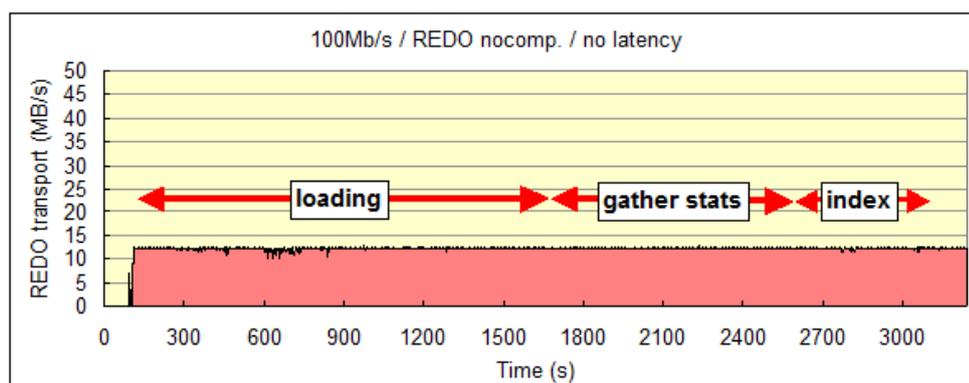
ここでは、Oracle Data Guard上でのバッチ処理について、考慮すべきポイントを確認します。まず、バッチ処理によって転送されるREDOのサイズを確認します。以下のグラフ 8-1はバッチ処理中にプライマリからスタンバイに転送されるREDOサイズの推移（REDO転送量）を示しています。グラフ上の矢印”loading”はデータロードが、”gather stats”は統計情報取得が、”index”は索引作成が実行されていたことを示します。このときのREDO転送ネットワークの帯域は1Gb/sに設定しています。REDO転送量は最大時で約40MB/s 平均で約15MB/sで、グラフに示すとおり、REDOが生成されるのはデータロードと索引作成処理の間で、統計情報取得の間にはREDOはほとんど生成されませんでした。転送量はプライマリのRACデータベースの2ノードから転送されるREDOサイズの合計です。



グラフ 8-1 バッチ処理中の REDO 転送量 (1Gb/s)

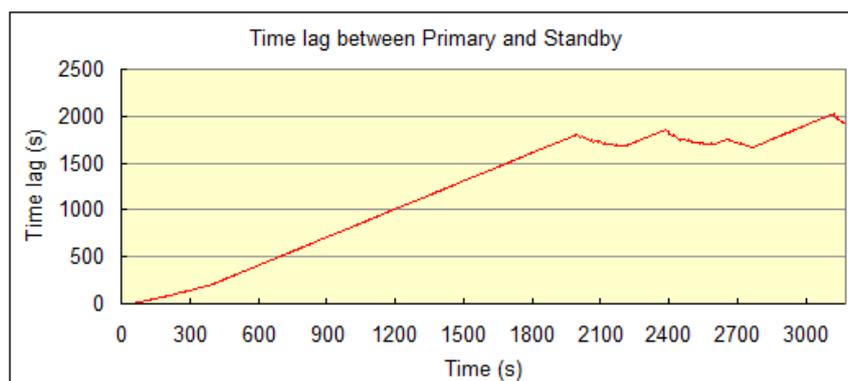
グラフ 8-2はREDO転送ネットワークを100Mb/sに設定した場合の転送REDOサイズの推移です。転送量は常時12.5MB/s程度となっているため、バッチ処理中はネットワーク帯域を使い切っていると言えます。<sup>3</sup>また、統計情報収集にもバッチ処理による未転送のREDOが転送され続けていることが分かります。

<sup>3</sup>8bit = 1Byte であることから、100Mb/s = 12.5MB/s となるため。



グラフ 8-2 バッチ処理中の REDO 転送量 (100Mb/s)

この状態では、REDO転送がプライマリ・データベースの更新に追いつかないため、スタンバイ・データベースとの間に大きなタイムラグが発生する可能性があります。この時のプライマリ - スタンバイ間のタイムラグの推移をグラフ 8-3 に示します。ここで言うタイムラグとは、スタンバイ・データベースが何秒前のプライマリ・データベースの状態であるかを示します。タイムラグは、バッチ処理とは別に 1 秒間隔で現在のタイムスタンプをプライマリ・データベースに INSERT し、スタンバイ・データベースでは 5 秒間隔で INSERT された最新のタイムスタンプを検索することで測定しています。グラフ 8-3からは、時間が経過するごとにタイムラグが大きくなっていくことがわかります。このような状態は、Oracle Data Guardを導入する本来の目的であるデータの保護という観点から、問題であると言えます。



グラフ 8-3 バッチ処理中のプライマリ- スタンバイ間のタイムラグの推移

---

---

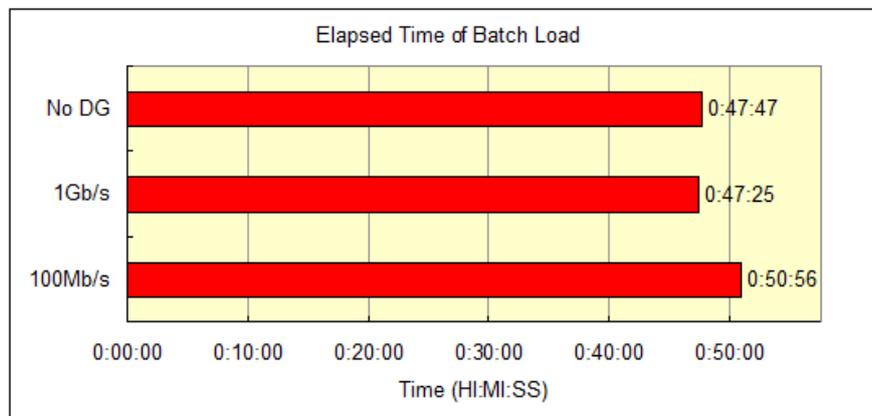
### ポイント：Oracle Data Guard におけるタイムラグ

Oracle Data Guard のタイムラグには”転送ラグ”と”適用ラグ”2つの考え方があります。転送ラグはプライマリの最新 REDO とスタンバイの受信済み最新 REDO の時間差を、適用ラグはプライマリの最新 REDO とスタンバイの適用済み最新 REDO の時間差を表します。今回の検証環境では、REDO 適用性能が REDO 転送量を大きく下回ることがなかったため、転送ラグと適用ラグに大きな差は生じていません。そのため、本検証の結果については、転送ラグと適用ラグをまとめて”タイムラグ”と記述しています。転送ラグ/適用ラグの正確な値はスタンバイ・データベースの V\$DATAGUARD\_STATS ビューより確認可能です。

