



THE POSSIBILITIES ARE INFINITE



**富士通 PRIMEQUEST における
Oracle Real Application Clusters / Oracle Application Server
スケーラビリティ検証結果**

2007 年 11 月 22 日
第1.0 版

目次

1. 実施要綱	3
2. 検証環境	3
2-1. ハードウェア	3
2-1-1. 使用ハードウェア	3
2-1-2. 富士通「PRIMEQUEST」のハードウェア機能	4
2-2. ソフトウェア	6
2-3. システム構成全体	7
2-4. サーバ、ストレージ構成	8
3. 検証内容	10
3-1. 検証用プログラム(JPetStore ベンチマーク)解説	10
3-1-1. 使用プログラム	10
3-1-2. データベーススキーマ	11
3-1-3. 処理手順	12
3-1-4. 使用パラメータ	13
4. 検証結果	14
4-1. 検証① - Oracle Real Application Clusters 8 ノード構成	14
4-2. 検証② - Oracle Real Application Clusters 4 ノード構成	15
4-3. チューニングの効果 - Oracle Real Application Clusters 4 ノード構成	16
4-3-1. 検証②未チューニングの結果	16
4-3-2. 検証②チューニング1 索引 PK_LINEITEM の Insert 競合削減	18
4-3-3. 検証②チューニング2 インターコネクト通信処理の最適化	20
4-3-4. チューニング前後の比較 Oracle Real Application Clusters 4 ノード構成	23
5. まとめ	24
6. 付録	25
7. 参考情報	26

1. 実施要綱

規模：10 名（検証環境構築 5 名、検証作業担当 5 名）

場所：富士通株式会社「Platform Solution Center」検証ルーム

<http://triole.fujitsu.com/jp/psc/>

検証者および担当：

富士通株式会社（ハードウェアおよび OS 環境の提供、及び検証実施）

日本オラクル株式会社（アプリケーションサーバおよびデータベースソフトウェアの提供、及び検証実施）

2. 検証環境

2-1. ハードウェア

2-1-1. 使用ハードウェア

今回の検証で使用したハードウェアは以下のとおりです。

【データベースサーバ】

◆富士通 PRIMEQUEST580

・CPU: Intel Dual-Core Itanium2 Processor 9050 (1.6GHz 24MB) × 32CPU (64Core)

・メモリ: 64GB

・I/O:

・LAN: 1000BASE-T × 32 ポート、100Base-TX × 16 ポート)

(業務 LAN × 16 ポート、クラスタインターコネクト × 16 ポート、管理 LAN × 16 ポート)

・HBA: 4Gbps FC カード × 16 枚

・147GB 内蔵 HDD × 24 本

[1 パーティション当たり] ※8 パーティション構成、詳細は後述。

4CPU (8CORE), 8GB メモリ, 147GB HDD × 3 本 (システムボリューム [RAID1] + DUMP 領域),

FC カード × 2 枚 (冗長化), 業務 LAN × 2 (冗長化), クラスタインターコネクト × 2 (冗長化),

管理 LAN × 2 (冗長化)

【ストレージ】

◆富士通 ETERNUS8000 model900 × 1 台

・4CM (4CA/CM, 2Ports/CA)

・146GB HDD × 240 本

・DATAFILE 用 RAID0+1(8+8) × 4LUN_R

・REDO log 用 RAID0+1(8+8) × 8LUN_R

・ARCHIVE Log 用 RAID0+1(8+8) × 2LUN_R

・OCR 用 RAID0+1(8+8) × 1LUN_R

【SAN スイッチ】

- ◆富士通 ETERNUS SN200 model430 ×2 台
- ・4Gbps FC ポート×16

【アプリケーションサーバ】

- ◆富士通 PRIMERGY BX620 S3 ×16 台 (Blade Server)
- ・CPU: Intel Dual-Core Xeon Processor 5080(3.7GHz)
- ・メモリ: 2GB
- ・HDD: 73GB×2 (RAID1)

【クライアント(負荷生成ツール実行環境)】

- ◆富士通 PRIMERGY RX200 S2 ×16 台
- ◆富士通 PRIMERGY RX300 S2 ×12 台
- ◆富士通 PRIMERGY RX200 S3 ×4 台

【ネットワークスイッチ】

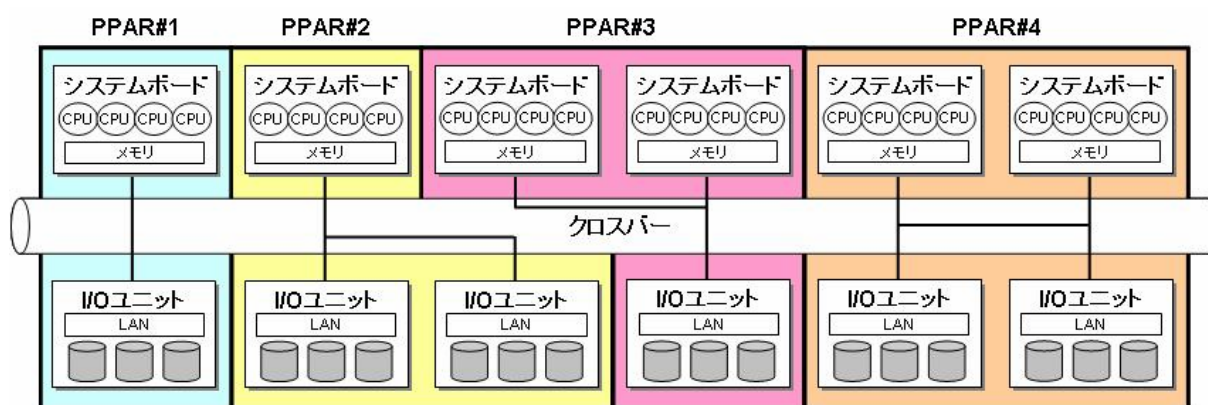
- ◆Cisco Catalyst3560G(24Ports/48Ports)

2-1-2. 富士通「PRIMEQUEST」のハードウェア機能

データベースサーバとして使用した富士通「PRIMEQUEST580」のハードウェア機能の内、今回使用した機能について説明します。

●物理パーティショニング (PPAR)

1 筐体内を複数のパーティション(区画)に物理的に分割できる機能です。システムボード (CPU/メモリを搭載するボード) と I/O ユニット (内蔵 HDD/LAN/FC カード等を搭載するユニット) を組み合わせることで、大規模パーティションや小規模パーティションを自由に構成することが可能です。



※クロスバー: システムボードと I/O ユニットを結ぶ超高速システムバス

図 2-1. 物理パーティショニング (PPAR) の構成例

本機能により PRIMEQUEST 1 筐体内に完全に独立した複数のサーバ環境を構成することが出来ます。各 PPAR では Linux や Windows を稼働させることができ、1 筐体で複数のサーバ環境を構築することができます。

本機能は論理的なパーティショニングと比較し、処理オーバーヘッドが発生しません。また障害隔離性に優れており、1つのパーティションで万一障害が発生した場合でも他パーティションには一切影響を与えません。

ミッションクリティカルな用途で使用する大規模 Oracle Real Application Clusters 環境では、処理能力や隔離障害性が特に重視されることを想定し、今回の検証環境において本機能を採用いたしました。

なお、PRIMEQUEST ではパーティション粒度を更に増す機構として、拡張パーティショニング (XPAR) や VM 機能を選択することも可能です。

●MMB (マネージメントボード)

PRIMEQUEST のシステム全体を管理するコンポーネントです。PRIMEQUEST では MMB を筐体内に内蔵しており、システム管理機能のすべてを集約している為、専用の管理サーバを外部に設置する必要がありません。

またシステム管理を行うための各種操作・設定は、MMB の Web-UI (ウェブユーザーインターフェース) を使用する為、汎用 PC の Web ブラウザからすべての運用・操作を行うことができます。

以下、MMB の主な提供機能です。

- ・パーティション設定/構成表示

システムボードや I/O ユニットの追加/削除設定を行うことで、柔軟なハードウェアパーティショニングを実現します。

- ・各パーティション管理

設定された各パーティションの電源 ON/OFF、リセット運用や、ハードウェア監視を行います。

- ・コンソールリダイレクション

各パーティションへのコンソール情報の入出力先を管理 LAN 経由で行うことが出来ます。

2-2. ソフトウェア

今回の検証で使用したソフトウェアは以下のとおりです。

【データベースサーバ】

- ◆Red Hat Enterprise Linux AS4 for Itanium (Update4)
- ◆Oracle Database 10g Enterprise Edition Release2 (10.2.0.3)
- ◆Oracle Real Application Clusters R10.2.0.3
- ◆PRIMECLUSTER Enterprise Edition 4.2A00
- ◆PRIMECLUSTER GLS 4.2A00
- ◆PRIMECLUSTER GDS 4.2A00
- ◆ETERNUS マルチパスドライバ

【アプリケーションサーバ】

- ◆Red Hat Enterprise Linux AS4 for x86 (Update3)
- ◆Oracle Application Server 10g Release2 (10.1.2)
- ◆JPetStore
- ◆Systemwalker Resource Coordinator V13

