

富士通ブレードサーバ上での Oracle RAC / Oracle BIEE の性能検証 － ノード追加によるスケーラビリティ －

Creation Date: October 1, 2008

Last Update: March 27, 2009

Version: 1.0

ORACLE®

FUJITSU

はじめに

日本オラクル株式会社(以降、日本オラクル)と富士通株式会社(以降、富士通)は、1989 年に OEM 契約を締結して以来、お客様が安心してご利用いただけるソリューションを提供するためのシステム構築、共同検証、導入後のサポートなど、様々なアライアンス活動を行って参りました。

2006 年 11 月、日本オラクルは富士通をはじめとする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 Center¹」を開設しました。富士通はその開設に賛同し、富士通のもつサーバ、ストレージ製品を使用して Oracle GRID Center にて共同で技術検証を行っています。

この度、両社は富士通の最新ブレードサーバ PRIMERGY BX620 S4 上において、Oracle Application Testing Suite で生成した負荷を使用して、Oracle Business Intelligence Suite 及び Oracle Real Application Clusters で構成される BI システム全体の性能測定及びスケーラビリティ検証を Oracle GRID Center にて実施しました。その成果をここに報告致します。

¹ 詳細な活動内容は次の Web サイトをご覧ください。

Oracle GRID Center (http://www.oracle.co.jp/solutions/grid_center/index.html)

目次

1. 検証目的.....	4
2. 検証結果サマリー.....	6
2.1. ノード追加による BI システムのスケールビリティ.....	6
2.2. 複数レイヤーでのリソース消費.....	6
2.3. ノード追加手順に関する検証.....	6
3. プラットフォーム紹介.....	7
3.1. PRIMERGY(プライマジー) ブレードサーバ.....	7
3.2. Systemwalker Resource Coordinator Virtual server Edition V13.3.0.....	8
3.3. ストレージシステム ETERNUS (エターナス).....	9
3.4. ネットワークサーバ IPCOM(アイピーコム).....	10
4. ソフトウェア紹介.....	12
4.1. Oracle Business Intelligence Suite Enterprise Edition Plus.....	12
4.2. Oracle Real Application Clusters.....	14
4.3. Oracle Application Testing Suite.....	16
5. 検証環境.....	18
5.1. システム全体構成.....	18
5.2. ソフトウェア構成.....	20
5.3. データベース構成.....	21
5.4. キャッシュの種類.....	22
6. 検証モデル.....	25
6.1. 業務モデル.....	25
6.2. 検証用モデル.....	26
7. ノード追加による BI システムのスケールビリティ.....	32
7.1. Oracle BIEE のスケールビリティ.....	32
7.2. Oracle RAC のスケールビリティ.....	40
8. 複数レイヤーでのリソース消費.....	46
8.1. 検証目的.....	46
8.2. 検証方法.....	46
8.3. 検証結果.....	47
9. ノード追加手順に関する検証.....	49
9.1. RCnr VE によるプラットフォーム追加手順検証.....	50
9.2. Oracle RAC ノード追加手順検証.....	51
10. 総合評価.....	57

1. 検証目的

近年の急速に発展/変化するビジネス環境において速やかに正確な判断を下す企業経営や業務を実現する為に、情報の「見える化」が必要となってきました。情報の「見える化」により現状と理想のギャップを把握することが可能となり、今現在抱えている問題等の早期発見や効率化及び改善に役立ち、より理想像へと近づけることが可能となります。

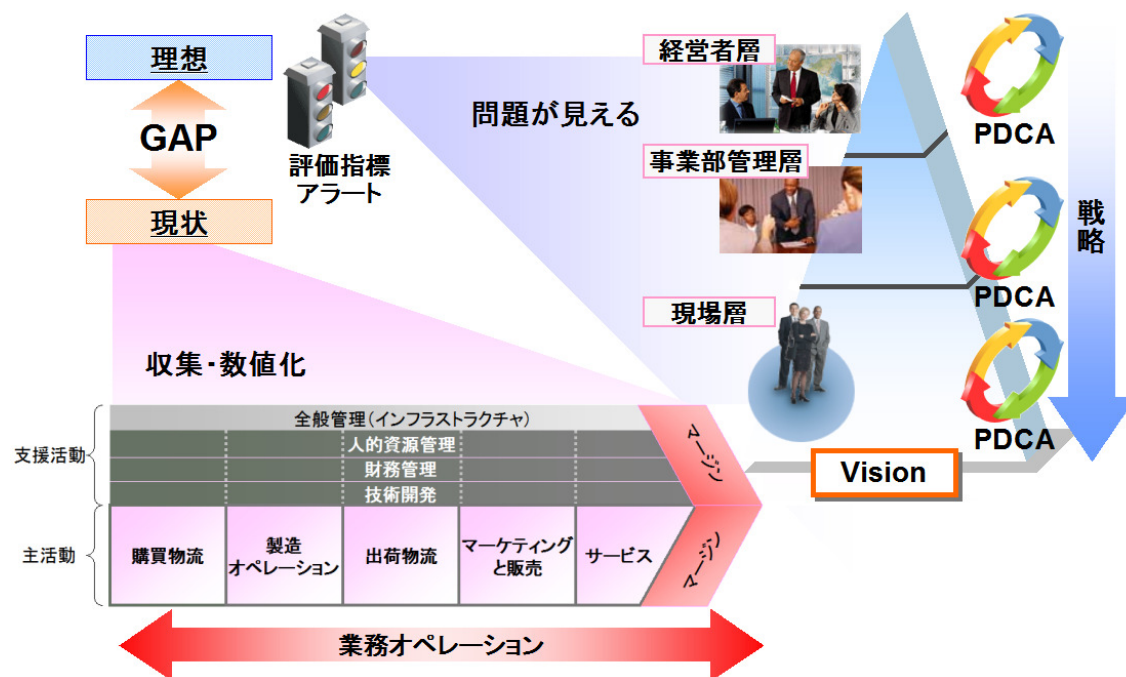


図 1-1 情報の「見える化」

この「見える化」を実現するために必要となる、分散しているデータを整理し正しく導き出す枠組み=Business Intelligence(以降、BI)を提供するのが Oracle Business Intelligence Suite Enterprise Edition(以降、Oracle BIEE)であり、優れた BI システムを構築することが可能となります。

エンタープライズ BI システムにおいては、大量のデータに多数のユーザがアクセスするため、データベースのレイヤーだけでなく、アプリケーションサーバのレイヤーにも大きな負荷がかかります。また、負荷の傾向も様々なタイプが混在します。そのため、サーバのサイジングには複数のレイヤーを考慮する必要があります。