---

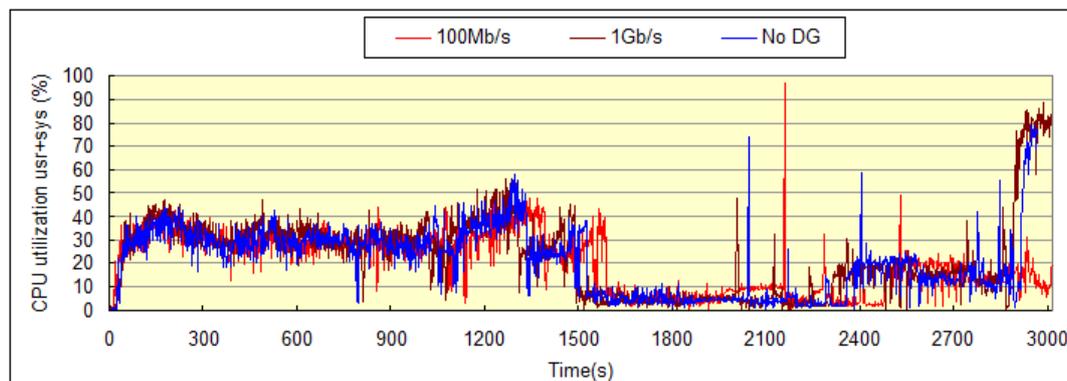
---

次に、バッチ処理の処理時間を比較します。比較するケースはバッチ処理中に REDO 転送を行わないケース (No DG)、1Gb/s のネットワークで REDO 転送を行うケース (1Gb/s)、100Mb/s のネットワークで REDO 転送を行うケース (100Mb/s) の3つです。比較結果をグラフ 8-4 に示します。No DG と 1Gb/s ではバッチ処理時間はほとんど変わらないので、REDO 転送が行われることによるバッチ処理性能への影響はなかったことが分かります。一方、100Mb/s ではバッチ処理時間が No DG や 1Gb/s と比較すると若干長くなっています。1Gb/s と 100Mb/s の動作を比較すると、100Mb/s では、ネットワーク帯域を使い切っている点と、REDO 転送が生成に追いつかないことにより System Global Area (SGA) 内のログ・バッファではなく、オンライン REDO ログやアーカイブ REDO ログから REDO を読み取って転送する点が異なります。今回の検証環境では、これらの動作がバッチ処理のオーバーヘッドに関連していると考えられます。



グラフ 8-4 バッチ処理時間の比較

バッチ処理中のCPU使用率の推移についても確認しました。比較結果をグラフ 8-5に示します。いずれのケースにおいても、CPU使用率に大きな差はないことが分かります。グラフは 2 ノードのRACデータベースサーバーのうち、1 ノードのCPU使用率を比較していますが、もう 1 ノードも、同様のCPU使用率の推移を記録しました。



グラフ 8-5 CPU 使用率の比較

今回の検証シナリオでは、Oracle Data Guard 構成において、特にネットワーク帯域を上回る REDO が生成されるようなバッチ処理では、以下のような考慮事項があることが分かりました。

- (1) プライマリ・データベースのデータを適切に保護できず、RPO に抵触する可能性がある
- (2) バッチ処理性能に影響を及ぼす可能性がある

(1) についてはネットワーク帯域を上回る REDO が生成されるバッチ処理では必ず起こる現象で、(2) についてはバッチ処理の内容や構成によって影響が異なります。上記の課題の最もシンプルな解決方法は、対象システムにおいて REDO 生成量が最も高い時間を確認し、その生成率を上回る REDO 転送ネットワーク帯域を確保することです。但し、災害対策で遠隔地にスタンバイ・データベースを配置する構成では、REDO 転送ネットワークに十分な帯域を確保するのは困難な場合も多くあります。

---

---

### ポイント：logging とアーカイブ・ログ・モード

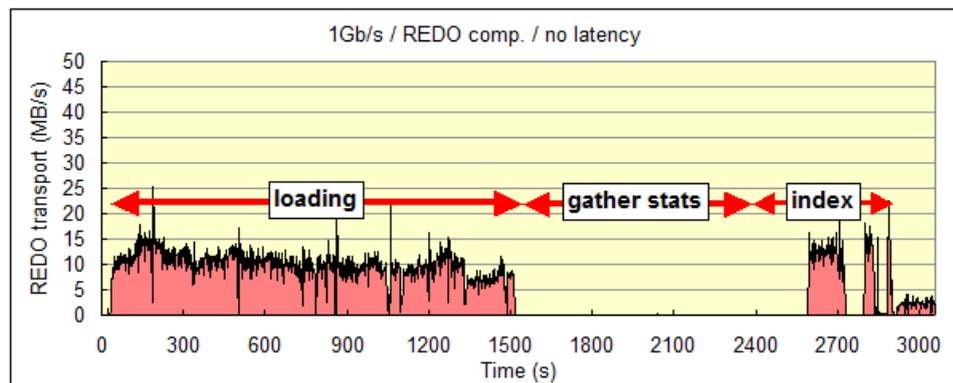
Oracle Data Guard では、REDO 転送によってデータを保護するため、プライマリ・データベースの全ての処理を REDO ログに記録 (logging 設定) し、アーカイブを有効 (アーカイブ・ログ・モード) にする必要があります。バッチのような大量の更新によって大量の REDO が生成される処理では、REDO ログへの REDO の書き込みとアーカイブ出力のための読み取り動作が性能のボトルネックとなる可能性があります。今回の検証構成においても、REDO ログが配置されたディスクに高い負荷がかかり、バッチ処理性能に影響を及ぼしました。このボトルネックを回避し、処理性能をさらに向上させるためには、REDO ログに対する I/O を高速化するためのディスク構成 (高速なディスクの使用やストライピング) を検討する必要があります。