【クライアント】

- ◆Red Hat Enterprise Linux AS4 for x86 (Update3)
- ◆JMeter

2-3. システム構成全体

今回使用したシステムの概要を下图に示します。

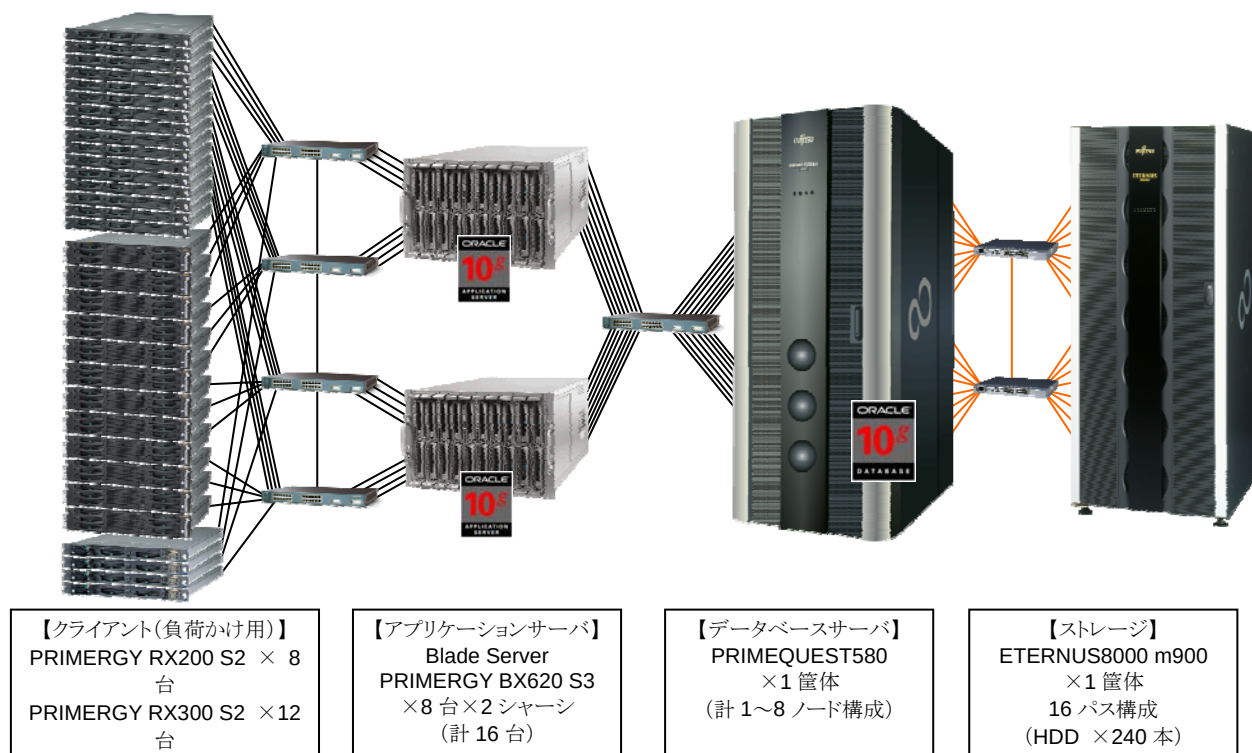


図 2-2. システム構成概略図

【データベースサーバ/ストレージ】

データベースサーバは PRIMEQUEST580 上に構成した物理パーティション(以下 PPAR)で構成しています。システムボード 1 台と I/O ユニット 1 台の組み合わせを 1 パーティション(PPAR)とし、合計 8 パーティション(8 ノード)の構成としました。これにより 8 ノードの Oracle Real Application Clusters 環境を構築しました。

また、構成を変更しシステムボード 2 台(8CPU,16MB メモリ)と I/O ユニット 1 台を 1PPAR とする 4 ノード構成における検証も実施いたしました。

アプリケーションサーバ側のネットワークは各パーティション(ノード)からギガビットイーサネットで結び、PRIMECLUSTER GLS を使用して全て二重化を施しています。

【アプリケーションサーバ】

計 16 枚のブレードサーバを使用し、Oracle Application Server 10g + JPetStore を稼動させました。データベースサーバ通信用、クライアント機通信用として全てギガビットイーサネットで結びました。

【クライアント(負荷生成ツール実行環境)】

PRIMERGY を 32 台使用し、負荷生成ツール JMeter の実行環境としました。ネットワークはフラット構成で各アプリケーションサーバへ均等に負荷がかかる構成としました。

2-4. サーバ、ストレージ構成

データベースサーバの1ノード当りの構成は4CPU,8GBメモリとしました。システムボリュームは147GB内蔵HDD×3本で構成し、Oracleで使用するDBファイルやREDO Logファイルはストレージ上に作成しました。

サーバとストレージ間のFC接続について、PRIMEQUESTの各ノードから2パス構成とし8Gの帯域を確保しました。またETERNUS8000側は各コントローラに均等配分することで、負荷分散を図っています。

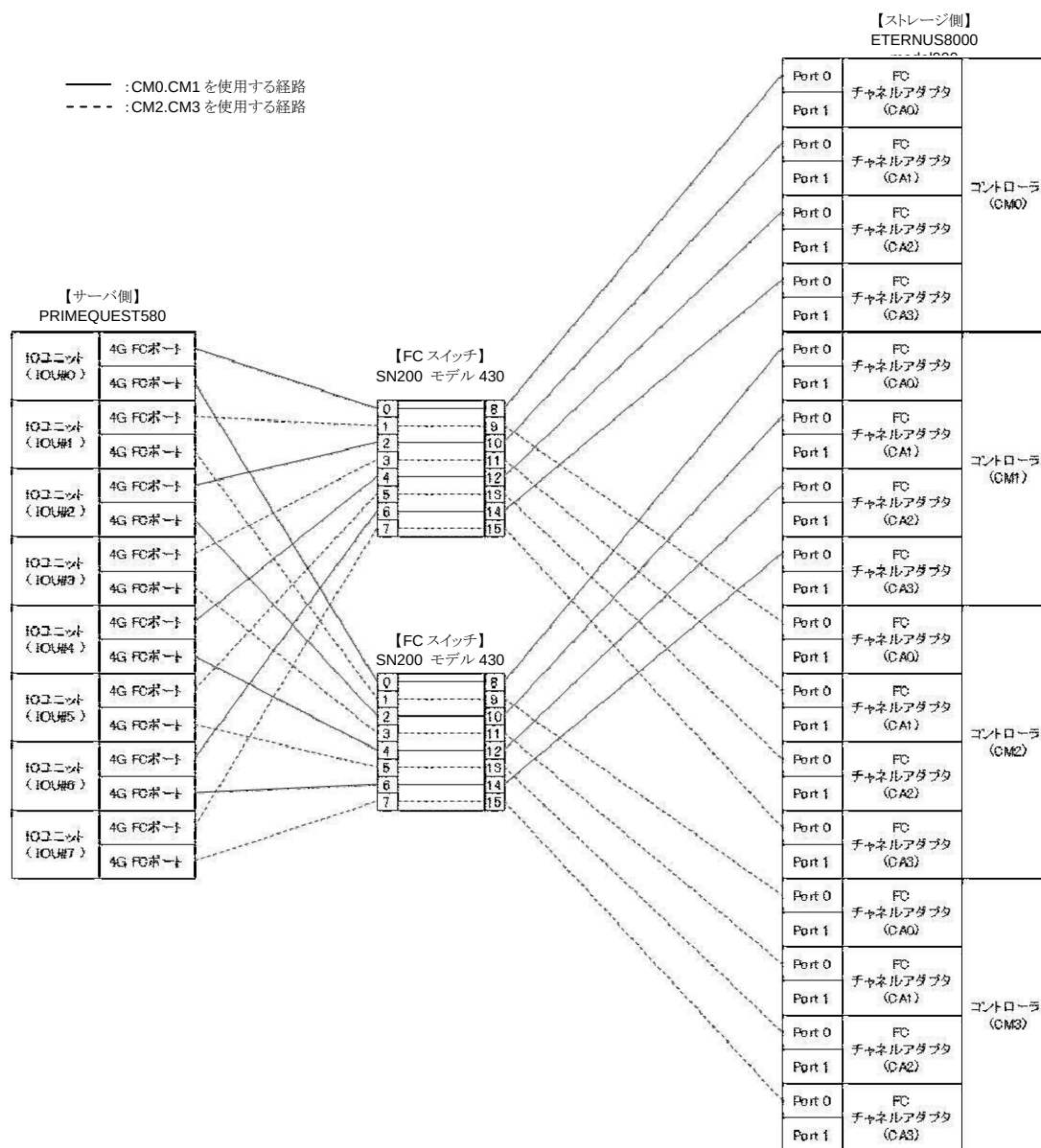
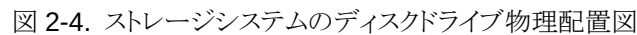


図 2-3. サーバとストレージ間の FC 接続構成図

なお、図 2-4.中、特に指定のないハードディスクドライブについては本試験で利用していない領域です。



3. 検証内容

3-1. 検証用プログラム(JPetStore ベンチマーク)解説

3-1-1. 使用プログラム

今回の検証では http 経由にてアプリケーションサーバおよびデータベースサーバに対し実行するベンチマークプログラムとして、3tier テストのアプリケーションである JPetStore を日本オラクル(株)にて改変したプログラムを使用しました。

JPetStore はオープンソースで提供されている J2EE アプリケーションフレームワークである Spring Framework(<http://www.springframework.org/>) のサンプルプログラムとして提供されており、Sun Microsystems 社より J2EE ブループリントとして提供される Java Pet Store を元に Spring Framework を適用して再構築されたアプリケーションです。

今回利用した JPetStore は Spring Framework 1.2.8 に含まれるサンプルプログラムで、検証に当たって実施した改変は次の 4 点です。