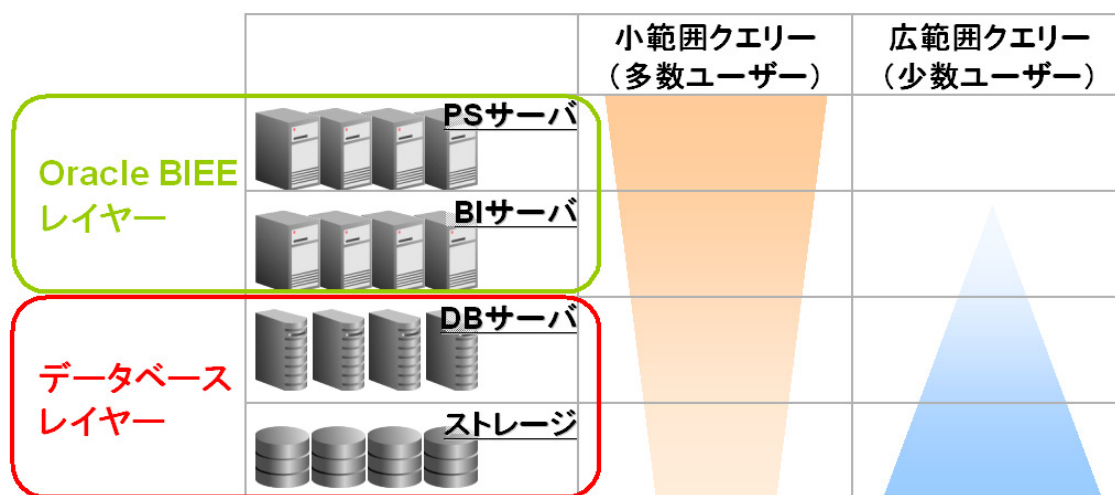


図 1-2 BI システムの消費リソース

例えば、検索範囲の狭い小範囲クエリは、多数のユーザが連続的に実行することが予想され、Oracle BIEE レイヤーでの画面生成などによるリソース消費が高くなる傾向にあります。広範囲クエリはその検索範囲の広さから、図 1-2 のデータベース・レイヤーのリソース消費が高くなると考えられます。反面、広範囲クエリが発行されるのは月次レポートの作成時などであり、小範囲クエリのように多数のユーザが連続的に発行するわけではないことから、Oracle BIEE レイヤーのリソース消費は少ないと考えられます。²

本検証では、日々増加する業務処理量へ対応できる高い性能/拡張性とシステムの安定稼動を実現する信頼性/可用性を備えている富士通 PRIMERGY ブレードサーバ上に、Oracle BIEE と Oracle Real Application Clusters(以降、Oracle RAC)で構築した BI システムを構築し、各レイヤーにおけるスケールアウトによる処理性能のスケラビリティを確認することを目的として実施しました。

² 小/広範囲クエリの定義については 6.2. 検証用モデルで詳説します。

2. 検証結果サマリー

本検証結果の概要を次に示します。詳細については後続の該当章をご確認下さい。

2.1. ノード追加による BI システムのスケーラビリティ

2.1.1. Oracle BIEE のスケーラビリティ

BI Presentation Server のキャッシュ・ヒット率毎 (0~100%) の処理性能を測定することで、キャッシュ・ヒット率が高いほど処理性能が高いことを確認しました。

詳細⇒7.1.3. PS サーバのキャッシュ・ヒット率毎の性能

ユーザ数の増加に応じて BI システムを構成する BI Presentation Server と BI Server のノードを追加することで処理性能の向上を確認し、スケールアウトによる Oracle BIEE の高いスケーラビリティが確認できました。

詳細⇒7.1. Oracle BIEE のスケーラビリティ

2.1.2. Oracle RAC のスケーラビリティ

Oracle BIEE レイヤー³がボトルネック要因でない場合、特に広範囲を検索するクエリーを実施する場合は、データベースサーバの CPU、またはストレージがボトルネックになる傾向があります。このような場合、Oracle RAC のノード追加及びストレージディスクの増設によって処理性能が向上しました。従って Oracle RAC の高いスケーラビリティが確認できました。

詳細⇒7.2. Oracle RAC のスケーラビリティ

2.2. 複数レイヤーでのリソース消費

図 1-2 で示した小範囲クエリーと広範囲クエリーは、BI Presentation Server のキャッシュの効果により消費するリソースの傾向が異なっています。このような場合、同時に実行しても他方のクエリー性能への影響は小さいと確かめられました。

詳細⇒8. 複数レイヤーでのリソース消費

2.3. ノード追加手順に関する検証

富士通の Systemwalker Resource Coordinator 及び Oracle RAC のクローニング機能を活用することによって、業務への影響を最小限に抑えながら迅速にクラスタにノードを追加できることが確かめられました。

詳細⇒9. ノード追加手順に関する検証

³ Oracle Presentation Services および Oracle BI Server で構成されるレイヤーを示します。

3. プラットフォーム紹介

3.1. PRIMERGY(プライマジー) ブレードサーバ

近年、ビジネスの多様性に伴い、様々なニーズに柔軟に対応できるシステムが求められています。富士通のブレードサーバは、日々増加する業務処理量へ対応できる高い性能/拡張性と、システムの安定稼動を実現する信頼性/可用性を備えています。

ブレードサーバ特有の省スペース性に加え、消費電力や CO₂ の削減など環境にも配慮した設計により、大幅な TCO の削減を実現します。

●ブレード構造により省スペース化を実現

発電効率に優れた AC200V 対応のシャーシに加え、日本国内で一般的に使用されている AC100V 対応のシャーシを提供します。高さ 7U のシャーシにサーバブレードを最大 10 枚まで搭載可能であり、サーバブレードだけでなく、その他のモジュールに関しても同シャーシ内に搭載可能です。標準でマネジメントブレード、内蔵電源ユニットを各 2 台搭載しており、オプションでストレージブレード、ネットワークモジュールなどを搭載することができます。

BladeServer Lineup

PRIMERGY BX600 S3 シャーシ



7U

200Vシャーシ

- サーバブレード最大10枚(ホットプラグ対応)
- マネージメントブレード×2標準搭載(ホットプラグ対応)
- 冗長電源(オプション)/冗長ファン(標準)
- AC200V対応
- RoHS対応

100Vシャーシ

- サーバブレード最大10枚(ホットプラグ対応)
- マネージメントブレード×2標準搭載(ホットプラグ対応)
- 冗長電源(オプション)/冗長ファン(標準)
- AC100V対応
- RoHS対応

PRIMERGY BX620 S4 サーバブレード



メモリスロット CPU
LANコントローラ ハードディスク (2.5インチ)
ファイバーチャネル拡張ボードまたは LAN拡張ボード (オプション) 拡張カードスロットモジュール

CPU クアッドコア インテル Xeon プロセッサ X5460/E5420/E5405
デュアルコア インテル Xeon プロセッサ X5260/E5205
クアッドコア インテル Xeon プロセッサ L5410(低消費電力)

メモリ 2GB(～32GB)DDR2 SDRAM FB DIMM

HDD 2ベイ(MAX 296.3GB)

PRIMERGY SX650 ストレージブレード



PRIMERGY BX620 S4
サーバブレード

PRIMERGY SX650
ストレージブレード

- サーバブレードに接続し、内蔵ハードディスクを拡張可能
- PRIMERGY BX600 S3シャーシに最大5台搭載可能
- 2.5インチSAS HDDを最大5台(MAX 734GB)搭載(ホットプラグ対応)

図 3-1 BladeServer Lineup

3.2. Systemwalker Resource Coordinator Virtual server Edition V13.3.0

PRIMERGY ブレードサーバでは、高機能なミドルウェアである「Systemwalker Resource Coordinator Virtual server Edition V13.3.0（以降、RCnr VE）」と連携することで、導入時の省力化と運用の自動化を実現し、TCOを抑えながら24時間365日の安定稼働を引き出せます。

RCnr VE は、ブレードサーバの導入から保守に至る一連の運用ライフサイクルにおける簡易化と自動化を実現し、システム運用管理者の負担を大幅に削減するサーバ管理ソフトウェアです。ブレードサーバに集約されたサーバを簡単操作で一括管理することが可能であり、サーバの導入/増設時や万が一のサーバ障害時における予備サーバへの短時間での切り替え等の運用を自動化することで、コストを抑えながらの安定運用を実現します。