---

---

## 8.2 非同期REDO圧縮の効果

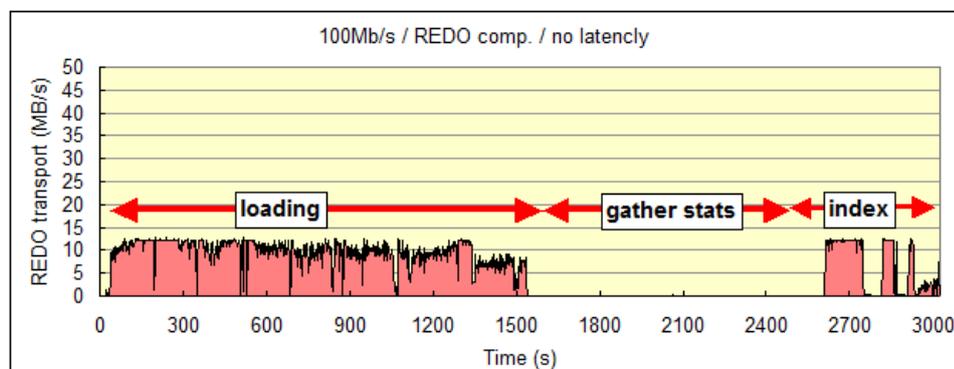
ここでは、バッチ処理時に非同期REDO圧縮を設定した場合の効果と影響を確認します。まず、非同期REDO圧縮を設定した場合のバッチ処理時のREDO転送量を、ネットワーク帯域を 1Gb/sにした状態で確認しました。グラフ 8-6 に結果を示します。グラフ 8-1と比較すると、非同期REDO圧縮によってREDO転送量が半分以下になっていることが分かりました。



グラフ 8-6 バッチ処理中の REDO 転送量 (1Gb/s)

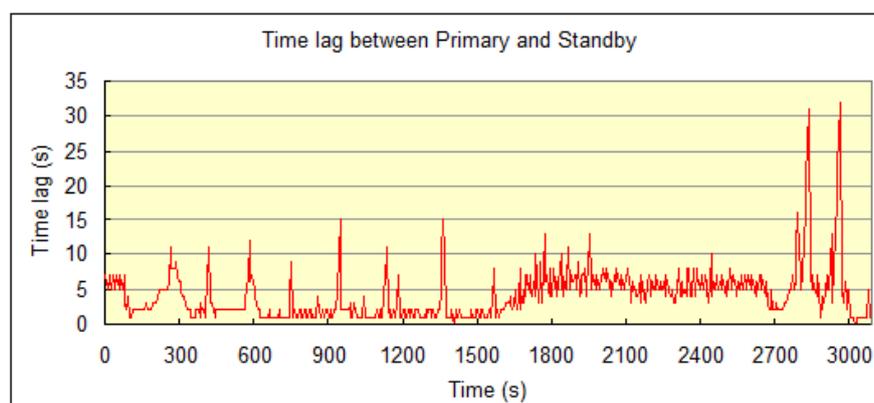
非同期REDO圧縮設定時には、REDO転送量は最大時では約 15MB/sとなっていますが、転送量が 12.5MB/sを超える時間はごくわずかなため、REDO転送ネットワークの帯域を 100Mb/sに変更した場合もネットワークの使用状況に大きな変更はありません。検証結果をグラフ 8-7に示します。グラフ 8-2と比較すると、REDO

圧縮によってREDO転送の処理がバッチ処理に追いつていることが分かります。



グラフ 8-7 バッチ処理中の REDO 転送量 (100Mb/s)

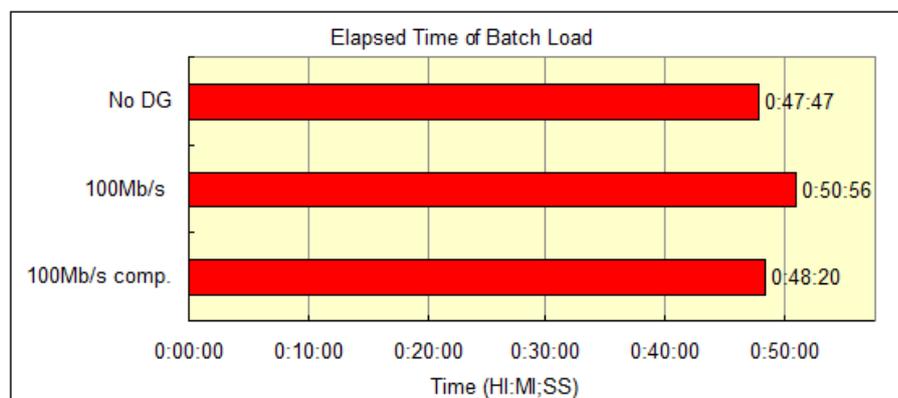
このときのプライマリ・データベースとスタンバイ・データベースのタイムラグをグラフ 8-8に示します。グラフより、大きなタイムラグは発生せずにスタンバイ・データベースがプライマリ・データベースの変更を追従できていることがわかります。データ保護の観点において、非同期REDO圧縮が効果的に働いたといえます。また今回の検証シナリオにおいては、スタンバイ・データベースを使用した検索処理がより早く開始できるというメリットもあります。



グラフ 8-8 バッチ処理中のプライマリ-スタンバイ間のタイムラグ (圧縮あり)

