

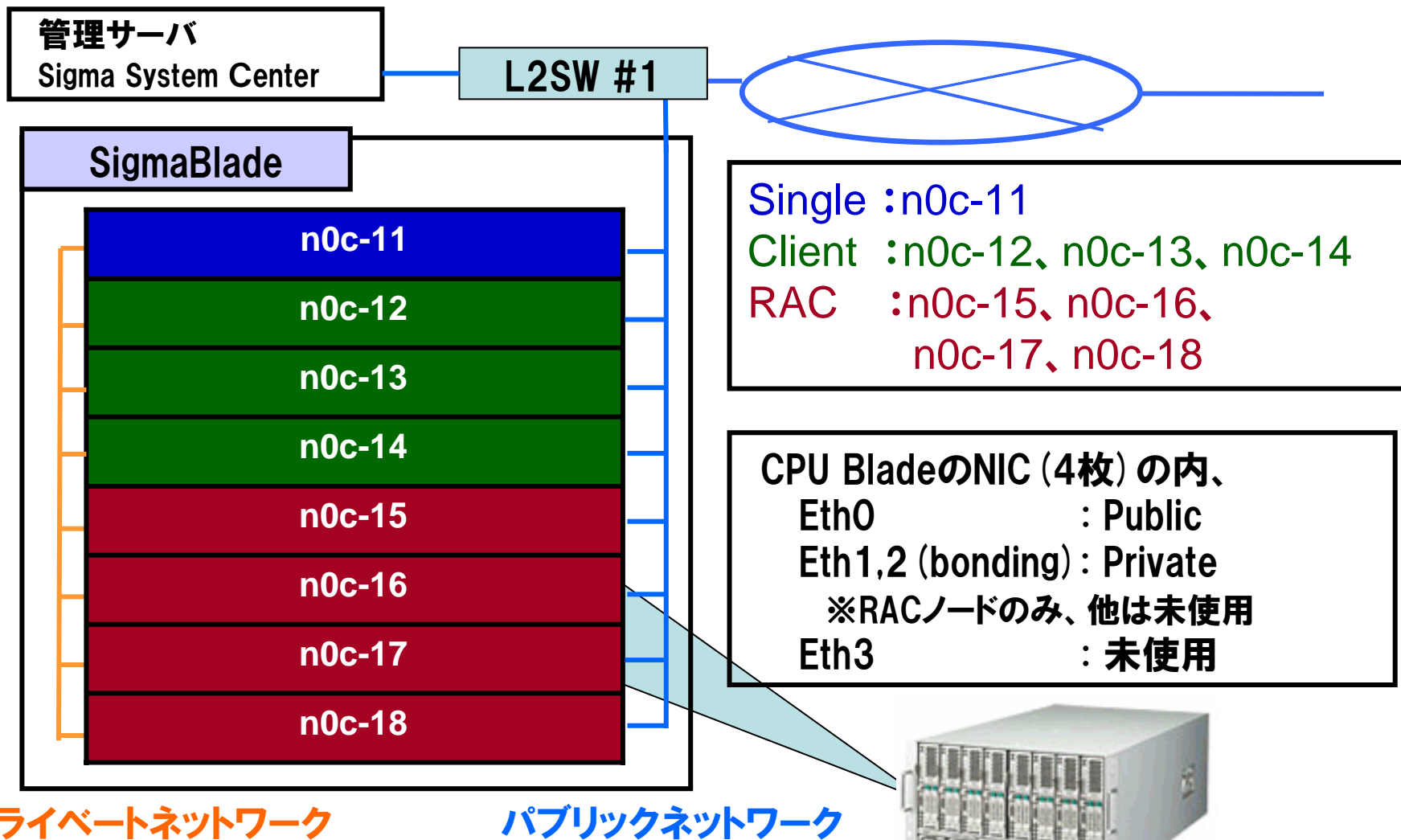
Oracle Database 11g Real Application Testing 検証結果 報告資料