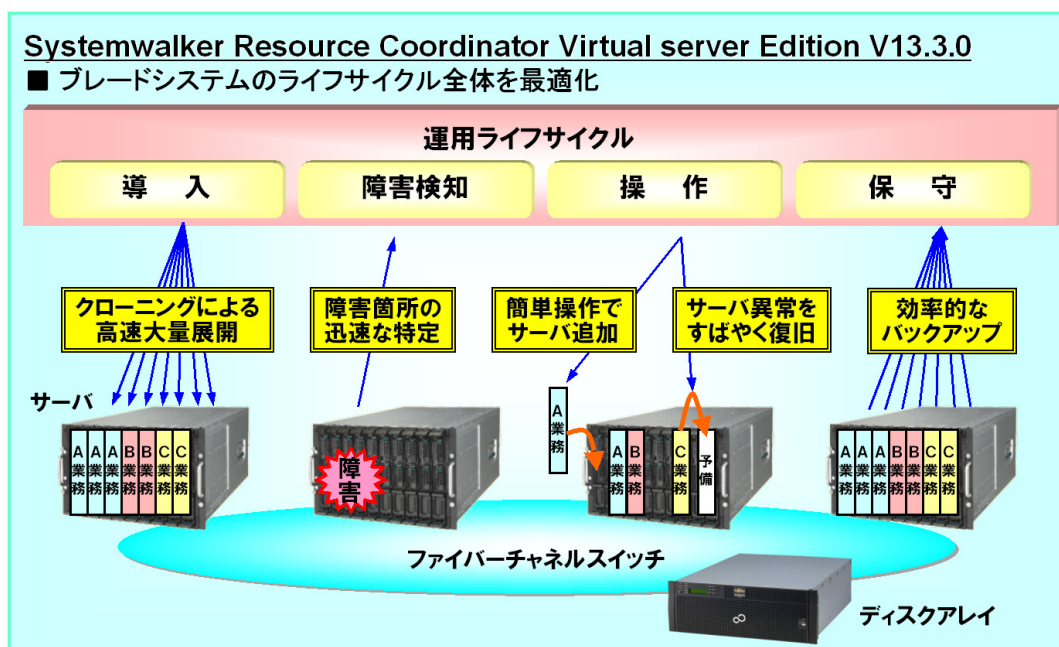


図 3-2 Systemwalker Resource Coordinator Virtual server Edition V13.3.0

●サーバ導入・設定の自動化・サーバ増設の簡易化

サーバの I/O 仮想化技術により、サーバ管理者は SAN の設定を意識せずにサーバの導入・設定を行うことが可能となります。

1 台のサーバで作成したシステムディスクの内容(システムイメージ)を複数の物理サーバへ複写(クローニング)することで、一括してインストールすることが可能です。また、業務 LAN の設定も自動的に行われます。

サーバ増設では、サーバに仮想アドレス WWN⁴を設定し、システムイメージを選択するだけで自動的に SAN ストレージが認識され、OS 設定・起動を完了することができます。

⁴ World Wide Name の略

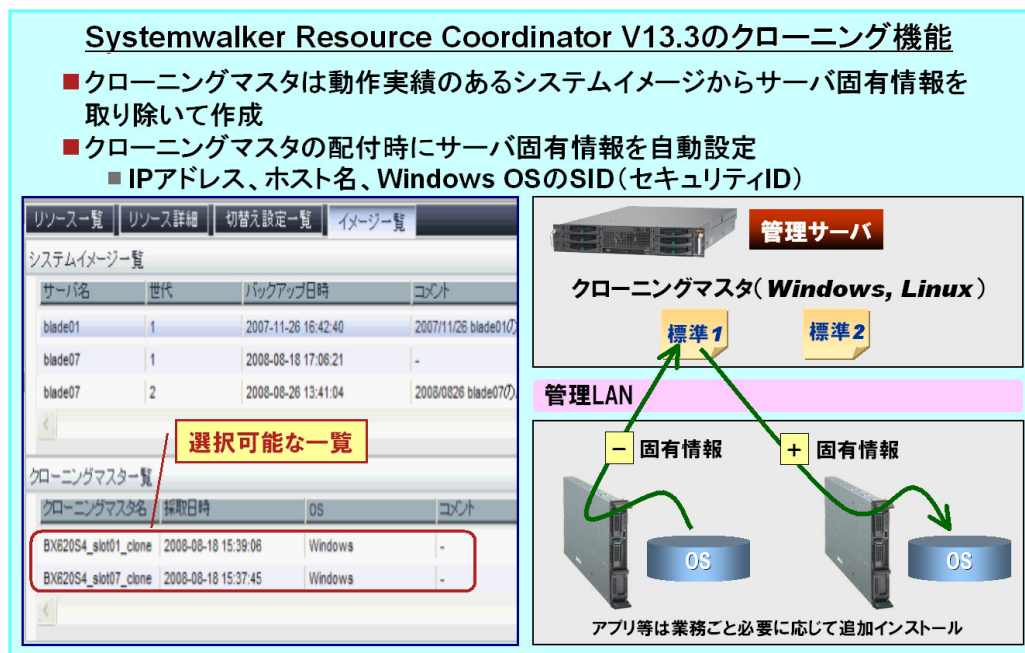


図 3-3 Systemwalker Resource Coordinator Virtual server Edition V13.3.0 のクローニング機能

3.3. ストレージシステム ETERNUS (エターナス)

富士通は、SAN 対応ディスクアレイとして、2.04PB(ペタバイト)の世界最大の容量を実現したエンタープライズディスクアレイ「ETERNUS8000」と、コストパフォーマンスに優れたミッドレンジディスクアレイ「ETERNUS4000」、省スペース・省電力・静音設計なエントリーディスクアレイ「ETERNUS2000」を提供し、幅広い製品ラインナップであらゆるニーズに応えます。

部品レベルからシステムレベルまで徹底した信頼性を追及し、重要なデータを確実に保管します。また、マルチプラットフォーム環境に対応する優れた接続性を備え、SAN を利用したストレージ統合を実現します。オンラインでの高速バックアップを実現するアドバンス・コピー機能を採用し、バックアップに要する時間を大幅に削減します。

企業の重要なデータを確実にかつ効率的に保管し、コンプライアンス対応の強化など、企業の抱える様々な課題に対して最適なストレージソリューションを提供します。

●ETERNUS4000 モデル 300

ミッドレンジディスクアレイ「ETERNUS4000 モデル 300」は、2GHz の高速デュアルコアプロセッサと4Gbps のファイバチャネル・インターフェースを採用し、高い処理能力を実現します。また、最大 117.2TB の優れた拡張性を備え、大規模なストレージ統合やデータ量の増加にも対応します。

コントローラや電源、ファン、バッテリー等の主要コンポーネントを冗長化しており、停電発

生時にも内蔵バッテリーでキャッシュ内のデータをディスクに退避させ、保持日数の制限無く保護します。さらに、RAID グループ内にパリティディスクを 2 つ持つ RAID6 を利用可能で、二重ディスク障害時にもデータを保護。高信頼設計で 24 時間 365 日無停止稼動をサポートします。

装置のアドバンスド・コピー機能とソフトウェア「ETERNUS SF AdvancedCopy Manager」との連携により、業務運用中の高速コピー/バックアップを行うことができます。バックアップデータは複数世代持つことができ、バックアップの世代管理に対応します。さらに、ディスクアレイ間での遠隔地へのデータ転送も可能で、万が一の災害にもデータを保護します。