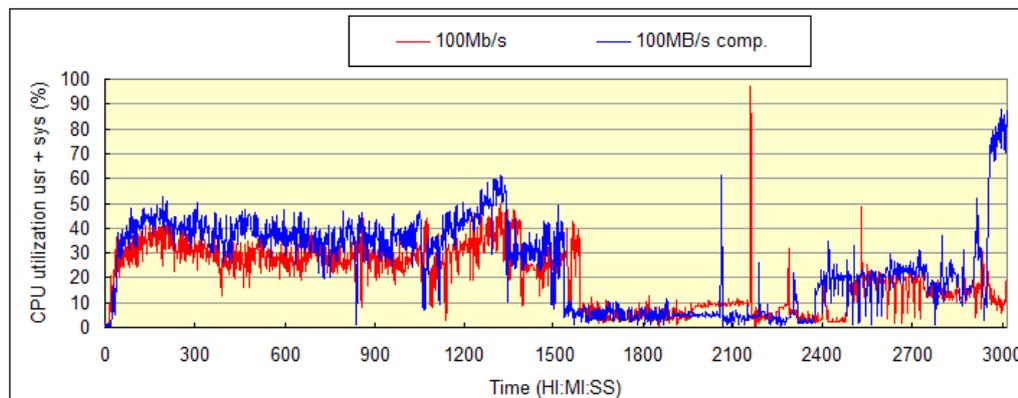
次に、バッチ処理の処理時間を比較します。比較するケースはバッチ処理中にREDO転送を行わないケース (No DG)、100Mb/sのネットワークでREDO転送を行うケース (100Mb/s)、100Mb/sのネットワークで非同期REDO圧縮を設定してREDO転送を行うケース (100Mb/s comp.) の3つです。検証結果をグラフ 8-9に示します。100Mb/s comp. の処理時間は100Mb/sの場合よりも短く、No DG と比較した場合にもオーバーヘッドは大きくありません。これは、圧縮によってREDO転送量

がネットワーク帯域をほとんどの時間上回らなくなったためと考えられます。



グラフ 8-9 バッチ処理時間の比較

同様に、バッチ処理中のCPU使用率の推移も比較します。比較結果をグラフ 8-10に示します。グラフより、非同期REDO圧縮を設定している場合、REDO転送中はCPU使用率が高くなっている時間帯があることが分かります。これは、圧縮処理によってCPUリソースが消費されていることを示しています。



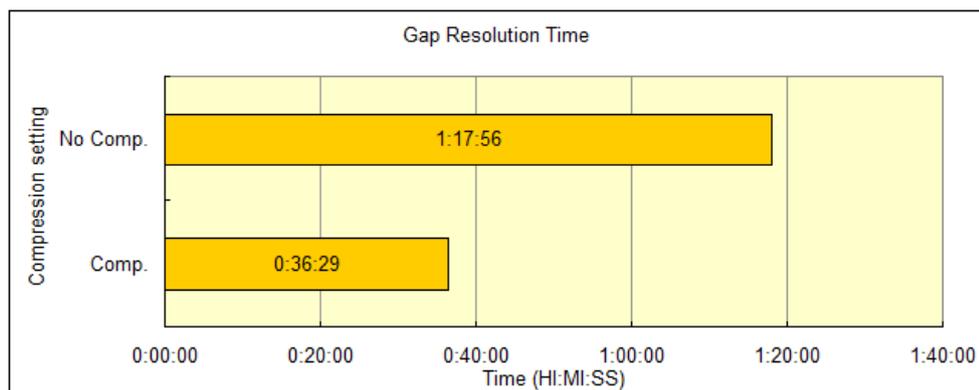
グラフ 8-10 バッチ処理中の CPU 使用率のグラフ

以上のように、非同期 REDO 圧縮を設定することにより、REDO 転送量を抑えることが可能です。今回の検証シナリオでは、非同期 REDO 圧縮によって REDO 転送量がネットワーク帯域を下回りました。これによってプライマリ-スタンバイ間のタイムラグが短い状態を維持し、適切なデータ保護を実現しました。さらに圧縮を設定しない場合に比べ、バッチ処理時間へのオーバーヘッドを軽減できることが確認できました。CPU リソースのオーバーヘッドが許容範囲内であれば、Oracle Data Guard 構成において非常に有効な機能であるということが言えます。

### 8.3 バッチ処理時のREDO転送を停止する運用方法

Oracle Data Guard 環境におけるバッチ処理では、大量の REDO を生成するバッチ処理時には REDO 転送を停止するという運用方法も考えられます。この運用方法には、バッチ処理性能に対する REDO 転送の影響を考慮する必要がないという利点と、バッチ処理中はスタンバイ・データベースでデータが保護されないという課題があります。今回の検証シナリオにこの運用方法を適用した場合、バッチ処理後の OLTP 処理中に、バッチ処理で生成された REDO がギャップ解決の仕組みで転送されます。ここでは、ギャップ解決にかかる時間やギャップ解決処理が発生することによる OLTP 処理への影響およびギャップ REDO 圧縮の効果について確認します。

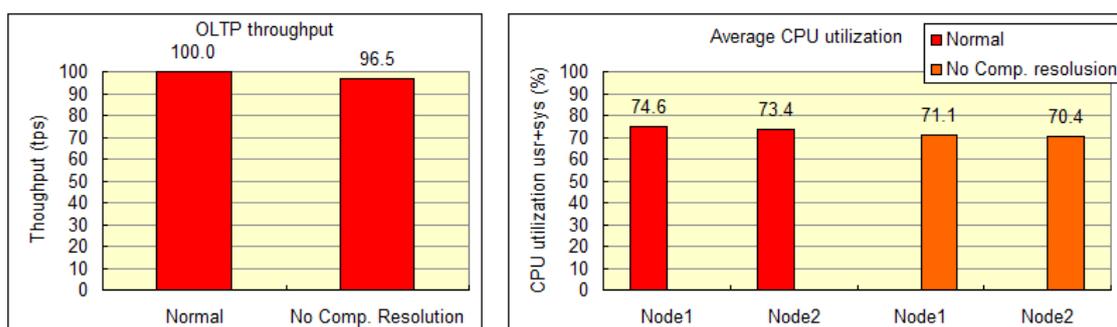
まず、ギャップ REDO 圧縮設定の有無によるギャップ解決時間を比較します。ここでいうギャップ解決時間とは、OLTP 処理が始まってからバッチ処理の REDO が全て適用されるまでの時間を指し、この時間が短いほど、バッチ処理による更新データをより早く保護できることとなります。また、今回の検証シナリオでは、ギャップ解決時間が短いほど、スタンバイ・データベースを使用した検索処理を早く開始することが可能です。検証結果をグラフ 8-11 に示します。グラフより、ギャップ REDO 圧縮によってギャップ解決時間が半分以下になっていることがわかります。