- build.xml の修正

Apache Ant 用 build.xml ファイルに対して ear ファイルを生成するように修正しました。

spring-framework-1.2.8/samples/petsotre/build.xml

- 明示的な setAttribute の追記

オリジナルのソースでは明示的に setAttribute が行われていないものがあります。これはシングル構成のアプリケーションサーバでは問題なく動作しても、クラスタ構成のアプリケーションサーバにおいては期待される動作が得られない可能性があります。

このためセッションレプリケートを設定したクラスタ構成環境において modifiedAttribute スコープ属性を使用している場合にはセッション属性を変更しても setAttribute が呼び出されないため、変更が他ノードに伝播されない現象が発生します。このため以下のソースに対し setAttribute を明示的に実施する変更を加えました。

spring-framework-1.2.8/samples/petstore/src/org/springframework/samples/jpetstore/web/spring/AddItemToCartController.java

spring-framework-1.2.8/samples/petstore/src/org/springframework/samples/jpetstore/web/spring/RemoveItemFromCartController.java

- 商品検索条件変更

オリジナルのソースでは商品検索の条件が SQL において「LIKE '%keyword%」という形式の問い合わせを実施しています。この SQL 文の場合 Oracle Database では索引スキャンではなく全表走査が行われるため効率的な検索が行えません。Oracle Database を実システムで利用しこの様な形式の検索を行う場合には Oracle TEXT などの全文検索を行うための機能を利用します。ただし今回のテストに使用したアプリケーションでは単純に「LIKE 'keyword%」形式で前方一致への書き換えを行っています。

spring-framework-1.2.8/samples/petstore/src/org/springframework/samples/jpetstore/dao/ibatis/sqlMapProductDao.java

また比較条件が「lower(name) like 'keyword%」と大文字小文字を区別しない実装となっているため lower 関数の結果に対するファンクション索引を作成しています。

・商品検索 SQL の OR 条件への対応

オリジナルのソースでは商品検索の SQL は PRODUCT 表の商品名・カテゴリ名・商品説明の列に対するキーワード検索を行っており、具体的には次の SQL が発行されています。

```
select * from product where lower(name) like '1%' or lower(category) like '2%' or  
lower(descn) like '3%'
```

OR 条件が並んでいる場合、その列に索引が張られていても全表走査が実行されることがあります。Oracle Database ではこのような場合、OR 条件ではなく UNION ALL を使用することが推奨され、今回のテストに使用したアプリケーションでは商品名のみを検索で動作する様にしたためカテゴリ名と商品説明に対する検索は行わせない対応をとっています。

これらの変更を加えた JPetStore をこの文書内において以下 JPetStore ベンチマークと呼びます。

3-1-2. データベーススキーマ

テーブル名	件数	概要
ACCOUNT	1,000,000	アカウント表。ユーザー情報を格納
SIGNON	1,000,000	ユーザーID とパスワードを格納
PROFILE	1,000,000	プロフィール表。ユーザーの設定情報を格納
PRODUCT	50,000,000	商品表。商品 ID、商品名、概要等を格納
ITEM	500,000,000	商品 ID に紐づいた項目で、注文で選択されるもの。 アイテム ID 等を格納
INVENTORY	500,000,000	在庫管理表。商品アイテム ID に紐づいて、それぞれの アイテムの在庫数を格納

テーブル名	索引名	概要
ACCOUNT	PK_ACCOUNT	主キー用索引
SIGNON	PK_SIGNON	主キー用索引
PROFILE	PK_PROFILE	主キー用索引
PRODUCT	PK_PRODUCT	主キー用索引
	PRODUCT_NAME	商品名(商品検索用)
	PRODUCT_DESCN	商品概要(商品検索用)
	PRODUCT_CATEGORY	商品カテゴリ(商品検索用)
ITEM	PK_ITEM	主キー用索引
	ITEMPROD	PRODUCTID 列
INVENTORY	PK_INVENTORY	主キー用索引

3-1-3. 処理手順

トランザクション

JPetStore ベンチマークの性能結果は単位時間でどれだけの処理が行えるか、すなわちスループットとして示されます。本検証ではスループットを Transactions Per Second(TPS)つまり毎秒あたりの処理量として定義します。JPetStore ベンチマークでのトランザクションは各 web (http) リクエストに対しての応答をトランザクションと定義しており、一般の web アプリケーションにおける 1 Page View が JPetStore ベンチマークでの 1 トランザクションに相当します。

またトランザクションによってはデータベースにアクセスするものとししないものが存在します。

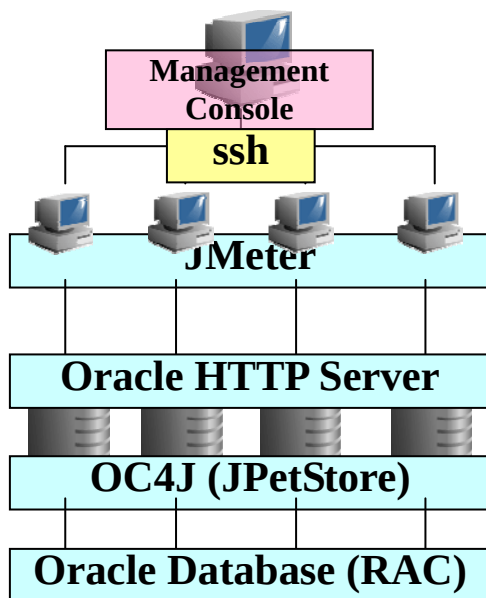
・トランザクション一覧

トランザクション名	説明	DB アクセス
1. TopPage		
	トップページの表示	無し
2. SignInPage		
	サインインページ	無し
3. SignIn		
	サインイン実行	有り
4. 商品検索、カートへの追加		
4.1 Search	Category 指定	有り
4.2 Product	Product 指定	有り
4.3 Item	Item 指定	有り
4.4 AddCart	商品のカートへの追加	無し
5. Proceedd to Checkout	チェックアウトに進む	無し
6. Continue to Checkout	チェックアウトを続ける	無し
7. Submit	チェックアウトを決定する	有り
8. Confirm	購入を確認する	無し
9. SignOut	サインアウト	無し

・JPetStore ベンチマークアクセスシナリオ

アクセスシナリオ	
1. TopPage	
2. SignInPage	
3. SignIn	
4. 商品検索、カートへの追加	
4.1 Search - 100 レコードがマッチ	1～5 回繰り返し (平均カートに 3 アイテム)
4.2 Product	
4.3 Item	
4.4 AddCart	
5. Proceedd to Checkout	
6. Continue to Checkout	
7. Submit	
8. Confirm	
9. SignOut	

・テスト環境および計測シナリオ



負荷生成における各 tier の役割は次のとおりです。

Management Consolessh を通し各クライアントに配置された JMeter に対して前述のアクセスシナリオに従い、JPetStore の画面遷移を実行します。

また management console では JMeter のログ・全層の OS 統計情報・Oracle Database の statspack が収集されます。

1 つの JMeter クライアントは 1 つのアプリケーションサーバにのみアクセスを行い、各 JMeter クライアントは Non-GUI モードにて起動され http 経由での負荷を 10 分間生成します。

アプリケーションサーバ層のノードにおいてはノードあたり 1 つの Oracle HTTP Server インスタンスと 1 つの OC4J インスタンスを起動します。

Oracle Database インスタンス・Oracle HTTP Server・OC4J インスタンスはすべて計測ごとに再起動を実行します。

・TPS の算出

テストにあたっては JMeter からのベンチマーク実行時間をトータル 360 秒とし、実行開始 60 秒後よりすべての JMeter が動作を完了した時間の 30 秒前までの値を測定値として用いています。

・統計情報

データベース、アプリケーションサーバおよびクライアントマシンにおいて CPU と I/O の統計情報を取得しており、これらの OS 統計より本検証においては I/O ネックは発生していない事が確認できています。

またデータベースに関しては Oracle8i Database 以降で標準的に含まれているデータベース統計取得ツールである statspack を用いています。

3-1-4. 使用パラメータ

データベースおよびアプリケーションサーバに対する負荷の変更は JMeter より起動される thread 数を変更して行っており、各 thread が実システムにおける個々のユーザーが利用するブラウザをシミュレートしています。本ホワイトペーパーで記載する thread 数は全クライアントから同時に実行した総数で示しています。

本検証ではデータベースを 8 ノードおよび 4 ノードとして構成を行っていますが、どちらのケースでも利用した総 CPU 数は同一としています。そのため 4 ノード構成時におけるノードあたりの CPU 数は 8 ノード構成時の倍となります。

JMeter の thread 数はデータベース ノードの CPU 数と比例させているため、構成が異なる場合でもデータベースにおける CPU 数が同一であれば負荷を与えた thread 総数は同一となります。

JPetStore ベンチマークにおけるページ遷移間の ThinkTime はすべて 500 ミリ秒として検証を実施しました。

4. 検証結果

4-1. 検証① - Oracle Real Application Clusters 8 ノード構成

この検証では、3 階層であるシステム構成のデータベースサーバ層/アプリケーションサーバ層/クライアント層の各構成台数比を 1: 2: 4 の固定とし、台数を比例させて増加する事で、Oracle Real Application Clusters、Oracle Application Server 両者のスケーラビリティ(スケールアウト)を評価しています。

クライアント 1 台あたりの同時接続数は 110~150 に変動させてスループットを計測しています。

PRIMEQUEST 580 のパーティション機能を利用する事で、合計 32CPU を 8 等分し、1 パーティション(=1 ノード)辺り 4CPU とし、データベースサーバ層を Oracle Real Application Clusters 8 ノード構成としています。