【ETERNUS4000 モデル 300 の特徴】

- 最大 117.2TB の優れた拡張性
- 2GHz の高速デュアルコアプロセッサと 4Gbps のファイバチャネル・インターフェースで高い処理能力を実現
- 24 時間 365 日無停止稼動に対応する高信頼設計
- アドバンスド・コピー機能で業務運用中に高速バックアップが可能
- ディスクアレイ間で遠隔地へデータ転送が可能で、災害時にもデータを保護

3.4. ネットワークサーバ IPCOM(アイピーコム)

IPCOM EX シリーズは、IT システムに必要な高信頼性ネットワークを実現するネットワークサーバです。負荷分散(ロードバランサ)・帯域制御・暗号機能(IPsec/SSL)・ファイアーウォール(FW)・UTM(アンチウイルス・Web コンテンツフィルタリング)など IT システムのフロントに必要な機能を提供するネットワークサーバです。



図 3-4 IPCOM EX IN シリーズ

「IPCOM EX IN シリーズ」は、サーバとネットワークの最適化とセキュリティ機能を集約します。高信頼なネットワーク環境の構築は IT システムのフロントを複雑にし、システム構築期間や運用コストを増大させます。IPCOM EX IN シリーズは多彩な機能を一台に統合することにより、システム構築と運用を簡略化し、TCO の大幅削減を実現します。

【IPCOM EX IN シリーズの特長】

- サーバロードバランサと UTM を統合

サーバ集約ポイントで一括した脅威対策を行うために、サーバロードバランサ(サーバ負荷分散)とUTMを統合。セキュリティの脅威への対策をワンポイントに集中させ、強固なセキュリティ環境を提供します。

- シンプルかつスピーディーなネットワーク構築

IPCOM EX シリーズは、様々な機能を一台に統合しているため、複数の装置を組み合わせた場合と比べ、システム設計や設置作業、設置作業にかかる時間を大幅に短縮できます。

- 運用管理が容易で、トラブルからの復旧時間も短縮

運用管理において様々な機能統合としての効果を発揮します。複数装置を組み合わせた場合と比べ、監視するログや適用するパッチなどが減少するため、管理者の負担は大幅に軽減されます。

また、トラブル時の障害切り分けも容易なため、トラブルからの復旧時間も大幅に短縮できます。

4. ソフトウェア紹介

4.1. Oracle Business Intelligence Suite Enterprise Edition Plus

Oracle Business Intelligence Suite Enterprise Edition Plus(以降、Oracle BIEE)は、あらゆる分析やレポート作成機能を提供する、包括的なエンタープライズ BI スイート製品です。統一的でスケーラビリティの高い最新アーキテクチャを利用して、Oracle BI EE Plus は企業全体のさまざまなソースやデータにまたがるデータを元にしたインテリジェンスと分析を可能にし、全社規模での BI インフラとして完全で妥当性の高い洞察をもたらします。

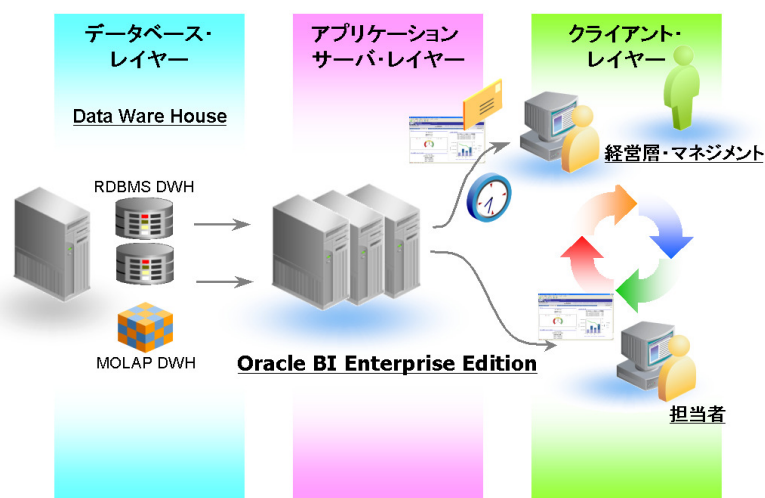


図 4-1 Oracle BI EE

4.1.1. Oracle Presentation Services

Oracle Presentation Services は、ダッシュボードの作成および表示、Web ベースの非定型分析/クエリー機能を提供します。ドリルダウン等のインタラクティブな環境も提供し、ユーザ操作性に優れており、複雑なデータ構造に煩わされることなく、簡単に分析結果レポートを作成できます。レポートには、豊富な分析スタイル(チャート、表、ピボットテーブル、グラフ、ティッカー等)が使用可能で、より可視性を高めます。また、役割と必要性に応じて、個々のユーザ単位でアクセス制御が可能となっているなど、万全なセキュリティ対策機能も実装されています。

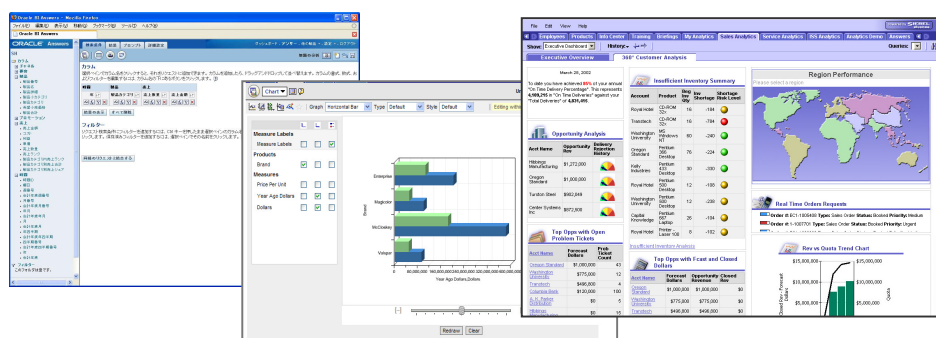


図 4-2 Oracle Presentation Services

4.1.2. Oracle BI Server

Oracle BI Server は Presentation Services からのリクエストを受け取り、各データソースにクエリーを発行してデータの取得と計算を行います。マルチデータソースに対応し、Oracle BI Server が複数のデータソースを仮想統合することで、ユーザは画面設計やクエリーの作成が可能となります。

データソースから取得した大量のデータに対して高度な計算を行うことができます。クエリーの実行結果をクエリー・キャッシュとして保存することにより、別のクエリーでキャッシュを再利用しデータソースでの検索やデータ取得後の計算を省くことができ、パフォーマンスを向上させることができます。