日本電気株式会社

検証の目的

- **ミッションクリティカルシステムにおけるシステム変更リスクを低減させるために、Real Application Testing Database Replay (以降、RAT)機能が大規模環境で確実に動作し、負荷再現性の観点において利用可能な品質であること確認する**
- **RAT使用時のリソース使用状況およびデータを取得し、実際に使用する際のリソース・サイジングを可能にする**

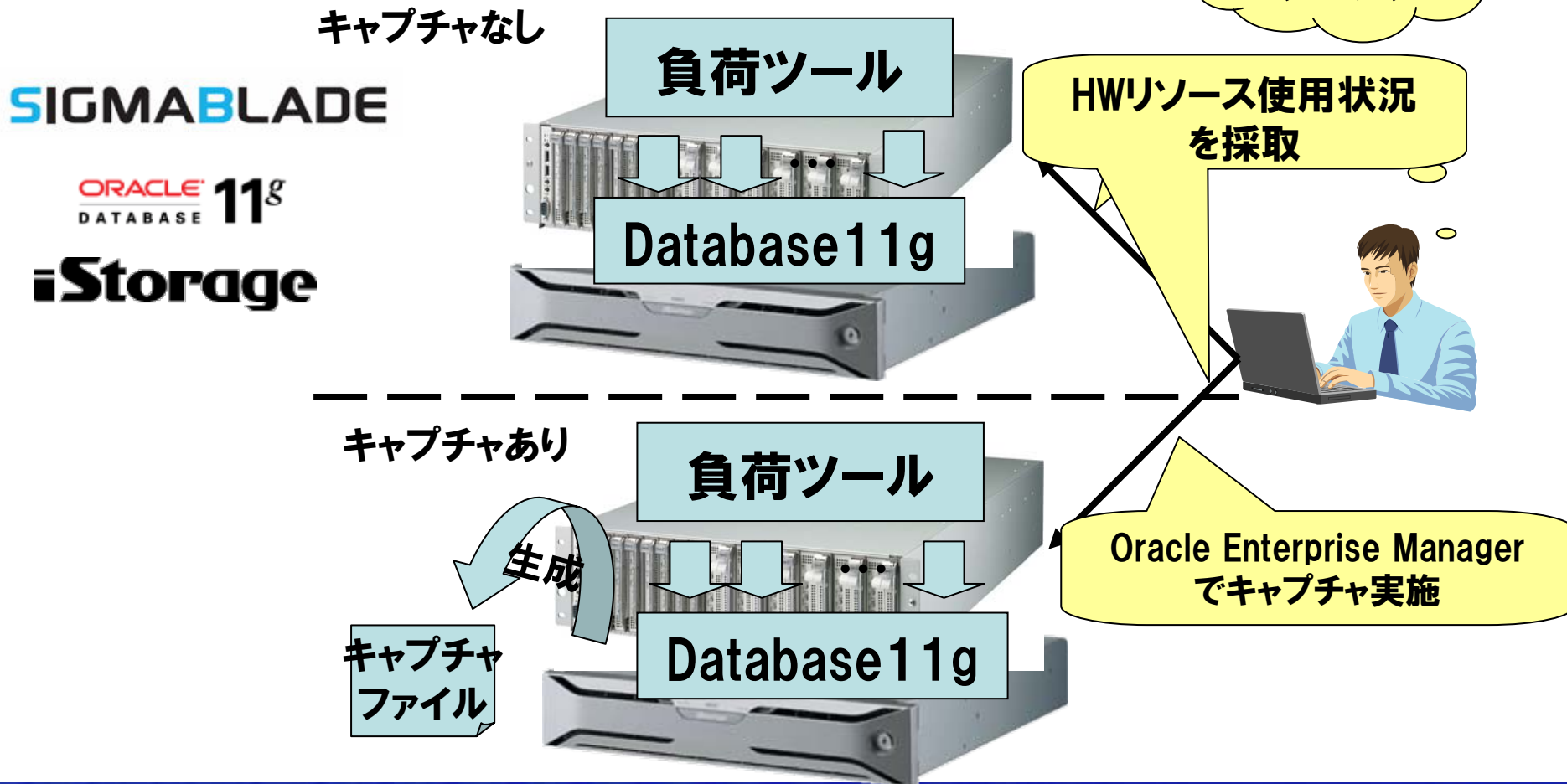
検証環境(HW構成図) [ネットワーク・サーバ]