この結果については特にチューニングを施していない結果となります。

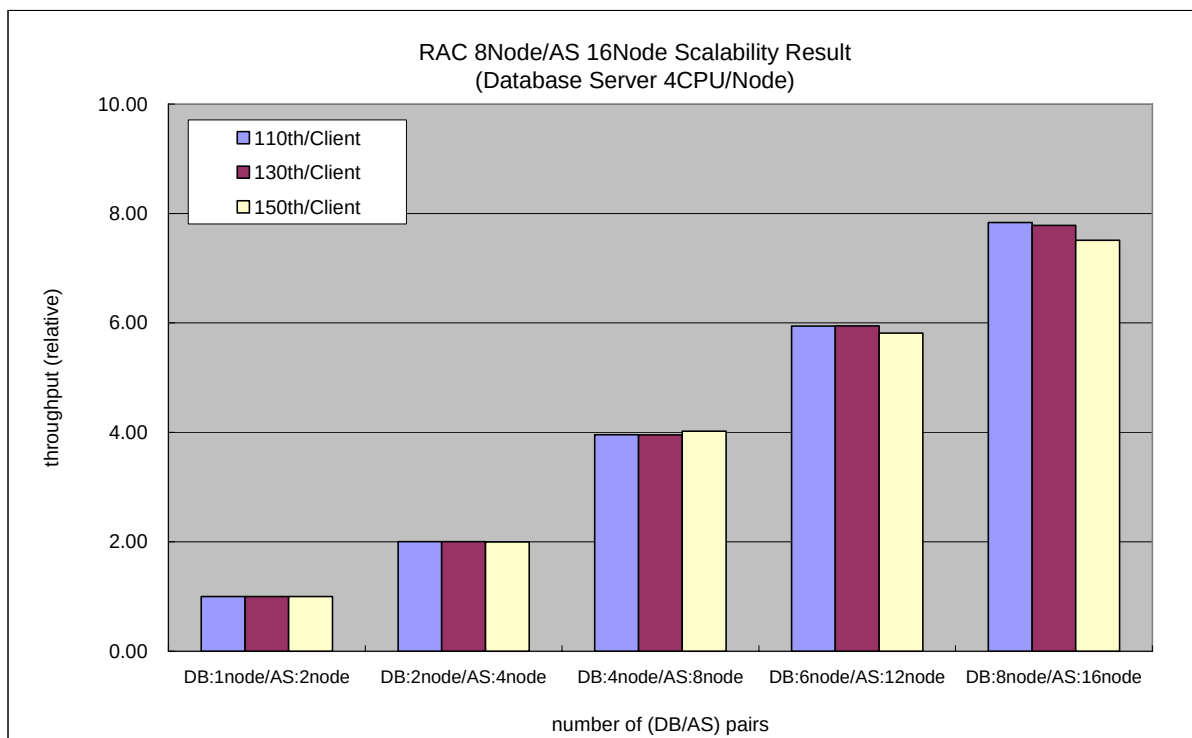


図 4-1 検証 DB 層 8 ノード構成時のデータベースサーバ/アプリケーションサーバ スケーラビリティ測定結果
未チューニング

図 4-1 は、各同時接続数におけるデータベースサーバ 1 ノード(4CPU)/アプリケーションサーバ 2 ノード構成時のスループットを 1.00 とし、データベースサーバ/アプリケーションサーバの構成台数を増加させていった際のスループット比率を示しています。縦軸はスループット比率、横軸はデータベースサーバ/アプリケーションサーバの各構成台数を示しています。

データベースサーバ 1 ノード/アプリケーションサーバ 2 ノード構成に対しノード数を 8 倍(データベースサーバ 8 ノード /アプリケーションサーバ 16 ノード)とした際のスループット比率は同時接続数 110 の場合で 7.84 倍、同時接続数 150 の場合で 7.51 倍となっており、各層の構成台数と性能が完全に正比例した場合と比べて約 98%、94%というスループットが出ています。

4-2. 検証② - Oracle Real Application Clusters 4 ノード構成

この検証では、3 階層であるシステム構成のデータベースサーバ層/アプリケーションサーバ層/クライアント層 の各構成台数比を 1: 4: 8 の固定とし、台数を比例させて増加する事で、Oracle Real Application Clusters、Oracle Application Server両者のスケーラビリティ(スケールアウト)を評価しています。

クライアント 1 台あたりの同時接続数は 110～150 に変動させてスループットを計測しています。

PRIMEQUEST 580のパーティション機能を利用する事で、合計 32CPUを 4 等分し、1 パーティション(=1 ノード)辺り 8CPUとし、データベースサーバ層をOracle Real Application Clusters 4 ノード構成としています。

合計で利用するハードウェア資源としては「4-1. 検証 」の 8 ノード構成と同等となります。

この結果については後述するチューニングを施した結果となります。

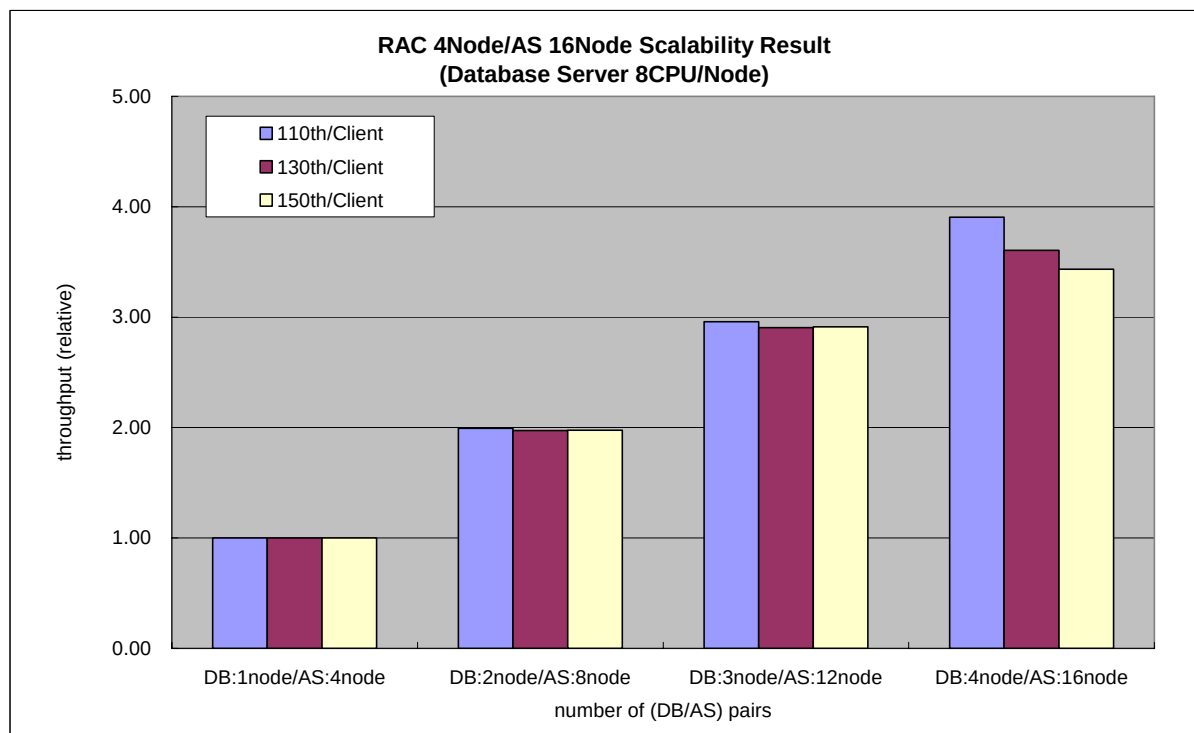


図 4-2 検証② DB 層 4 ノード構成時のデータベースサーバ/アプリケーションサーバ スケーラビリティ測定結果
チューニング済

図 4-2 は、各同時接続数におけるデータベースサーバ 1 ノード(8CPU)/アプリケーションサーバ 4 ノード構成時のスループットを 1.00 とし、データベースサーバ/アプリケーションサーバの構成台数を増加させていった際のスループット比率を示しています。縦軸はスループット比率、横軸は データベースサーバ/アプリケーションサーバの各構成台数を示しています。

データベースサーバ 1 ノード/アプリケーションサーバ 4 ノード構成に対しノード数を 4 倍(データベースサーバ 4 ノード /アプリケーションサーバ 16 ノード)とした際のスループット比率は、同時接続数 110 の場合で 3.90 倍、同時接続数 150 の場合で 3.43 倍となっており、各層の構成台数と性能が完全に正比例した場合と比べて約 97%、86%というスループットが出ています。

4-3. チューニングの効果 - Oracle Real Application Clusters 4 ノード構成

「4-2. 検証②」では、データベースサーバ層に対するチューニングを施した結果を掲載しています。

ここでは、性能改善のために段階的に実施したチューニング内容とその効果について詳細を説明します。

4-3-1. 検証②未チューニングの結果

検証②では、PRIMEQUEST 580 のパーティション機能を利用する事で合計 32CPUを 4 等分し、1 パーティション(=1 ノード)辺り 8CPUとして、データベースサーバ層をOracle Real Application Clusters 4 ノード構成としています。