また、Oracle BI Server はクラスタ構成を組むことによりミッションクリティカル業務に耐える高可用性・拡張性を提供します。

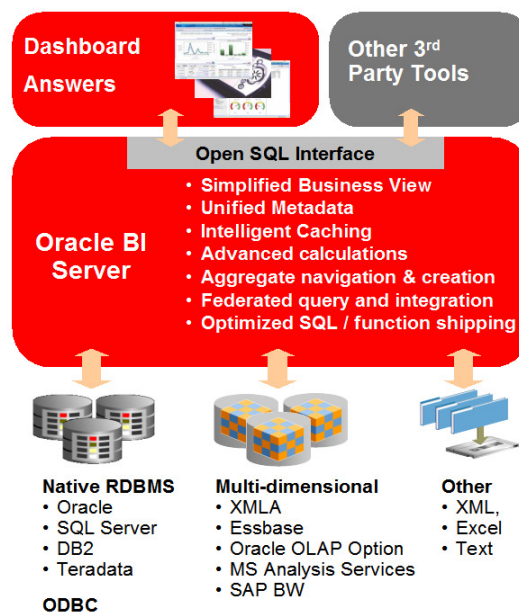


図 4-3 Oracle BI Server

Oracle BIEE は、図 4-4 のようなクラスタ構成を組むことが可能です。

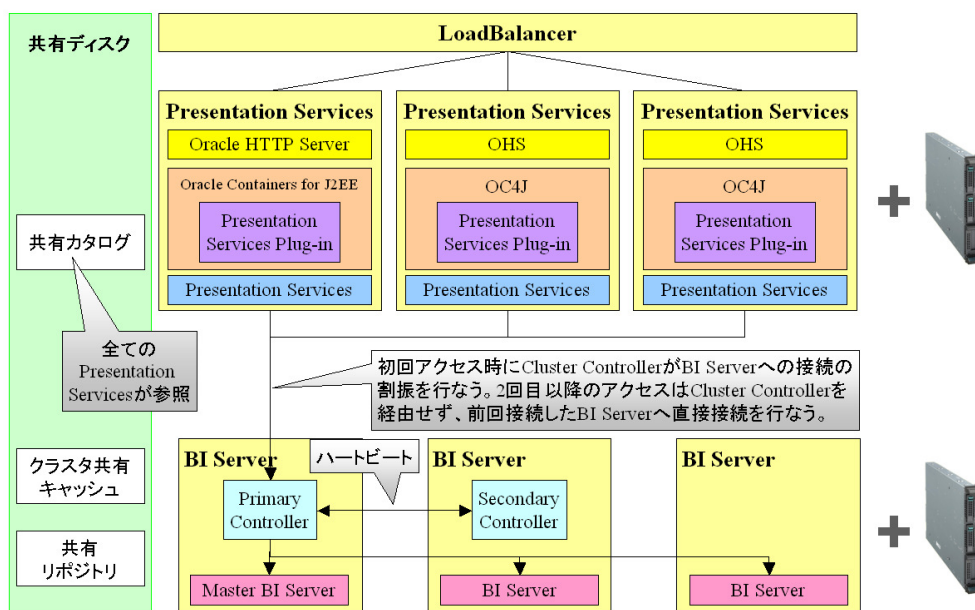


図 4-4 Oracle BIEE クラスタ構成

4.2. Oracle Real Application Clusters

Oracle Real Application Clusters (以下 Oracle RAC) とは、Grid 技術の根幹となるリソースを有効活用し、かつ障害にも強いオラクル独自のクラスタ技術です。

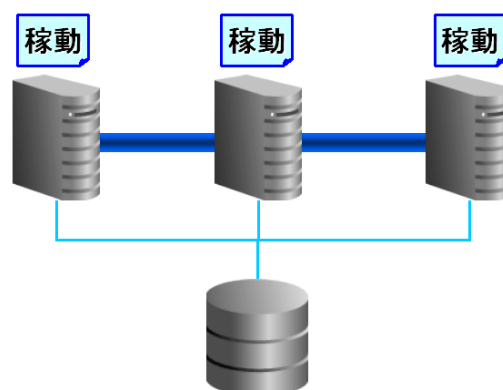


図 4-5 Oracle RAC

4.2.1. Oracle RAC の特長

Oracle RAC は複数ノードでディスクとメモリ上の情報を共有し、全ノードで並列に処理が可能なシェアドエブリシング型のデータベースです。

Oracle RAC を採用することにより、次のメリットを享受することが可能です。

1. アクティブ-アクティブ構成による、サーバリソースの有効活用
2. ノードを追加することによる、柔軟なシステム拡張
3. シェアドエブリシング構成による、障害発生時のシステム停止時間、切り替え時間の低減

Oracle RAC は、サーバを複数台並べて全ノードでデータベース処理を実行できます。処理量の異なる各業務のワークロードを、仮想的に統合された各サーバへ処理を分散し、リソースを有効活用するため、待機系サーバのために行っていた投資の稼働率を向上させます。

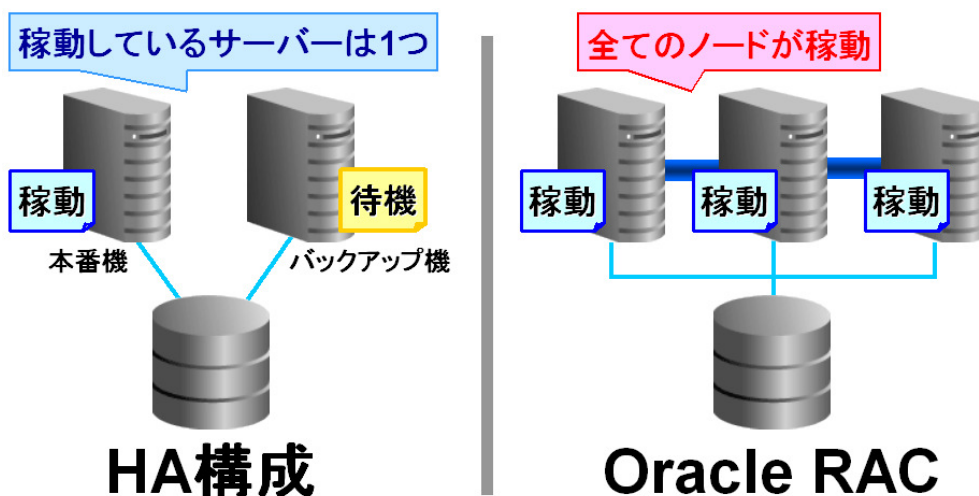


図 4-6 HA 構成と Oracle RAC

システムのハードウェアリソース増強が必要な場合も、それまで用いていたシステムを置き換えるのではなく、既存の環境に新たなノードを追加することで処理能力を向上させることが可能です。つまりビジネスの拡大に併せて、必要に応じて柔軟にリソースを追加できます。

さらに、全ノードで処理を実行している、すなわち全ノードがアクティブな状態で共有ディスクにアクセスしているため、万一、1 つのサーバがダウンしても、残りのサーバでシステムを継続でき、システムのサービス停止時間を低減することが可能です。

このように、Oracle RAC は投資対効果、システムの拡張性、高可用性の観点で、従来の HA 構成にはないメリットを提供します。

4.3. Oracle Application Testing Suite

4.3.1. 概要

Oracle Application Testing Suite は、機能テスト向けの「Functional Testing」と負荷テスト向けの「Load Testing」の 2 種類のテスト機能を備え、統合的なテスト管理をサポートする「Test Manager」から構成されています。