グラフ 8-11 ギャップ解決時間の比較

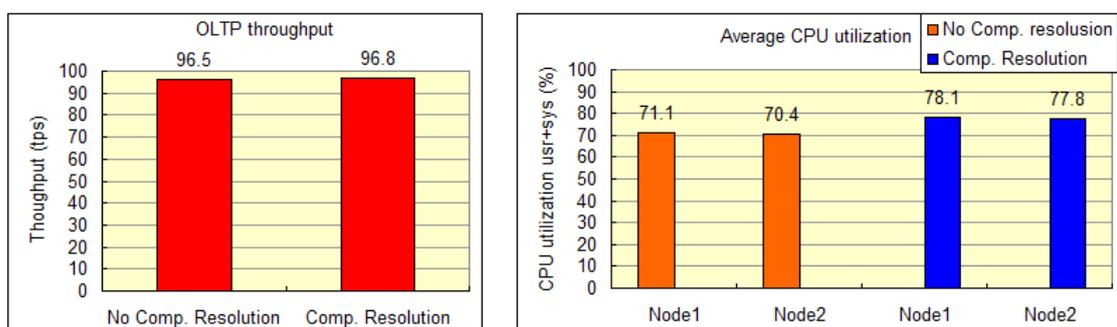
次に、ギャップ解決中の OLTP 処理性能について確認します。グラフ 8-12 では、通常の OLTP 処理 (Normal) とギャップ解決中の OLTP 処理 (No Comp. resolution)

について、平均スループットと平均CPU使用率<sup>4</sup>を比較しています。グラフより、No Comp. resolutionでは、スループットとCPU使用率共にNormalを若干下回っています。これは、スループットが若干低下しているが、ボトルネックはCPUではない可能性が高いことを示しています。ギャップ解決中は、非同期転送によってカレントのログ・グループのREDOが転送されるのに加え、アーカイブREDOログの両方からREDOを読み取って転送することになるため、この動作がスループットに影響を及ぼしたと考えられます。



グラフ 8-12 OLTP 処理の性能比較 (Normal / No Comp. resolution)

グラフ 8-13はギャップ解決中のOLTP処理においてギャップREDO圧縮設定のなし / あり (No Comp. resolution / Comp. resolution )の比較です。ギャップREDO圧縮を設定した場合、ギャップ解決によって転送されるREDOのみが圧縮され、OLTP処理によって転送されるREDOは圧縮されません。グラフより、ギャップREDO圧縮のあり/なしではスループットに差がほぼないことが分かります。また、CPU使用率に注目すると、ギャップREDO圧縮設定時には、RACデータベースそれぞれのノードで7%程度のCPU使用率のオーバーヘッドがあることがわかります。



<sup>4</sup>平均スループットと平均CPUはログ・スイッチ間の時間の平均を計算しています。また、平均スループットはNormal時が100となるように係数化しています。

## グラフ 8-13 OLTP 処理の性能比較 (No Comp. resolution / Comp. resolution)

REDO 圧縮の処理量は REDO 転送量によって決まります。転送量が多くなれば圧縮処理量も増加するため、結果として CPU 使用率のオーバーヘッドも増加します。

本項の冒頭で述べたとおり、バッチ処理時に REDO 転送を停止する運用方法では、Oracle Data Guard によるバッチ処理性能への影響を考慮する必要がないという利点がありますが、以下の課題があります。

- (1) バッチ処理による変更をスタンバイ・データベースに反映させるために、**REDO のギャップ解決が必要**
- (2) ギャップ解決の動作がバッチ処理後の処理に影響を与える可能性がある

これらの課題に対して、今回の検証シナリオにおける検証結果から、以下のことが確認できました。

- (1) ギャップ解決時間はギャップREDO圧縮によって短縮可能 (ネットワーク帯域 100Mb/s で確認)
- (2) ギャップ解決により、OLTP処理のスループットが若干低下したが、大きな影響はない。ギャップREDO圧縮を設定した場合も、CPU使用率のオーバーヘッドが許容範囲であれば同様

### 8.4 ネットワークチューニングの効果

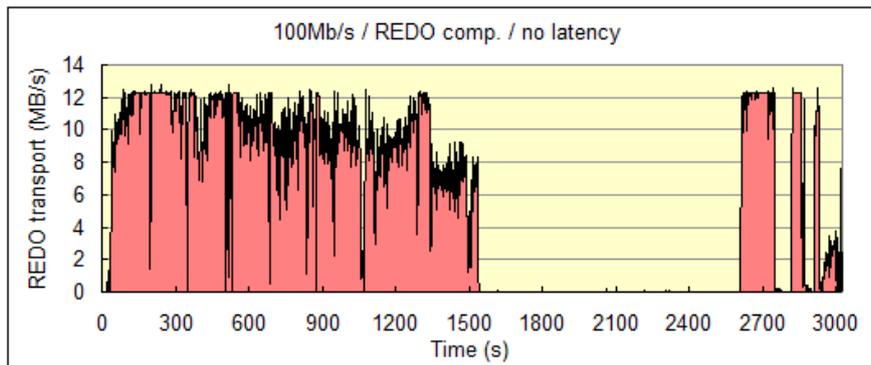
ここまでの検証結果は全て REDO 転送ネットワークに遅延を設定しない状態でのものでした。ここでは、より現実的なディザスタ・リカバリシステムを想定するため、REDO 転送ネットワークの遅延を設定した上でバッチ処理の検証シナリオを実行しました。ネットワーク遅延による RTT は 12ms と 40ms の 2 つのケースを検証しました。過去の事例より 12ms は東京 - 大阪間を、40ms は東京 - 沖縄間を想定しています。REDO 転送ネットワーク帯域は 100Mb/s に設定し、非同期 REDO 圧縮を有効にした状態で検証を行いました。

---

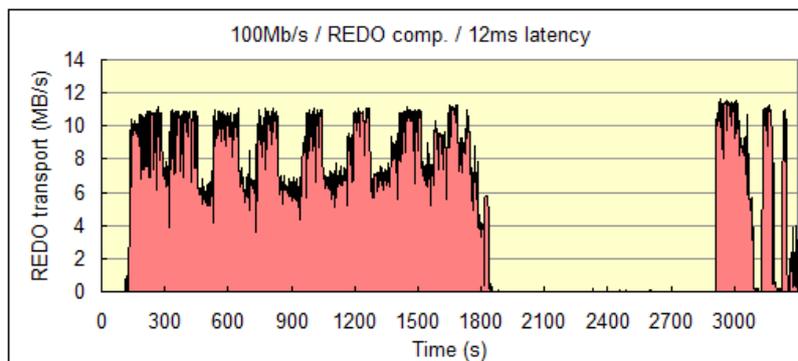
**注意：**転送距離とネットワーク遅延の関係は、構築されるネットワーク環境ごとに異なります。今回の設定値はあくまで参考としてご利用ください。