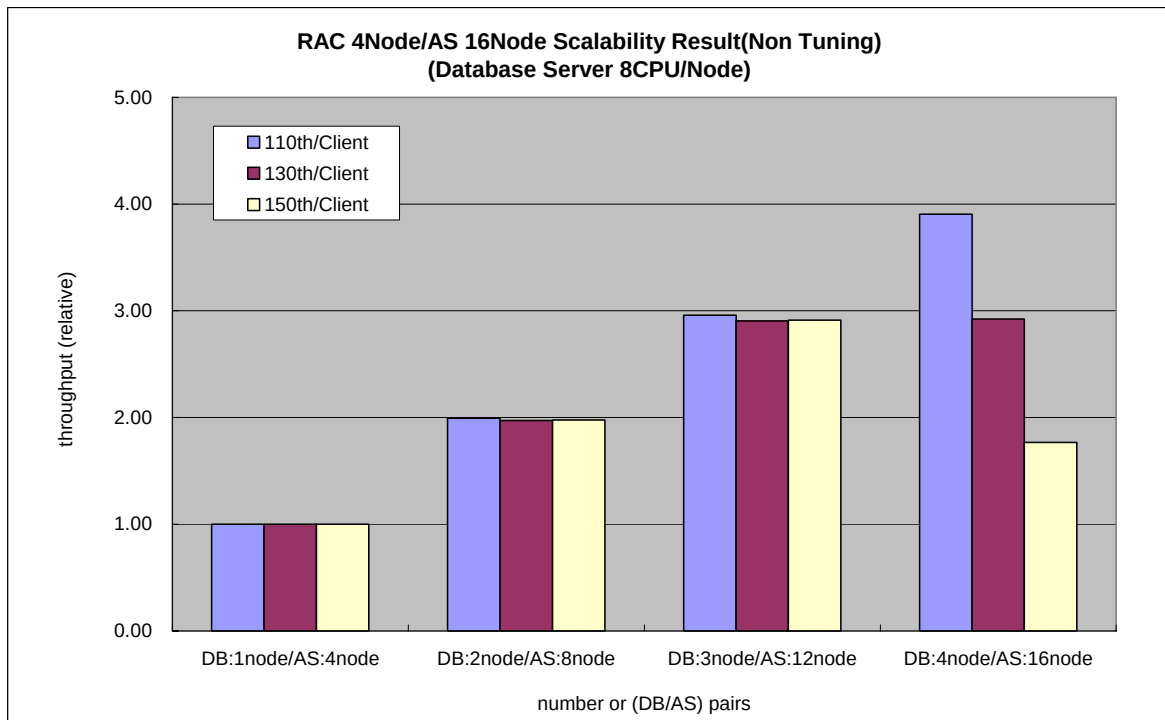


図 4-3 検証② DB 層 4 ノード構成時のデータベースサーバ/アプリケーションサーバ スケーラビリティ測定結果
未チューニング

図 4-3 は、各同時接続数におけるデータベースサーバ 1 ノード(8CPU)/アプリケーションサーバ 4 ノード構成時のスループットを 1.00 とし、データベースサーバ/アプリケーションサーバの構成台数を増加させていった際のスループット比率を示しています。横軸はデータベースサーバ/アプリケーションサーバの各構成台数、縦軸はスループット比率を示しています。

図 4-3 よりクライアント 1 台あたりの同時接続数が 110 と少ない場合には、データベースサーバ/アプリケーションサーバの各ノード数に比例したスループットを得られているのに対して、同時接続数を 130、150 と増加させるに従い、データベースサーバ/アプリケーションサーバの各ノード数とスループットが比例なくなっています。

検証 の場合は、同時接続数が増加しても良好なスループット比が得られていました。

検証 と の差異は、データベースサーバ 1 ノードあたりの PRIMEQUEST パーティション構成変更のみであり、アプリケーションサーバ/クライアントの構成・設定は変更していません。そのため、三階層のうちデータベースサーバ部分がボトルネックになっていると予測できます。

Oracle Databaseの性能については、STATSPACKでレポートされる待機イベント情報より詳細を分析可能です。以下、Oracle Databaseの待機イベントの内訳を示します。

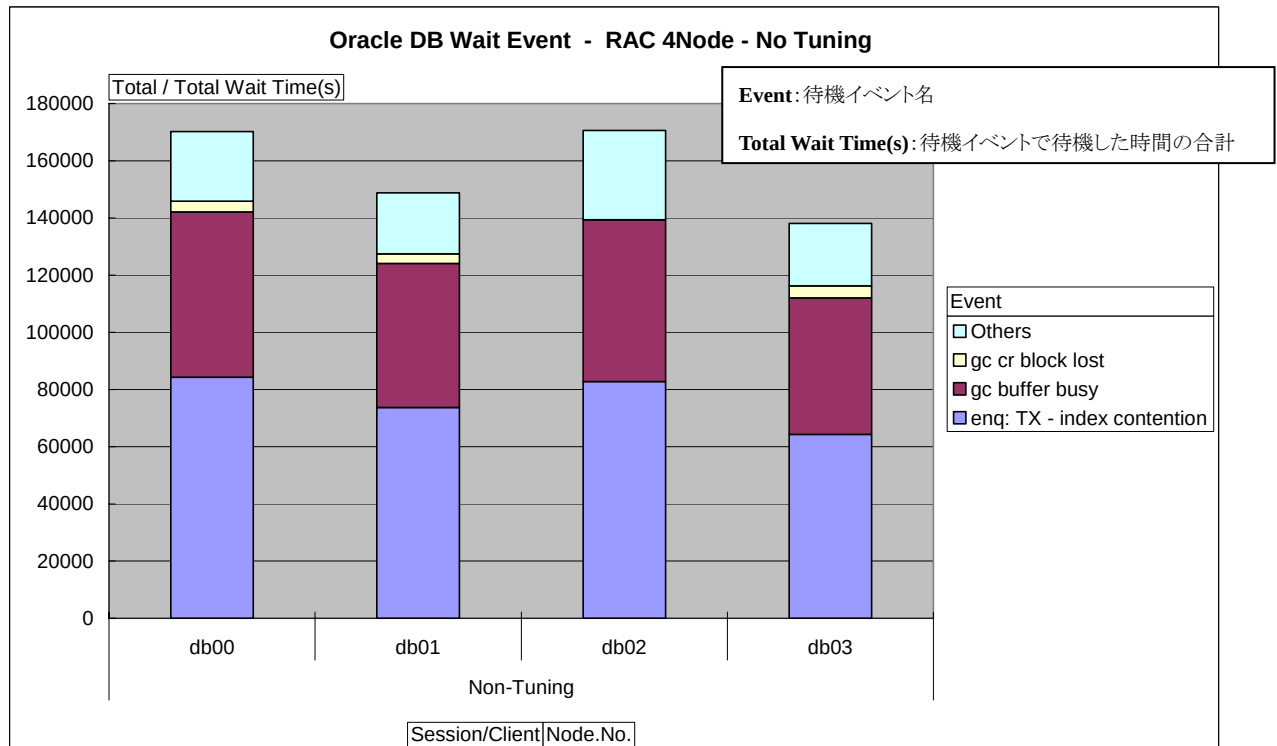


図 4-4 DB 層 4 ノード構成時のデータベースサーバ/アプリケーションサーバスケーラビリティ測定時の Oracle 待機イベント [同時接続数 150 の場合]

図 4-4 は、各層の台数増加に対するスループット比率の落ち込みが最も顕著な、同時接続数 150 の場合におけるノード毎の Oracle Database 待機イベント(Total Wait Time(s))です。

横軸はデータベースサーバのノード名、縦軸は各待機イベントで待機した時間の合計秒数を示しています。

全待機イベントのうち、「enq:TX-index contention」「gc buffer busy」が高い割合を占めている事が分かります。Oracle Database 全体の処理の中で注文情報を管理する表のうち Lineltem 表に付与している索引 PK_LINEITEM への Insert 処理についての競合が「enq:TX-index contention」待機イベントに非常に大きく影響している事が確認できました。

また、今回の試験ではベンチマークアプリケーション、利用データについては、特にReal Application Clusters固有の設計も行わずに、極力そのまま利用して試験を実施しています。今回利用したアプリケーションではノード数(CPU数)の増加にともない、単位時間あたりの処理量が増加する特性を持っており、同じタイミングに同一のデータベースブロックをリクエストする機会も上がるため「gc buffer busy」が増える事が確認できました。

以上から、全体的なチューニングの指針として、以下の 2 点を実施しました。

- チューニング 1 : 索引 PKLINEITEM への Insert 競合待機時間の削減
- チューニング 2 : インターコネクト通信処理の最適化

4-3-2. 検証②チューニング1 索引 PK_LINEITEM の Insert 競合削減

索引 PK_LINEITEM への Insert 競合待機時間対策として以下 2 点のチューニングを実施しました。

- (a) 同時実行トランザクション・エントリの初期値 (INITRANS) を 2 から 255 に変更
- (b) 索引を格納している領域のデータベースのブロック・サイズを 8KB から 2KB に変更

索引 PK_LINEITEM に対する同時実行トランザクション・エントリの範囲はデフォルト値を採用しており、初期値 (INITRANS) は 2、最大値 (MAXTRANS) は 255 です。索引 PK_LINEITEM が格納されている各ブロックへのトランザクションリストに十分な領域がなく ITL 競合が発生していると推測し、その対処とトランザクション・エントリを動的に割り当てる際のオーバーヘッド回避を期待し、初期値を 255 (最大値) に変更としました。

さらに索引 PL_LINEITEM を格納している領域のデータベースのブロック・サイズが 8KB であり、本試験においては、このサイズはブロック競合への影響が大きいと判断し、競合削減を期待し 2KB に変更としました。