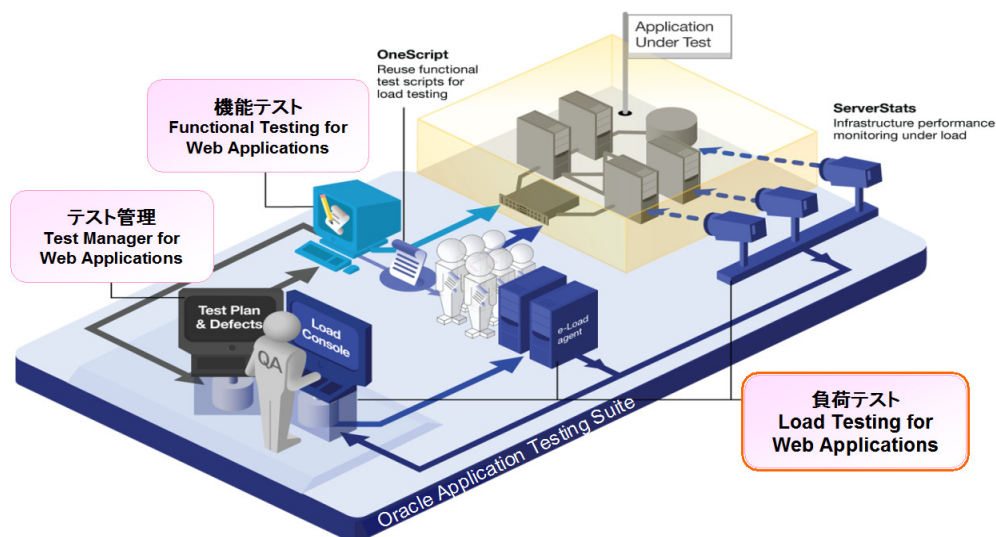


図 4-7 Oracle Application Testing Suite

4.3.2. システム開発のフェーズと Oracle Application Testing Suite

システム開発のフェーズに対する Oracle Application Testing Suite の各コンポーネントのマッピングは、図 4-8 のとおりです。

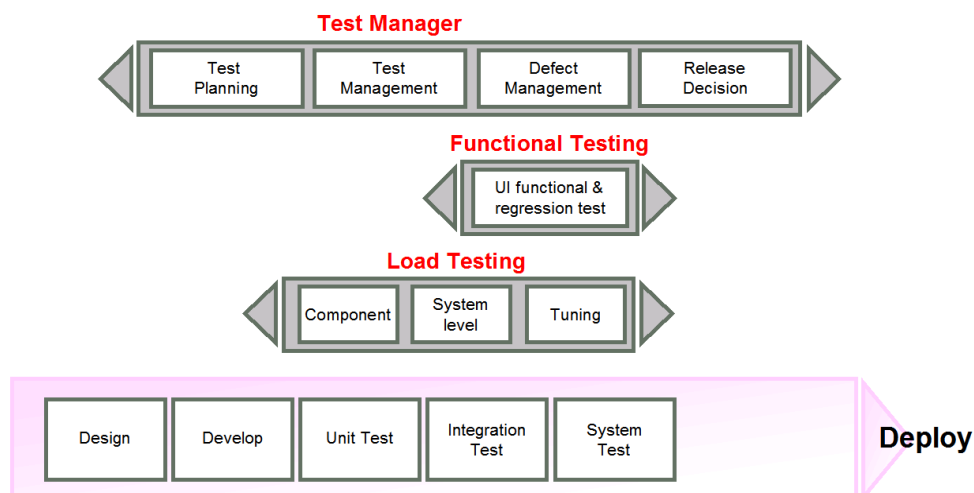


図 4-8 各コンポーネントのマッピング

4.3.3. Oracle Load Testing for Web Applications 概要

テスト環境において多数のユーザからのアクセスを擬似的に再現し、Web アプリケーションの性能検証を実現します。実運用開始前に、アプリケーションの性能問題を把握／解決することが可能となります。

優れた操作性

- ・ 理解しやすいビジュアルスクリプト
- ・ アイコンやウィザードによる GUI を提供

負荷テスト時のサーバモニタリング機能

- ・ サーバのリソース(CPU／メモリ)使用状況の取得
- ・ ミドルウェア (AP／DB サーバ) の性能情報の取得

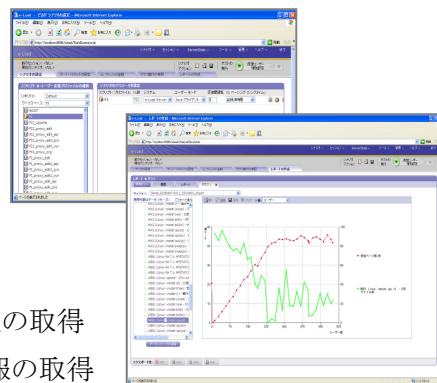


図 4-9 Oracle Load Testing for Web Applications

リアルタイムなテスト結果の表示

- ・ Web ページ毎のスループット／レスポンスタイムの表示
- ・ サーバのモニタリング結果と関連させたグラフレポートの作成
- ・ ユーザ視点によるエラー検知

4.3.4. Oracle Load Testing for Web Applications の構成例

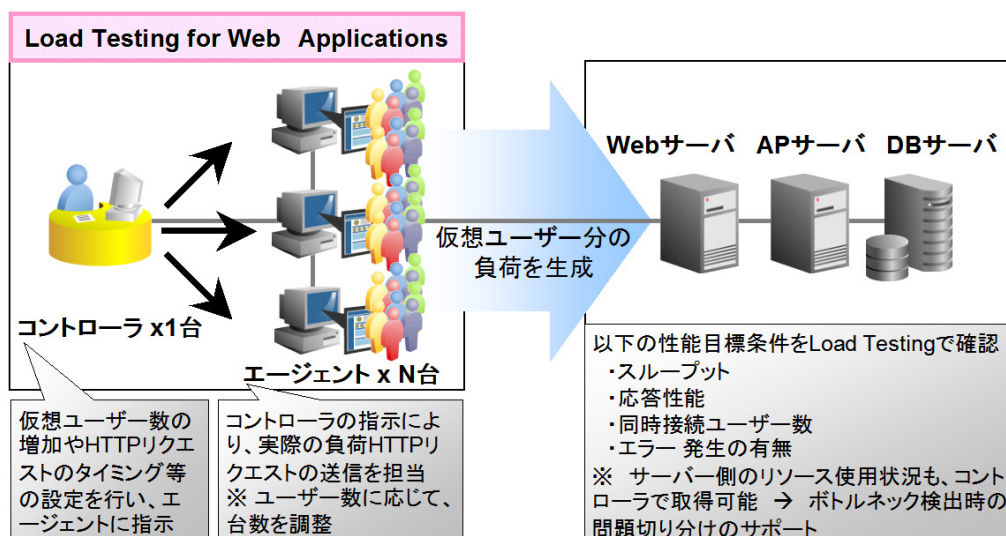


図 4-10 Oracle Load Testing for Web Applications の構成例

5. 検証環境

本検証で使用した GRID Center の検証環境について説明します。

5.1. システム全体構成

5.1.1. プラットフォーム構成

本検証で使用したプラットフォームの構成を、図 5-1 に示します。

Oracle BIEE レイヤーと DB レイヤーを別々のブレードシャーシ上に構成し、Oracle BIEE レイヤーは BI Presentation Server 用に最大 4 台のサーバブレード、BI Server 用に最大 4 台のサーバブレードを準備しました。DB レイヤーは Oracle RAC 4 ノード構成として、4 台のサーバブレードを使用しました。

Oracle BIEE レイヤーへの負荷分散として IPCOM EX2000 IN をラウンドロビンによるサーバロードバランサとして使用し、ネットワークファイルシステムである NR1000 は、BI Presentation Server 及び BI Server の各サーバで共有する情報を格納する用途で使用しました。データベースは DB サーバ用のブレードシャーシから Fibre Channel (以降、FC) で接続された ETERNUS 上に配置しました。

また、ETERNUS 上には、RCnr VE によるサーバブレードのクローニングで必要となるシステムイメージや、各サーバの Boot ディスク (SAN ブート環境) も配置した為、Oracle BIEE レイヤーのブレードシャーシからも FC で接続される構成となっています。