---

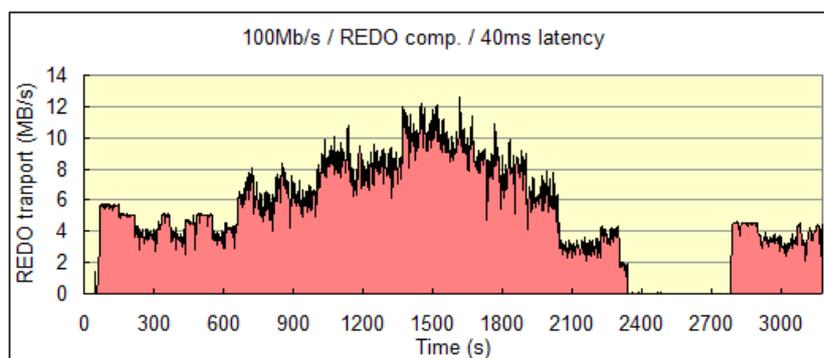
まず、バッチ処理時のREDO転送を有効にしたパターンでのREDO転送量を確認します。グラフ 8-14はネットワーク遅延を設定しない場合、グラフ 8-15は遅延を12msに設定した場合、グラフ 8-16は遅延を40msに設定した場合のREDO転送量の推移です。いずれもケースもネットワーク関連の設定はデフォルト状態です。遅延時間が長いほどREDO転送量が安定せず、転送効率が低下していることが分かります。これがネットワーク遅延の影響です。



グラフ 8-14 REDO 転送量の推移（ネットワーク遅延無し）

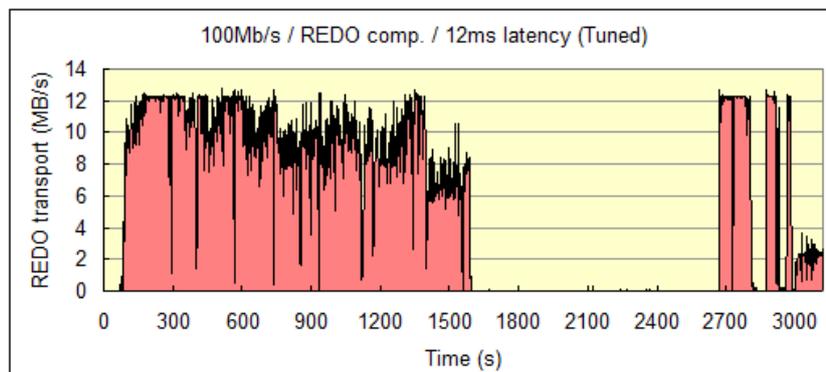


グラフ 8-15 REDO 転送量の推移（ネットワーク遅延 12ms）

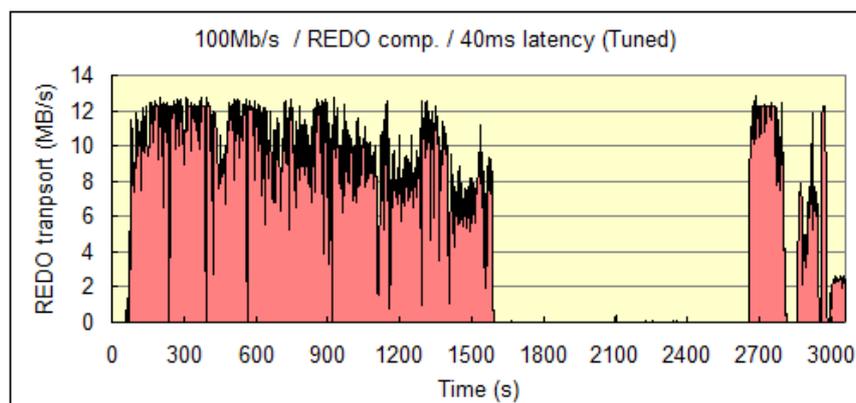


グラフ 8-16 REDO 転送量の推移（ネットワーク遅延 40ms）

これに対して、表 7-2に示したネットワークチューニング実施後のREDO転送量の推移を示したのが、グラフ 8-17とグラフ 8-18です。ネットワークチューニングによって、転送効率が改善し、ネットワーク遅延を設定していないグラフ 8-14とほぼ同じ推移をしていることがわかります。

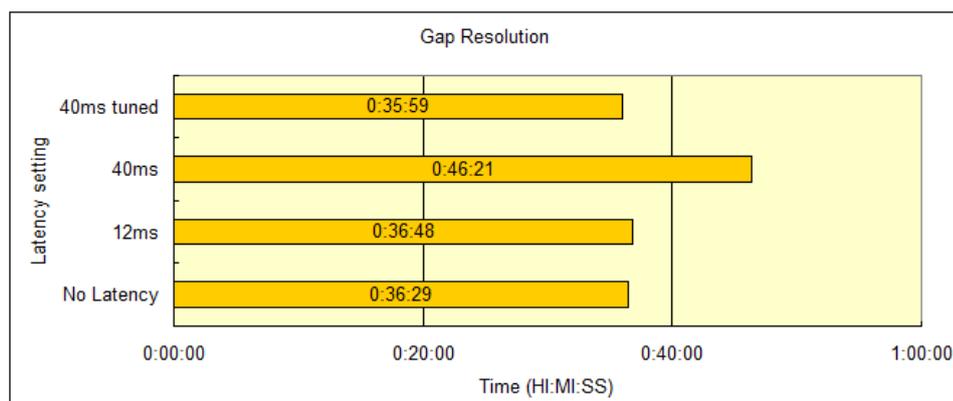


グラフ 8-17 REDO 転送量の推移 (ネットワーク遅延 12ms チューニング済み)



グラフ 8-18 REDO 転送量の推移 (ネットワーク遅延 40ms チューニング済み)