以上の 2 点のチューニングを実施し、索引を再構成した後に測定した結果が以下、図 4-5 となります。

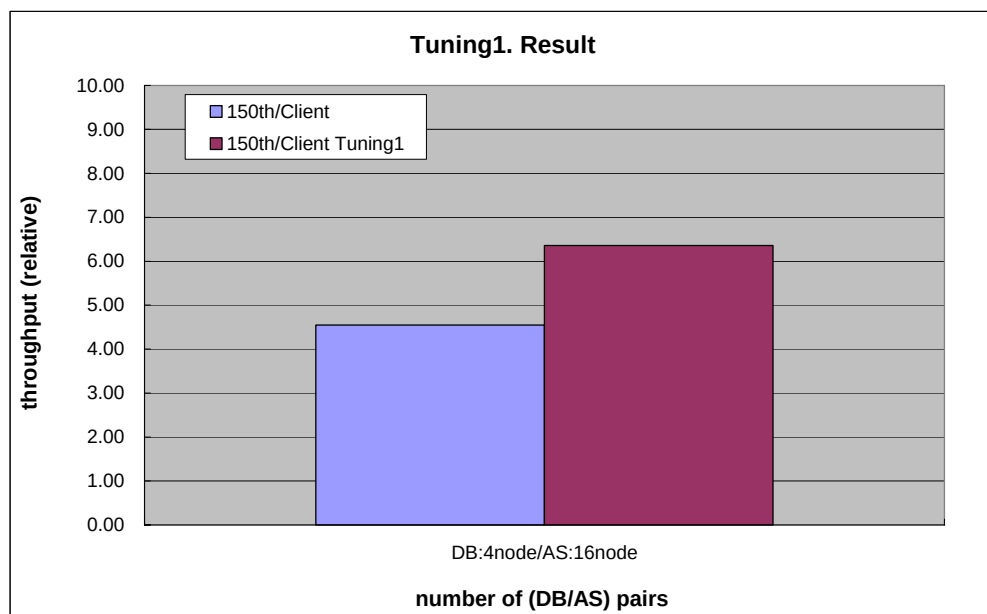


図 4-5 Tuning1. 実施前後のスループット比較

チューニング 1 によりスループット比が 4.55 から 6.36 へ増加しましたが、この数字は 4 ノード構成時のスケールビリティとしてはまだ不足と言えます (期待値は 8~9)。この際の Oracle 待機イベントの内訳が図 4-6 のようになっています。

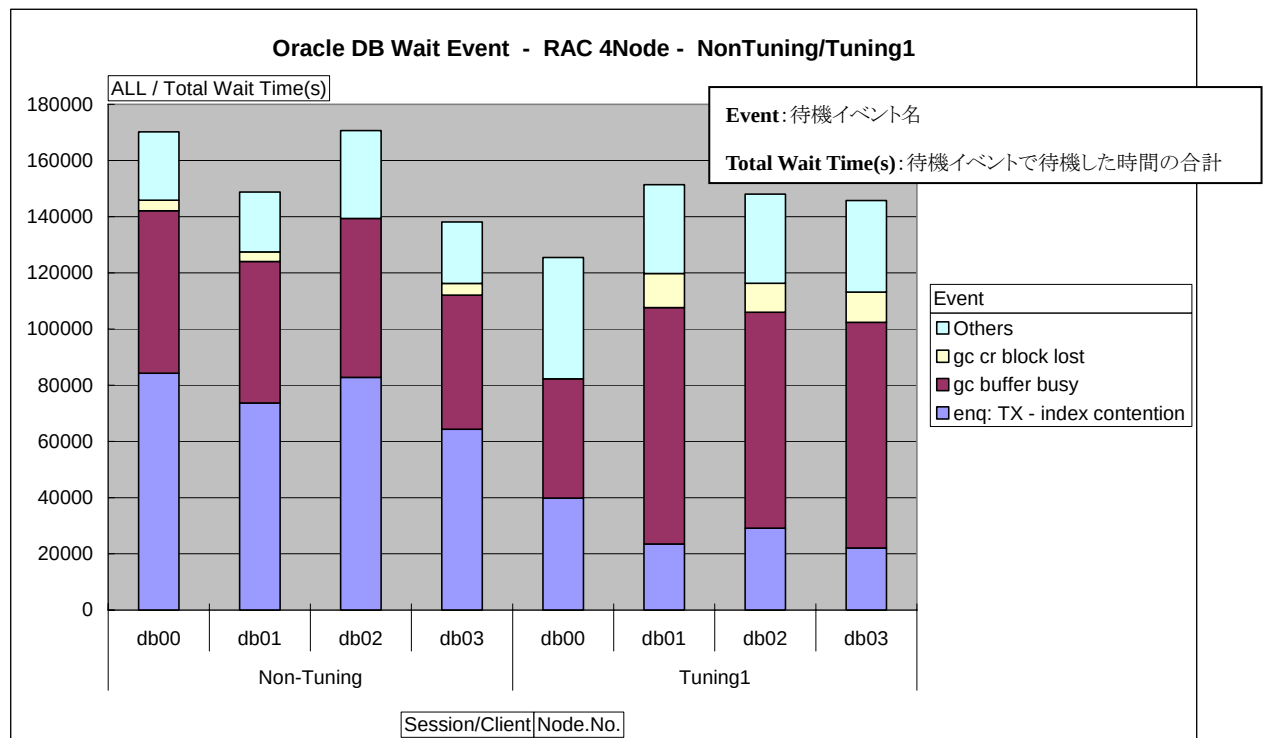


図 4-6 Tuning1. 前後の Oracle 待機イベント比較

図 4-6 はチューニング 1 実施前後の Oracle 待機イベント上位の比較です。

横軸は未チューニング/チューニング1実施後のデータベースサーバのノード名、縦軸は各待機イベントで待機した時間の合計秒数を示しています。

また表 4-1 は同試験時の待機イベント「enq:TX-index contention」「gc buffer busy」「gc cr block lost」「合計秒数(ALL)」の各詳細値を示しています。

表 4-1 Tuning1 結果 Oracle 待機イベント詳細

Tuning No.	ノード No.	enq: TX-index contention		gc buffer busy		gc cr block lost		ALL
		Total Wait Time(s)	%Wait Time	Total Wait Time(s)	%Wait Time	Total Wait Time(s)	%Wait Time	Total Wait Time(s)
Non-Tuning	db00	84,270s	50%	57,826s	34%	3,757s	2%	170,205s
	db01	57,826s	34%	50,390s	34%	3,388s	2%	159,315s
	db02	82,771s	49%	56,579s	33%	4,162s	3%	170,633s
	db03	64,307s	47%	47,770s	35%	4,162s	3%	138,094s
	Ave.	72,294s	45%	53,141s	34%	3,867s	3%	159,562s
Tuning1	db00	39,865s	32%	42,482s	34%	5,908s	5%	125,505s
	db01	23,456s	15%	84,199s	56%	12,123s	8%	151,390s
	db02	29,154s	20%	76,824s	52%	10,298s	7%	148,048s
	db03	22,096s	15%	80,299s	55%	10,736s	7%	145,713s
	Ave.	28,643s	21%	70,951s	49%	9,766s	7%	142,664s

チューニング 1 実施前後で「enq:TX-index contention」の各ノード平均値が 72,294 秒から 28,643 秒と大幅に削減されていますが、全待機イベント合計 (ALL) としては各ノード平均 159,562 秒から 142,664 秒と微減に留まっています。

また、「enq:TX-index contention」削減の代わりに、「gc buffer busy」の各ノード平均値が 53,141 秒から 70,951 秒と増加しています。この理由としては「gc cr block lost」の増加が関連している事を確認しています。これに対する対処を次章のチューニング 2 で実施しています。

4-3-3. 検証②チューニング2 インターコネクト通信処理の最適化

Linux OS 上における Oracle Real Application Clusters では、インターコネクト (Private Network LAN) の通信プロトコルとして UDP を使用しています。通常、ほとんどの Linux OS において、UDP 通信におけるデフォルトの送受信バッファサイズ (rmem_max/wmem_max) は Oracle Real Application Clusters を使用するにあたって十分なサイズが確保されている状態ではありません。

Oracle Real Application Clusters 利用時の UDP 通信における送受信バッファサイズの目安としては通常以下の式から求められます。

$$\text{送受信バッファサイズ} = \text{db_file_multiblock_read_count} (32) \times \text{db_block_size} (8\text{KB}) = 256\text{KB}$$

括弧内は本試験での設定値

上式により、UDP 通信における送受信バッファサイズとして 256KB を設定しましたが、チューニング 1 実施の結果として Oracle 待機イベント「gc cr block lost」が発生している事から、このサイズでは不足していると判断し 1MB まで増加しました。その結果が図 4-11 のグラフです。