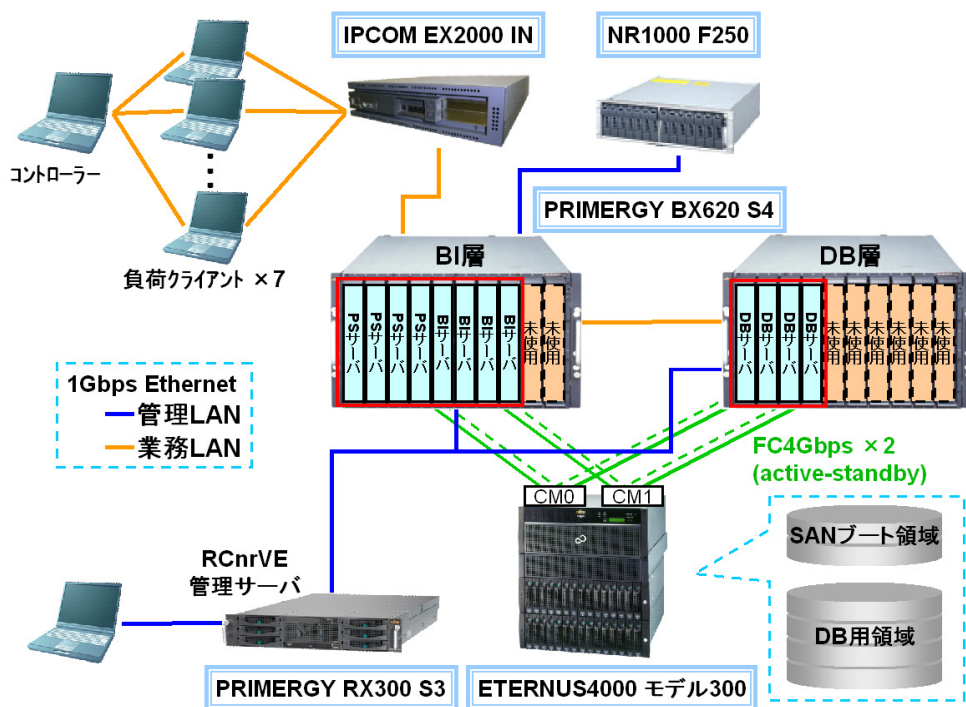


図 5-1 プラットフォーム全体構成

5.1.2. ストレージ構成

本検証で使用したストレージ(ETERNUS4000)の構成を図 5-2 に示します。

SAN ブート用はディスクドライブ 2 本 1 組で RAID1 を、データベース用に 4 本 1 組で RAID1+0 を構成しました。データベースでは、Oracle Automatic Storage Management (以降、ASM)を使用しています。各 RAID グループから 1 つの LU を切り出し、1 つの LU を 1 つの ASM ディスクとして ASM ディスクグループに割り当てました。詳細は、[5.3.2. ASM ディスクグループとユーザ]で説明します。

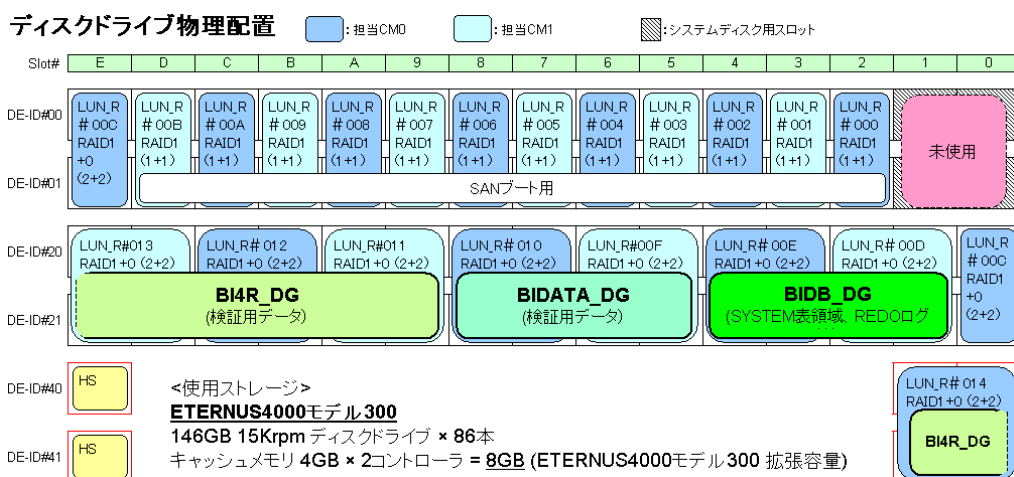


図 5-2 ストレージ構成

5.1.3. プラットフォーム仕様

以下に本検証における各ハードの仕様概要を示します。なお、以降は Oracle Presentation Services サーバを PS サーバ、Oracle BI Server を BI サーバと呼称します。

Oracle BIEE レイヤー: BX620 S4

[PS サーバ]

CPU	クアッドコア Xeon プロセッサ (2.66GHz) × 1
メモリ	5 GB
OS	Red Hat Enterprise Linux ES4.6 (x86)

[BI サーバ]

CPU	クアッドコア Xeon プロセッサ (2.66GHz) × 2
メモリ	8 GB
OS	Red Hat Enterprise Linux ES4.6 (x86)

DB レイヤー: BX620 S4

CPU	クアッドコア Xeon プロセッサ (2.66GHz) × 2
メモリ	8 GB
OS	Red Hat Enterprise Linux AS4.6 (EM64T)

共用ストレージ: ETERNUS4000 モデル 300 [データベースおよび SAN ブート用]

コントローラ	2 本
メモリ	8 GB
キャッシュメモリ	4 GB x 2
ディスクドライブ	146 GB 15 krpm × 86 本

BI クラスタ間共有ストレージ: NR1000 F250

コントローラ	2 本
キャッシュメモリ	512 MB
ディスクドライブ	275 GB 15 krpm × 2 本

5.2. ソフトウェア構成

負荷生成サーバ

- Microsoft Windows Vista
- Oracle Application Testing Suite (8.2)

PS サーバ

- Red Hat Enterprise Linux ES4.6 (x86)
- Oracle Application Server 10g R3 for Linux x86
- Oracle Business Intelligence Suite Enterprise Edition 10.1.3.3 for Linux x86
 - Oracle Business Intelligence Presentation Services

BI サーバ

- Red Hat Enterprise Linux ES4.6 (x86)
- Oracle Application Server 10g R3 for Linux x86
- Oracle Business Intelligence Suite Enterprise Edition 10.1.3.3 for Linux x86
 - Oracle Business Intelligence Server

DB サーバ

- Red Hat Enterprise Linux AS4.6 (EM64T)
- Oracle Clusterware 11g Release 11.1.0.6
- Oracle Database 11g Enterprise Edition Release 11.1.0.6

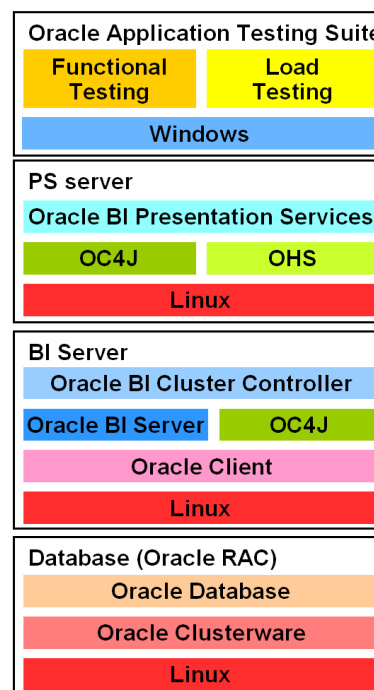


図 5-3 ソフトウェア構成

5.3. データベース構成

5.3.1. 初期化パラメータ

[6.2.6. BIサーバから発行されるSQL]にあるとおり、本検証においてBIサーバが発行する SQL はリテラルであるため、cursor_sharing=FORCE に設定しました。また、db_block_size は 16384 に設定しました。