SIGMABLADE

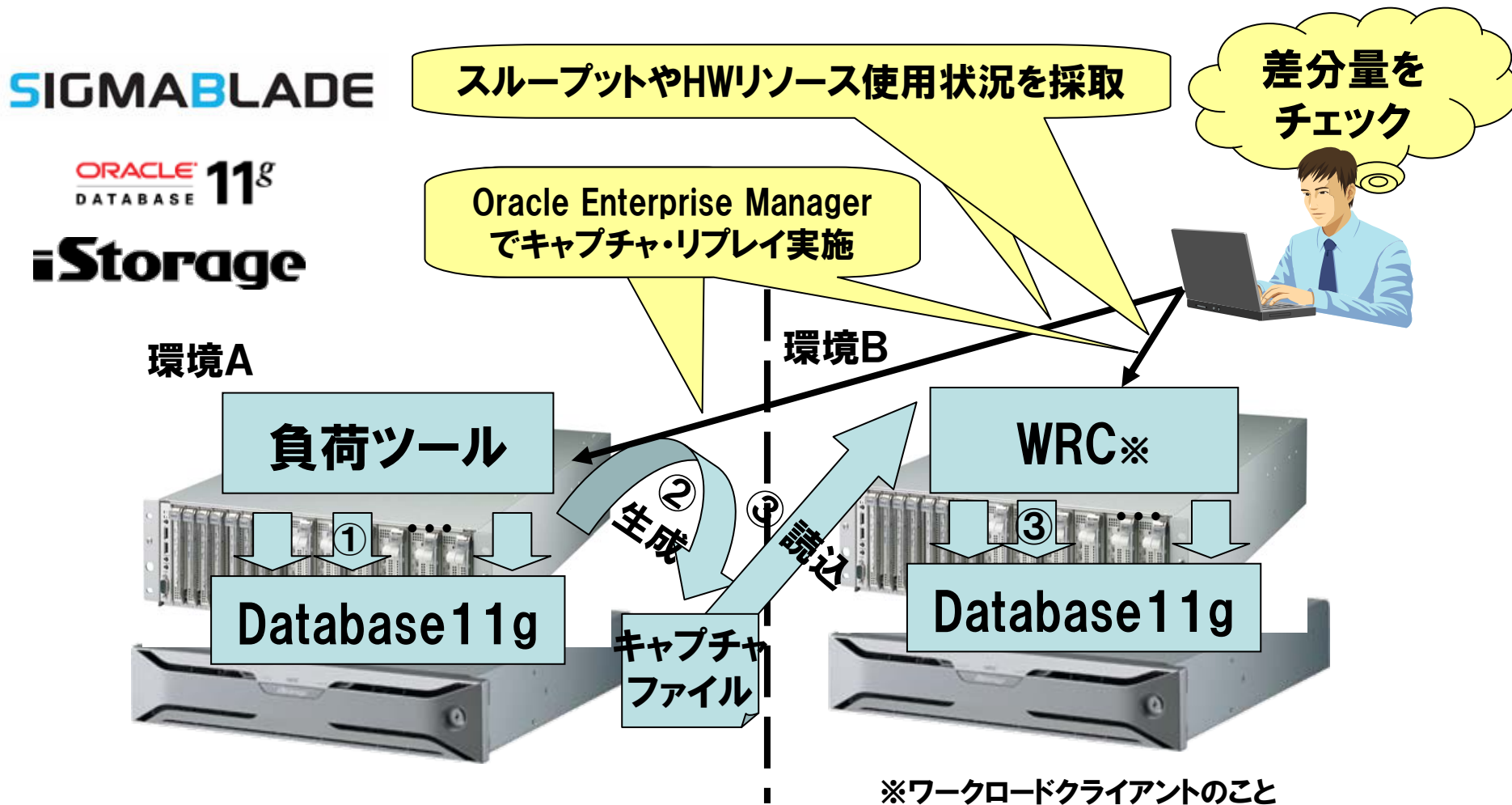
検証内容（概要:イメージ）

1. キャプチャ有無によるHWリソース増分量を確認



検証内容（概要：イメージ）～続き～

2. キャプチャ時とリプレイ時のスループット・HWリソース使用量の差分を比較



検証内容（詳細）

■ 検証内容

1. キャプチャ実施有無によるHWリソース増分量

- 負荷ツールのみと負荷ツール実行時のキャプチャ時の以下の差分(オーバーヘッド)を確認
 - ・CPU使用量差 (単位:%)
 - ・I/O(read/write)量 (単位:ブロック数/秒)
 - ・NWTラフィック(受信)量 (単位:ブロック数/秒)

2. キャプチャ時とリプレイ時でのスループット、HWリソース使用量の差分

- 負荷ツールで10分程度流したトランザクションをRATでキャプチャし、キャプチャファイルのリプレイを擬似テスト環境(Flashback Databaseで負荷ツールを流す前に戻す)で実施
- 以下のように多重度を変え、テストを実施
 - ・多重度(スレッド数) 4、8、12、40