次に、バッチ処理時にREDO転送を停止したケースについて確認します。検証結果をグラフ 8-19に示します。ネットワーク遅延を設定していない状態と 12msの遅延を設定した場合を比較すると、ギャップ解決までの時間にはほとんど差がなくネットワーク遅延による影響はほぼないことが分かります。一方、ネットワーク遅延を設定していない状態と 40msの遅延を設定した場合を比較すると、ネットワーク遅延設定がある場合にギャップ解決に時間がかかる結果が出ました。これはネットワーク遅延による転送効率の低下が原因であると考えられます。この劣化もネットワークのチューニングによって回避可能です。チューニング実施後のギャップ解決時間はネットワーク遅延を設定していない状態と同等でした。



**グラフ 8-19 ギャップ解決時間の比較**

以上の結果より、ネットワークチューニングの効果が確認できました。REDO 転送ネットワークに遅延が生じる場合、転送効率が低下しますが、ネットワークチューニングによって、転送効率は大幅な改善が期待できます。

## 9. まとめ

今回は、ディザスタ・リカバリ構成におけるバッチ処理に注目しました。検証は、日立の高信頼ブレードサーバー BladeSymphonyと、Oracle Database 11g Oracle Data Guardの組み合わせによる災害対策環境を構築した上で、実運用を想定したバッチ処理のシナリオを使用して実施しました。Oracle Data Guard構成上のバッチ処理では、処理性能の観点（処理時間がバッチウィンドウ内に収まっているか）に加え、データ保護の観点（スタンバイ・データベースとのタイムラグ）についても考慮する必要があります。今回の検証では、バッチ処理中にREDO転送を有効にする場合（DG）と無効にするパターン（No DG）を想定し、さらにOracle Database 11g の新機能である転送時のREDO圧縮機能設定（DG + Comp. / No DG + Comp.）による効果を確認しました。検証結果からバッチ処理性能とデータ保護の観点においてそれぞれのパターンの特徴をまとめると、以下の表 9-1のようになります。

パターン	バッチ処理性能	データ保護
DG	REDO 転送量がネットワーク帯域を上回るとオーバーヘッドが生じる可能性がある	REDO 転送量がネットワーク帯域を上回るとプライマリ-スタンバイ間のタイムラグが大きくなる
DG + Comp.	圧縮処理による CPU 使用率へのオーバーヘッドが許容範囲であれば、圧縮設定によるバッチ処理性能へのオーバーヘッドはほとんどない	圧縮によって、REDO 転送量がネットワーク帯域内に収まれば、タイムラグが大きく開くことはない
No DG	Oracle Data Guard による影響なし	バッチ処理中に更新データは保護されない。更新データをスタンバイに反映するために、バッチ処理後のギャップ解決が必要
No DG + Comp.	Oracle Data Guard による影響なし	バッチ処理中に更新データは保護されない。バッチ処理後ギャップ解決が必要だが、No DG に比べ解決時間は短縮や使用ネットワーク帯域の低減が可能

表 9-1 各パターンにおけるバッチ処理の特徴

今回の検証結果からは、Oracle Database 11gのAdvanced Compression Optionによって提供されるREDO圧縮機能が非常に有効に働くことが確認できました。実際には、表 9-1の特徴に加え、さらにバッチウィンドウやギャップ解決時間の許容範囲、REDO圧縮設定時のCPU使用率のオーバーヘッドなどがOracle Data Guard構成上でのバッチ処理をどのパターンで実装するかの判断材料となります。

また、遠隔地にスタンバイ・データベースを配置する場合に懸念されるネットワーク遅

延による転送効率の低下についても、Oracle のネットワーク・パラメータを適切に設定することによって大幅に改善されることが確認できました。

今回の検証から得たノウハウが、より多くの現場でご活用頂けることを期待します。

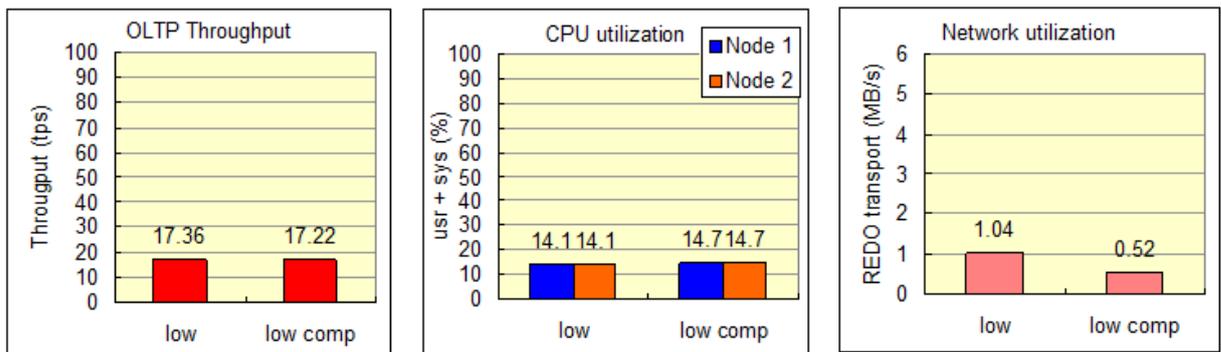
## 10. 謝辞

2006年11月、日本オラクル株式会社は株式会社日立製作所やグリッド戦略パートナー各社と協業体制を確立し、企業のシステム基盤の最適化を実現する次世代のビジネス・ソリューションを構築するため、先鋭の技術を集結した「Oracle GRID Center (オラクル・グリッド・センター)」([http://www.oracle.co.jp/solutions/grid\\_center/index.html](http://www.oracle.co.jp/solutions/grid_center/index.html)) を開設しました。