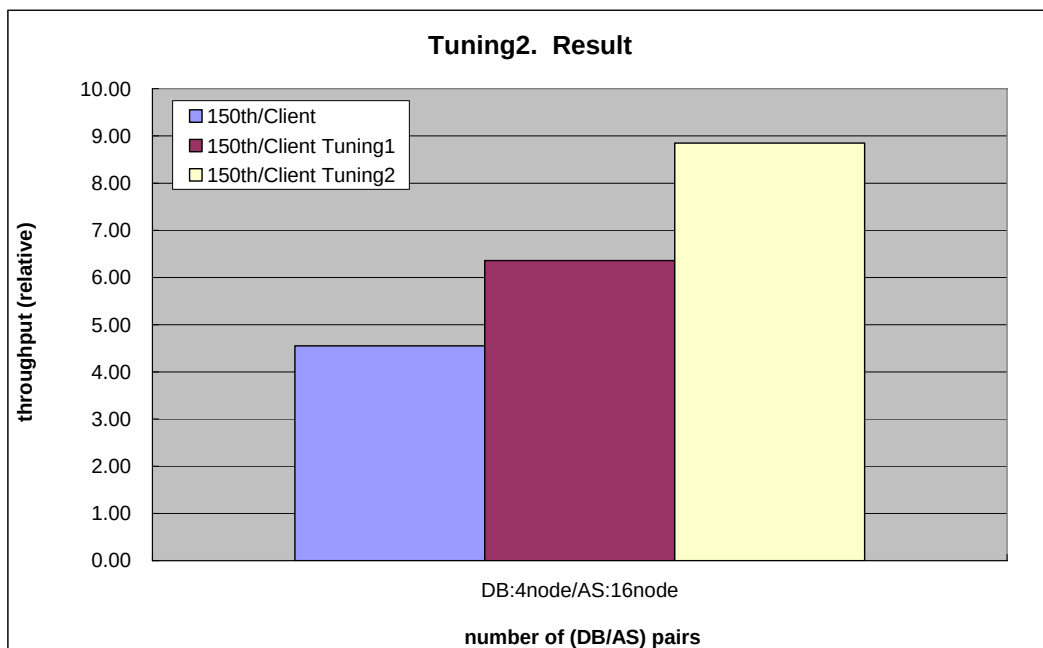


図 4-11 Tuning2.実施前後のスループット比較

チューニング 2 により、スループット比が 4.55 から 8.85 に増加しました。この数字は 4 ノード構成時のスケールビリティとしては十分な値といえます。

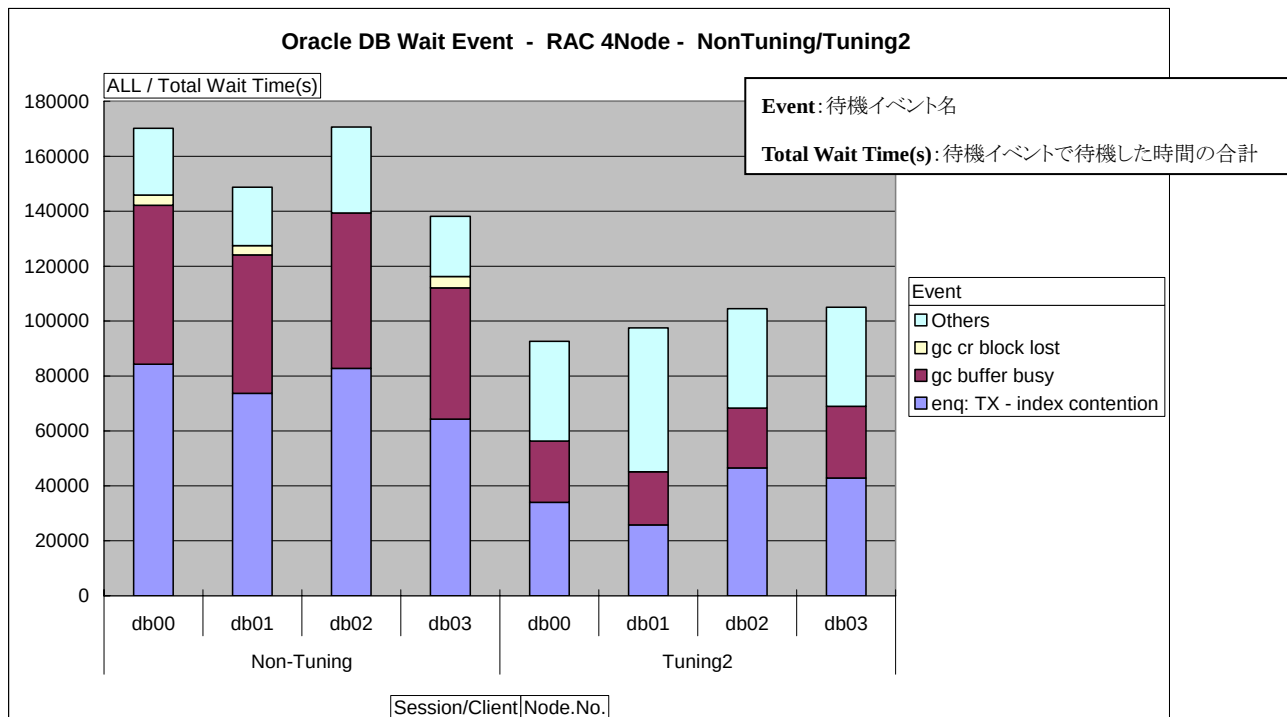


図 4-12 未 Tuning-Tuning2 の Oracle 待機イベント比較

図 4-12 はチューニング 2 実施前後の待機イベント上位の比較です。

横軸はデータベースサーバのノード名、縦軸は各待機イベントで待機した時間の合計秒数を示しています。

また表 4-2 はチューニング2実施前後の待機イベント「enq:TX-index contention」「gc buffer busy」「gc cr block lost」「合計秒数(ALL)」の各詳細値を示しています。

表 4-2 Tuning2 結果 Oracle 待機イベント詳細

		enq: TX-index contention		gc buffer busy		gc cr block lost		ALL
Tuning No.	ノード No.	Total Wait Time(s)	%Wait Time	Total Wait Time(s)	%Wait Time	Total Wait Time(s)	%Wait Time	Total Wait Time(s)
Non-Tuning	db00	84,270s	50%	57,826s	34%	3,757s	2%	170,205s
	db01	57,826s	34%	50,390s	34%	3,388s	2%	159,315s
	db02	82,771s	49%	56,579s	33%	4,162s	3%	170,633s
	db03	64,307s	47%	47,770s	35%	4,162s	3%	138,094s
	Ave.	72,294s	45%	53,141s	34%	3,867s	3%	159,562s
Tuning1	db00	39,865s	32%	42,482s	34%	5,908s	5%	125,505s
	db01	23,456s	15%	84,199s	56%	12,123s	8%	151,390s
	db02	29,154s	20%	76,824s	52%	10,298s	7%	148,048s
	db03	22,096s	15%	80,299s	55%	10,736s	7%	145,713s
	Ave.	28,643s	21%	70,951s	49%	9,766s	7%	142,664s
Tuning2	db00	34,004s	37%	22,363s	24%	発生せず	-	92,632s
	db01	25,806s	26%	19,330s	20%	発生せず	-	97,563s
	db02	46,539s	45%	21,789s	21%	発生せず	-	104,521s
	db03	42,911s	41%	26,098s	25%	発生せず	-	105,047s
	Ave.	37,315s	37%	22,395s	23%	発生せず	-	99,941s

表 4-2 からチューニング 2 により「gc block lost」発生しなくなり、それに伴い「gc buffer busy」の各ノード平均値が 70,951 秒から 22,395 秒と改善している点と、全待機イベント合計 (ALL) も未チューニングにおいてノード平均 159,562 秒からチューニング 1 の実施でノード平均 142,664 秒、さらにチューニング 2 の実施でノード平均 99,941 秒と、段階的に改善している事が確認できます。

4-3-4. チューニング前後の比較 Oracle Real Application Clusters 4 ノード構成

チューニング前後の結果を改めて比較してみます。

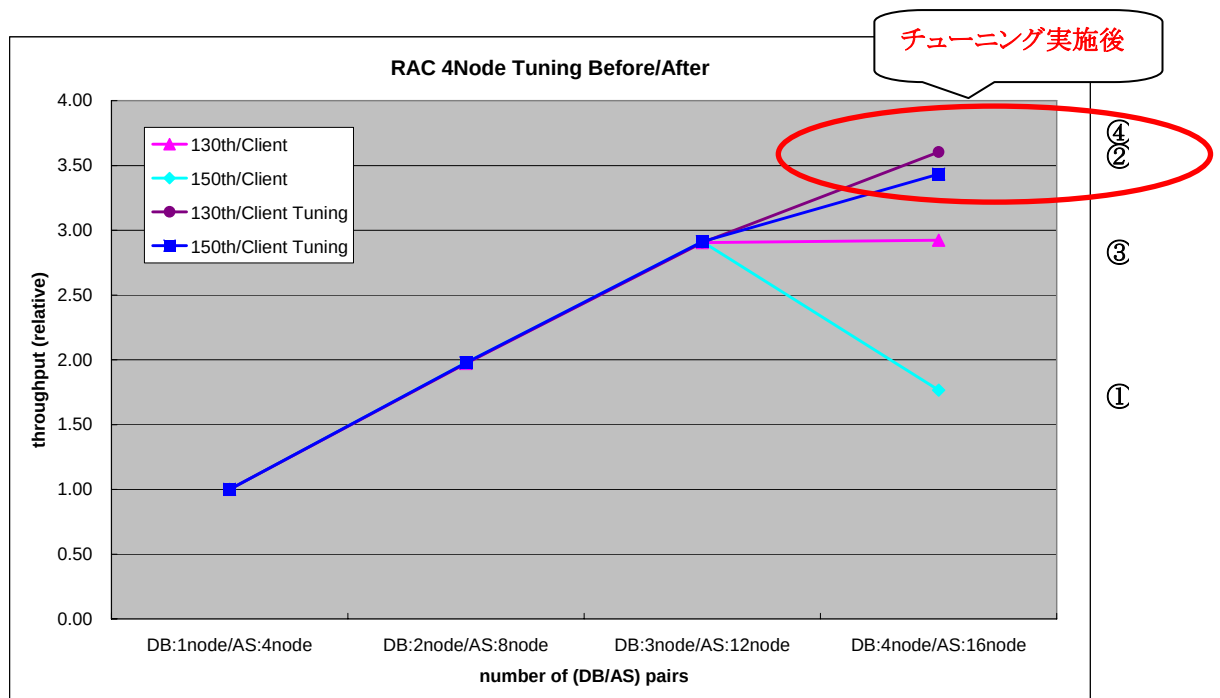


図 4-12 チューニング前後比較 DB 層 4 ノード構成 – 同時接続数 130/150

図 4-12 にクライアント1台毎の同時セッション数 110～150 のテスト結果について、各チューニング実施前後のスループット比を比較した結果を示します。

横軸はデータベースサーバ/アプリケーションサーバの各構成台数、縦軸はスループット比率を示しています。グラフの系列については以下のとおりです。

- ◆-: ① 1クライアントからの同時接続数が 150 の場合、未チューニング
- : ② 1クライアントからの同時接続数が 150 の場合、チューニング 1 と 2 を実施
- ▲-: ③ 1クライアントからの同時接続数が 130 の場合、未チューニング
- : ④ 1クライアントからの同時接続数が 130 の場合、チューニング 1 と 2 を実施