- キャプチャ時とリプレイ時の以下の差分を確認
 - ・トランザクションのスループット差 (単位:秒)
 - ・CPU使用量差 (単位:%)
 - ・I/O(read/write)量 (単位:ブロック数/秒)
 - ・NWTラフィック量 (単位:ブロック数/秒)

上記検証で使用した負荷ツールは以下の2種類

- JPetStore JTA:
SQLを任意の多重度で実行するJavaClientプログラム
- NEC HAテストキット:
ミッションクリティカルシステムを想定したテストプログラム群

検証結果（まとめ）

1. キャプチャによるHWリソース増分量

大きな増分はなし（CPU使用量、I/O (read/write) 量、NWトラフィック量）

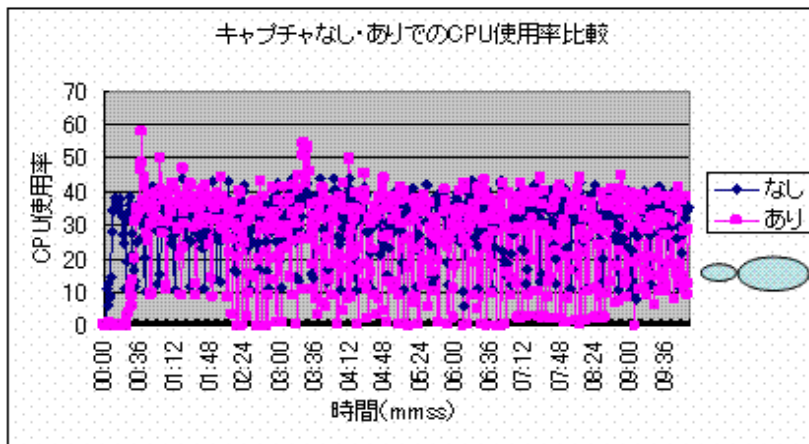
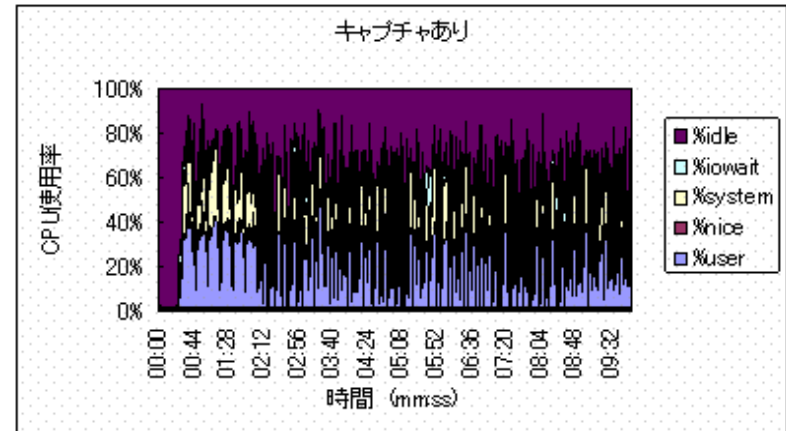
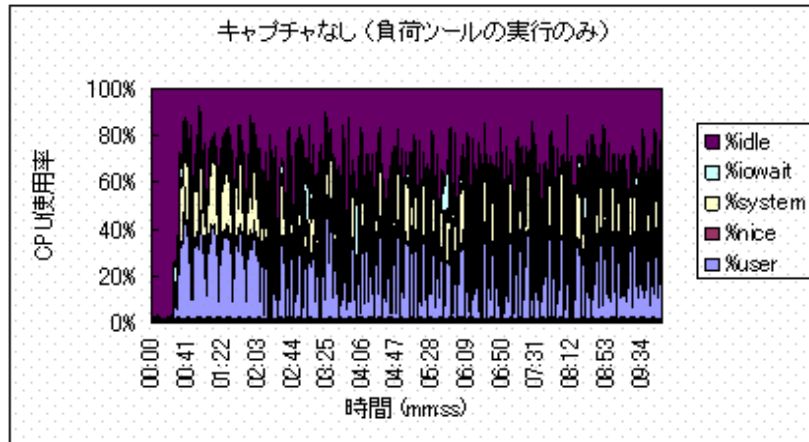
2. キャプチャ時とリプレイ時でのスループット、HWリソース使用量の差分