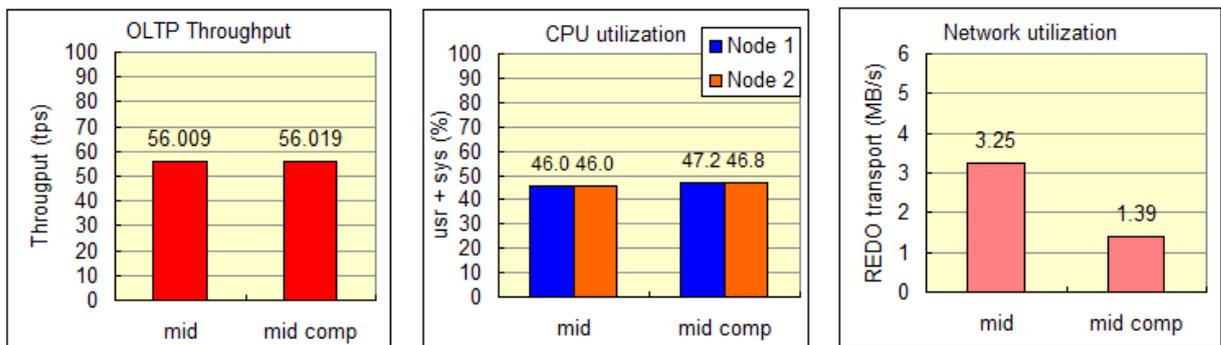
本稿は、Oracle GRID Center の趣旨にご賛同頂いたインテル株式会社、シスコシステムズ合同会社のハードウェア・ソフトウェアのご提供および技術者によるご支援などの多大なるご協力を得て作成しております。ここに協賛企業各社およびご協力頂いた技術者に感謝の意を表します。

## Appendix. OLTP処理における非同期REDO圧縮の効果

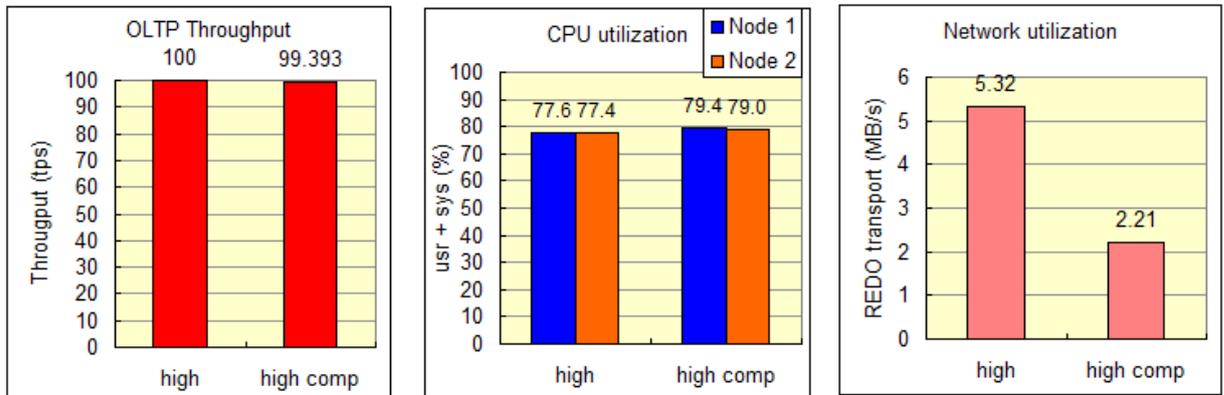
本ホワイトペーパーでは、Oracle Data Guard 構成におけるバッチ処理に注目し検証を行いました。ここでは Appendix として、OLTP 処理における非同期 REDO 圧縮の効果と影響についても確認します。OLTP 処理には、本編でも使用したオンライン・ショッピング・サイトを想定したテストアプリケーションを使用しました。検証では、まず 3 種類の負荷設定（低負荷(low) / 中負荷(mid) / 高負荷(high)）で性能測定を行い、次に非同期 REDO 圧縮を有効にした状態で同様の負荷設定（低負荷(low comp.) / 中負荷(mid comp.) / 高負荷(high comp.)）での性能測定を行いました。同種類の負荷設定において、圧縮の有無による性能比較を行います。比較するのは平均スループット、CPU 使用率、ネットワーク帯域の平均使用量の 3 つです。平均スループットは、圧縮無しの高負荷時(high)のスループットが 100 となるように係数化しています。検証結果を以下のグラフに示します。



グラフ 1 REDO 圧縮設定による OLTP 処理性能の比較（低負荷）



グラフ 2 REDO 圧縮設定による OLTP 処理性能の比較（中負荷）



グラフ 3 REDO 圧縮設定による OLTP 処理性能の比較 (高負荷)

いずれの負荷においても、非同期 REDO 圧縮設定の有無によるスループットおよび CPU 使用率に差はほとんどありません。一方で REDO の平均転送量は、圧縮設定時には低負荷時に半分、中負荷 / 高負荷時には半分を下回るサイズまで低減できています。つまり、非同期 REDO 圧縮は OLTP 処理においても REDO 転送ネットワーク帯域の節約という観点で非常に効果的であるといえます。以上の検証結果より、OLTP 中心のシステムでは非同期 REDO 圧縮を導入することにより、ネットワークにかかるコストを容易に抑えることが可能と考えられます。

## Author

日本オラクル株式会社

セールスコンサルティング統括本部

テクノロジーレディネス SC 本部

Grid Center

シニアエンジニア

後藤陽介

[yosuke.goto@oracle.com](mailto:yosuke.goto@oracle.com)

株式会社 日立製作所

プラットフォームソフトウェア本部

オラクルビジネス統括センター

技師

片山仁史

[yoshihito.katayama.bh@hitachi.com](mailto:yoshihito.katayama.bh@hitachi.com)



日本オラクル株式会社

〒107-0061

東京都港区北青山 2-5-8

オラクル青山センター

Copyright © 2008 Oracle Corporation Japan. All Rights Reserved.

Copyright © 2008 Hitachi, Ltd. All Rights Reserved.

無断転載を禁ず

このドキュメントは単に情報として提供され、内容は予告なしに変更される場合があります。このドキュメントに誤りが無いことの保証や、商品性又は特定目的への適合性の黙示的な保証や条件を含め明示的又は黙示的な保証や条件は一切無いものとします。日本オラクル株式会社および株式会社日立製作所は、このドキュメントについていかなる責任も負いません。また、このドキュメントによって直接又は間接にいかなる契約上の義務も負うものではありません。このドキュメントを形式、手段（電子的又は機械的）、目的に関係なく、日本オラクル株式会社および株式会社日立製作所の書面による事前の承諾なく、複製又は転載することはできません。

BladeSymphony は、国内および海外における日立製作所の登録商標です。

Oracle、JD Edwards、PeopleSoft、及び Siebel は、米国オラクル・コーポレーション及びその子会社、関連会社の登録商標です。その他の名称は、各社の商標または登録商標です。