チューニングの結果、データベースサーバ/アプリケーションサーバのノード台数にほぼ比例したスループットを得る事が出来ています。但し今回実施していない Index のパーティション化といったチューニングの余地を残しています。

5. まとめ

今回の検証では、PRIMEQUESTを中心に富士通のLinuxプラットフォームを利用したWeb三階層システムにおいて、データベースサーバ層/アプリケーションサーバ層のスケーラビリティを評価し、データベースについては8ノード、アプリケーションサーバについては16ノードまでのスケールアウトが無理なく可能である事が確認されました。

また、PRIMEQUESTのPPAR機能と、Oracle Real Application Clustersの組合せにより、1ノードあたりで利用するCPU数を4CPU単位、あるいは8CPU単位に変更して評価する事で、データベースサーバ層のノード構成を変更しても利用するCPU数合計に応じた性能が得られる事が確認できています。

基幹IAサーバ「PRIMEQUEST」はLinuxをはじめとしたオープンアーキテクチャを採用しつつ、メインフレームで培ったノウハウにより拡張性・信頼性・柔軟性を実現しており、大規模データベースシステムにも安心して導入頂けます。さらに基盤ソフトウェアPRIMECLUSTER、Systemwalkerの利用により、システム全体の信頼性・柔軟性をさらに高度なものとする事が可能です。

富士通はこれからも、様々なビジネスニーズに応えるIT基盤「TRIOLE(*)」に基づき、お客様のビジネスの「機敏性」「効率化」「継続性」を支えて参ります。

*TRIOLE(トリオーレ)...お客様のIT基盤を支えるプラットフォーム製品コンセプト

<http://triole.fujitsu.com/jp/concept/>

最後に本検証で実施したPRIMEQUESTとOracle Real Application Clustersのスケールアウト検証とは別にPRIMEQUEST1サーバ当りのCPU・メモリリソースを増強する事により、サーバ1台の性能向上を得るスケールアップについての検証について末尾付録にてご紹介します。

スケールアップ検証ではスケールアウト検証時と構成が異なるため、両者の直接比較ができる結果となっておりますが、参考として掲載します。

6. 付録

付録 PRIMEQUEST / Oracle Database のスケールアップ

本資料では、PRIMEQUEST の PPAR 機能を用い、Oracle Real Application Clusters(RAC)の構成ノード台数を増加した際にクラスタシステムトータルとしてのスケーラビリティ、すなわちスケールアウト性能がどの程度得られるかを検証しています。

このスケールアウトとは別の考え方としてサーバ1台辺りの CPU・メモリといったリソースを増強する事により、サーバ1台の性能向上を得るスケールアップという方法があります。

本資料の検証結果とは別に、Oracle Database (非クラスタ構成)の構成を PRIMEQUEST1台とし、PPAR 機能を用いた1台辺りの CPU 数を増加させた際のスケールアップ検証について紹介します。

スケールアップ検証でも JPetStore を利用しましたが、システム構成が異なり、データベースサーバ層/アプリケーション層/クライアント層 の各構成台数比を 1: 24: 24 としています。(*1)

その上でデータベースサーバを構成する PRIMEQUEST 580 の PPAR 機能を利用する事で、1パーティション(=1サーバ)の CPU 数を 4CPU~16CPU と可変させ、PRIMEQUEST, Oracle Database 両者のスケーラビリティ(スケールアップ)を評価しています。

この結果については特に Database に関するチューニングを施していない結果となります。

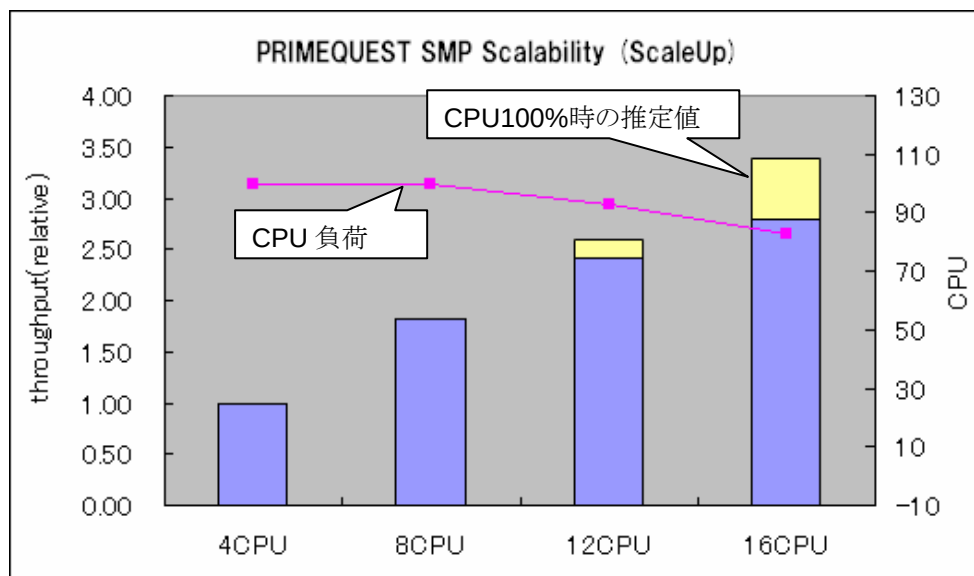


図-付 1 Oracle Database スケールアップ検証結果

図-付 1 は、PRIMEQUEST580 の 4CPU 構成時のスループットを1とし、CPU 数を16CPU まで増加させた際のスループットの増加比率を示しています。縦軸はスループット比率、横軸はデータベースサーバを構成している PRIMEQUEST の CPU 数を示しています。

4CPU 構成に対して、CPU 数を2倍/8CPU とした際のスループット比率は 1.82 倍、4倍/16CPU とした際のスループット比率は 2.80 倍となっており、CPU 数が増えるに連れてスループット向上比率が若干低下してはいますが、全体的に良好なスケールアウト性能が得られています。

スループット向上比率が若干している12CPU、16CPU 構成時はアプリケーションサーバ、クライアントの CPU 負荷が上昇してゆき、変わりにデータベースサーバの CPU 負荷が減少するという傾向となり、データベースサーバに十分な負荷を与える事ができていない状況が確認できました。

データベースサーバの統計情報、Network/Disk 負荷に特に問題が見つからなかった点からアプリケーションサーバ/クライアントの構成台数不足の状況であったと考えます。アプリケーションサーバ/クライアント構成台数を増やし適切なバランスとし、データベースサーバの各 CPU 構成時の負荷を均等とする事で十分なスケールアウト性能が見込まれます。

(*1) PRIMEQUEST のシステム構成中、LAN Driver として bcm5700-RHEL4-U4-8.3.14-1/e1000-RHEL4-U4-7.0.33-1 を利用し、OS 起動設定として nojitter オプションを利用しています。

7. 参考情報

Oracle Web サイト

- (1) Oracle Japan / Oracle GRID Center
http://www.oracle.co.jp/solutions/grid_center/

富士通 Web サイト

- (1) 基幹 IA サーバ PRIMEQUEST(プライムクエスト)
<http://primeserver.fujitsu.com/primequest/>
- (2) ストレージシステム ETERNUS(エターナス)
<http://storage-system.fujitsu.com/jp/>
- (3) ソフトウェア Oracle
<http://software.fujitsu.com/jp/oracle/>
- (4) Platform Solution Center (プラットフォームソリューションセンタ)
<http://triole.fujitsu.com/jp/psc/>

本書について

本書はOracle GRID Centerへの取組みにて実施された検証結果に関する技術情報を提供するものであり、本書に記載されている内容は改善のため、予告なく変更することがあります。

富士通株式会社は、本書の内容に関して、いかなる保証もいたしません。また、本書の内容に関連した、いかなる損害についてもその責任は負いません。

本書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。本書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。オラクル社は、本書についていかなる責任も負いません。また、本書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。本書を形式、手段(電子的又は機械的)、目的に関係なく、オラクル社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。

商標について

Oracleは、米国Oracle Corporation及びその子会社、関連会社の登録商標です。

Intel、Itanium、Xeonは、米国Intel Corporationの登録商標または商標です。

Red Hatは米国およびその他の国でRed Hat, Inc.の登録商標または商標です。

LinuxはLinus Torvaldsの商標です。

Windowsは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

記載されている会社名、製品名は各社の登録商標または商標です。

日本オラクル株式会社

〒102-0094
東京都千代田区紀尾井町4-1
ニューオータニガーデンコート
<http://www.oracle.co.jp/>

富士通株式会社

〒105-7123
東京都港区東新橋 1-5-2
汐留シティセンター
<http://primeserver.fujitsu.com/primequest/>