- 中低負荷時、キャプチャ時とリプレイ時の完了時間に差は殆どなし（実行時間誤差10%以下に収まっている）
- 高負荷時、キャプチャ時とリプレイ時で完了時間の差が大きくなる傾向あり（負荷状況に比例して、完了時間の差も大きくなる）
→ Replay時のパラメータであるsynchronizationをfalseにすることで完了時間の差が改善される

※詳細については、現在(2007/10/23時点)も検証、確認中

検証結果（詳細）

1. キャプチャによるHWリソース増分量を確認 ～CPU使用量～



キャプチャ有無でのCPU使用量の差はほとんどなし

※ I/O (read/write) 量、NWトラフィック量もほとんど差はなし

NEC HAテストキットを使用

検証結果（詳細）

2. キャプチャ時とリプレイ時でのHWリソース使用量の差分 ～実行時間、スループット～

■実行時間

スレッド数	4	8	12	40
キャプチャ (mm:ss) -①	11:29	11:31	11:44	11:42
リプレイ (mm:ss) -②	11:42	11:42	12:00	18:08
②÷①	1.02	1.02	1.02	1.55

高負荷だと実行時間や
スループットが延びた

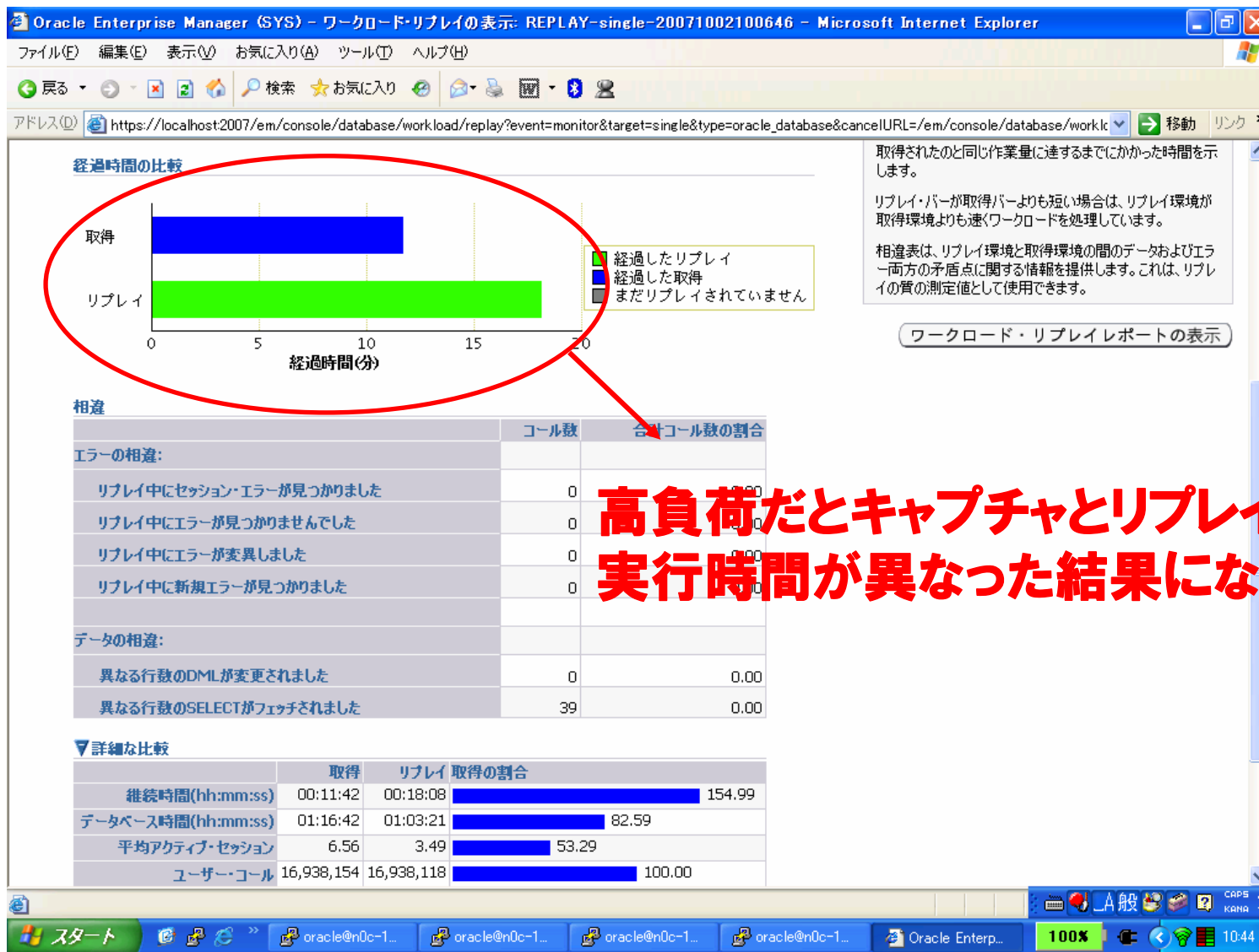
■スループット (TPS)

スレッド数	4	8	12	40
キャプチャ (tps) -①	20.87	41.7	62.11	170.42
リプレイ (tps) -②	21.55	42.82	63.53	115.39
②÷①	1.03	1.03	1.02	0.68



JPetStore JTAを使用

Oracle Enterprise Manager でのワークロード・リプレイの表示

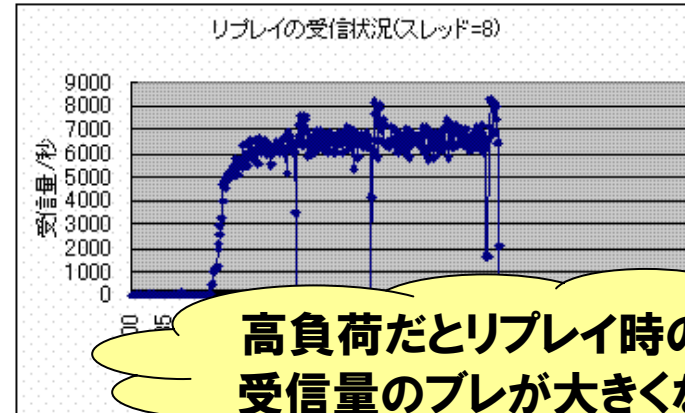
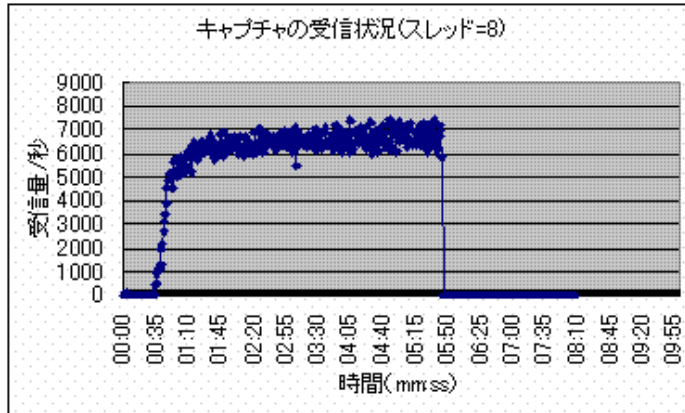


高負荷だとキャプチャとリプレイの実行時間が異なった結果になった

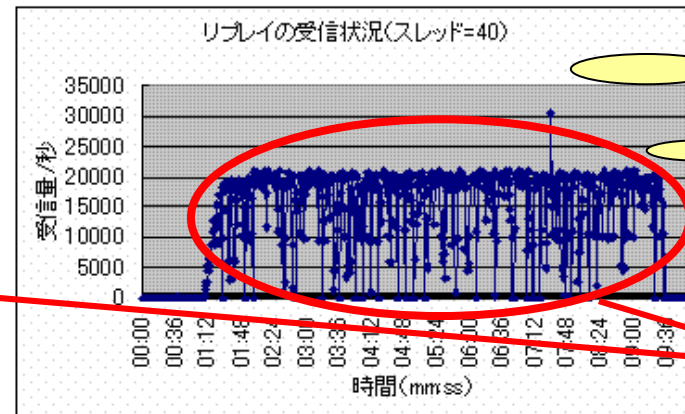
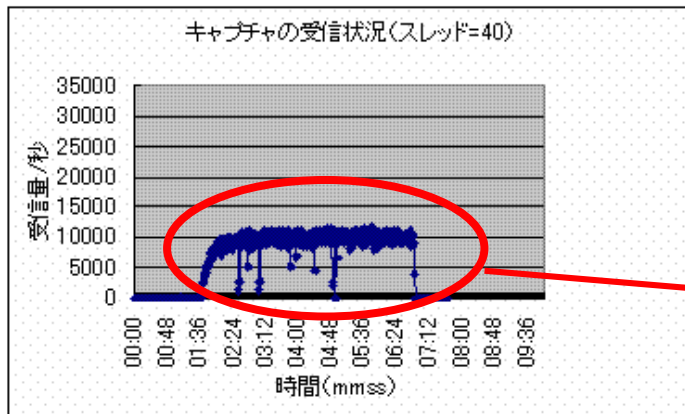
JPetStore JTAを使用

検証結果（詳細）

2. キャプチャ時とリプレイ時でのHWリソース使用量の差分 ～ネットワーク スレッド数8と40の比較～



高負荷だとリプレイ時のNW
受信量のブレが大きくなった

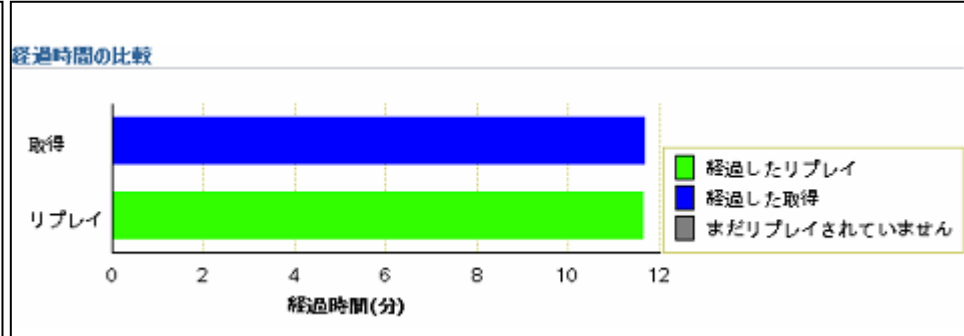
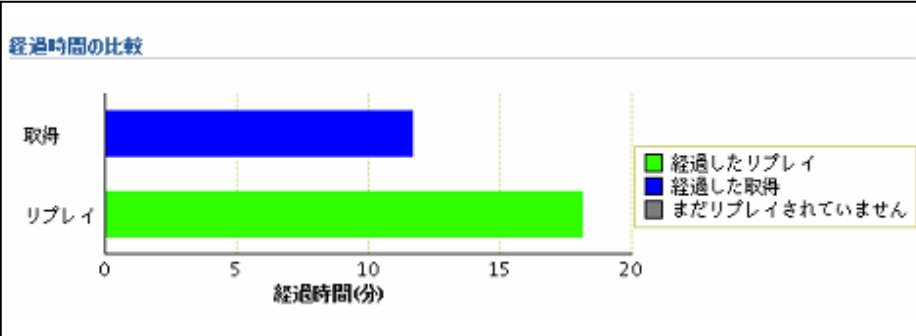


※ CPU使用量、I/O (read/write) 量の差は、ほとんどなし

JPetStore JTAを使用

検証結果（詳細）

2. キャプチャ時とリプレイ時でのHWリソース使用量の差分 ～スループットを再現させるためには～



Synchronization= true

	実行時間	スループット
キャプチャ (①)	11:42	170.42
リプレイ (②)	18:08	115.39
②÷①	1.55	0.68

Synchronization = false

	実行時間	スループット
キャプチャ (①)	11:42	170.42
リプレイ (②)	11:39	179.70
②÷①	0.99	1.05

⇒ Synchronizationパラメータをfalseに設定することで、キャプチャ時と同程度のスループットになる（ただし、トランザクションのコミットの順番は保障されない）

◎ Synchronization とは、キャプチャ時に取得したワークロードのcommitの順序を保持するかどうかを決定するパラメータのこと

JPetStore JTAを使用

Real Application Testing の訴求ポイント ～本検証を実施して～

本検証を通して、以下の2点を訴求できる可能性ありと判断。

1.Database運用(メンテナンス)設計ならびにSIコストの削減
運用設計のフェーズにてHWリプレイスやバージョンアップ等のシステム構成変更があった際、RATの動作特性※を把握した上でテスト方法を設計に盛り込めばよい。
→ 従来のテストAP作成等のコストが大幅に削減される

2.Database運用(メンテナンス)コストの削減

例:某製造系 基幹システムでは、業務に即したテストAPを持っておらず。

課題:・パッチ受け入れ評価に時間がかかる (本例では2ヶ月)

・テスト環境で行ったテスト内容の網羅度が低い

(例えば、新規業務APの動作テストを行うが、既存APのテストは行わない)

OracleDatabase11gへの最新パッチ適用、バージョンアップやHWリプレイスに伴う、事前テストとしてRATを使用することで上記2点の課題を解消できる。

※リプレイ時、以下の例のようにTx特性を変え、再現させることが可能
(詳細については、現在、検証中)

例:1. SQL文の発行順序を反映させたい → Synchronization TRUE (デフォルト)

2. スループットを反映させたい → Synchronization FALSE

今後の展望（～2007/11）

■検証（予定）

•基本検証の継続

例：キャプチャ時、リプレイ時のスループット限界値計測
Replayパラメータによるチューニング
キャプチャファイルのサイズと負荷の関連測定

•現場のニーズを踏まえた検証

例：TDE + Database Replay
スケーラビリティ検証への適用

•複数ワークロード / 複数サービス 環境下での Database Replayを使用した検証

例：NEC HAキットの複数シナリオを使用
NEC HAキット と JPetStore JTA の併用

今後の展望（～2007/12）～続き～

■成果物（予定）

検証ホワイトペーパー

※日本オラクル社 Oracle GridCenter のホームページ

(http://www.oracle.co.jp/solutions/grid_center/) よりダウンロード可能
にできるよう公開予定

■イベント（予定）

12/5（水）～ 7（金） C&Cユーザーフォーラム&iEXPO 2007
（会場：東京ビックサイト）*注

*本検証結果の公開有無は、現在検討中

ORACLE®

Empowered by Innovation

NEC