5.3.2. ASM ディスクグループとユーザ

[5.1.2. ストレージ構成]で示したとおり、本検証では ETERNUS 上でディスク 4 本を 1 組とした RAID(1+0)グループを構成しました。この4本1組の RAID グループの1つを、1 つの ASM ディスクとしました。

ディスク追加によるスケーラビリティを円滑に測定するため、本検証においては都度ディスクを追加するのではなく、初期環境構築の段階で+BIDATA_DG(BIUSER ユーザ用)、+BI4R_DG(BI4R ユーザ用)という ASM ディスクグループを作成し、各ディスクグループ上に同じデータをロードしました。テストパターンに応じて DB ユーザを切り替えることで、使用するディスクの本数を変更しました。

表 5-1 ASM ディスクグループ構成

ASM ディスク グループ名	RAID グループ数 (ディスク本数)	備考
+BIDB_DG	2 (8)	SYSTEM, SYSAUX, REDO, UNDO, TEMP
+BIDATA_DG	2 (8)	ディスク 8 本を使用するディスクグループ
+BI4R_DG	4 (16)	ディスク 16 本を使用するディスクグループ

検証用データを格納する表領域は表 5-2 のように構成しました。なお、表内の「表領域名」列の()内は使用した ASM ディスクグループ名、「サイズ」列の()内は、<1 データファイルあたりのサイズ> * <ファイル数> を示します。

表 5-2 表領域構成

表領域名	サイズ	備考
BITBS (+BIDATA_DG)	320GB (16GB * 20)	ディスク 8 本に対し検証用スキーマをロード
BI4RTBS (+BI4R_DG)	320GB (16GB * 20)	ディスク 16 本に対し検証用スキーマをロード

5.4. キャッシュの種類

Oracle BIEEを使用したBIシステム環境には、図 5-4 に示すとおり、各レイヤーにキャッシュ機能が存在します。キャッシュを利用することにより、高速な処理が可能となります。

レイヤーによってキャッシングの方法および方針は異なります。本節では、次に挙げる各キャッシュの概要を説明します。

- PS サーバのキャッシュ
- BI サーバのキャッシュ
- データベース・バッファ・キャッシュ
- ストレージ・キャッシュ

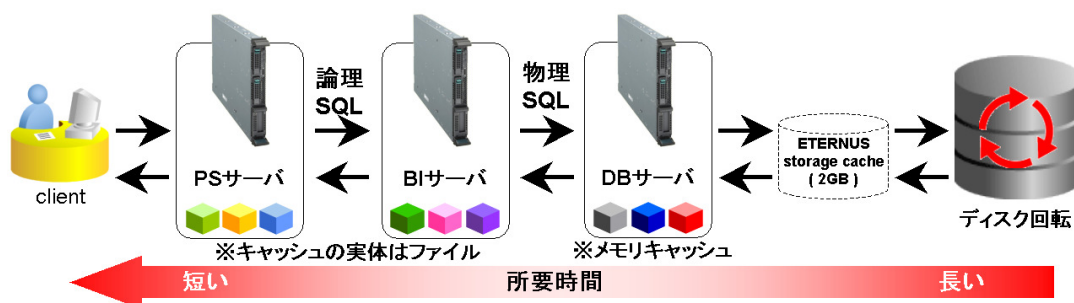


図 5-4 キャッシュの種類

5.4.1. 各キャッシュと処理の流れ

オラクル BI システム環境における各レイヤーのキャッシュとユーザリクエストの処理の流れのイメージは図 5-5 の通りです。

各レイヤーのキャッシュにヒットした時点で、その先のサーバへはリクエストが発行されない構成となっており、高速な処理を実現しています。

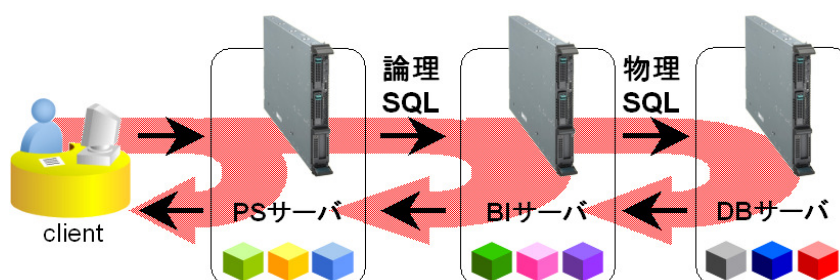


図 5-5 各キャッシュと処理の流れ

論理SQLとは、データベースに格納されているスキーマ構造に必ずしも基づかず、あくまでダッシュボードに表示するグラフ等の生成に必要な情報を得るための、Oracle BIEE が内部的に使用する論理的なSQLです。この論理SQLを、実際のスキーマ構造に基づいたSQL(これを物理SQLと呼びます)へ変換する役割をBIサーバが担います。BIサーバは論理SQLを基に、単一あるいは複数のデータソースに対し物理SQLを発行します。つまり論理SQLは1つでも、物理SQLは複数発行される場合があります。

例えば、地域別の売上推移を1つの表としてダッシュボードに表示する場合、PS サーバは表を生成するために1つの論理 SQL を発行します。BI サーバは論理 SQL を受け取り、売上と年月はデータベース A から取得し、顧客の情報はデータベース B から取得するといった、論理 SQL から物理 SQL への変換を行います。この例では BI サーバは2つの物理 SQL を発行し、その結果を1つにまとめ、論理 SQL の結果として PS サーバへ返します。

5.4.2. PS サーバのキャッシュ

PS サーバのキャッシュは、「問い合わせ」と「問い合わせの結果」をユーザ毎にキャッシュします。

ユーザが問い合わせを実行すると、PS サーバは論理 SQL を調べ、それがキャッシュされた既存の問い合わせに一致するか確認します。一致する場合、PS サーバは BI サーバに論理 SQL をリクエストせずに、キャッシュしていた結果をユーザへ戻します。また、ユーザは必要に応じて明示的に問い合わせをリフレッシュすることが可能です。

PS サーバのキャッシュは、instanceconfig.xml で設定する CacheMaxEntries パラメータで制御されます。

5.4.3. BI サーバのキャッシュ

BI サーバでは、個々の論理的な問合せとそのすべてのコンポーネントが保存され、受信する新しい問合せを分析し、キャッシュを使用して回答できるかを判断します。具体的には次の項目が保存されます。

- 論理 SQL (または他の問合せ言語) のテキスト
- 問合せの日時
- SQL で使用される物理表のリスト
- 問合せの結果

キャッシュの実体はファイルであり、キャッシュ・ファイルは、NQSCConfig.INI ファイルで設定している場所に作成されます。

5.4.4. データベース・バッファ・キャッシュとストレージ・キャッシュ

データベース・バッファ・キャッシュとは、DB サーバの Oracle インスタンスのシステムグローバル領域 (SGA) 上に確保されるメモリ領域であり、データファイルに格納されているテーブルなどのオブジェクトは、処理を行う際、データブロック単位でこのメモリ上に読み込まれます。

Oracle インスタンスは、サーバにリクエストされた SQL が解析された後、まずメモリ上に該当データがあるか検索します。メモリ上にデータが存在しなければ、ディスク上のデータファイルから読み込み、データベース・バッファ・キャッシュ上に保持します。一方、メモリ上にデータが存在する場合は、ディスクからの読み込みは行わず、直接バッファキャッシュ

ュ上のデータを処理するため、Disk I/O は大幅に削減されます。

データベース・バッファ・キャッシュは、頻繁にアクセスされるデータブロックをできるだけキャッシュ上に保持する仕組みになっています。以前アクセスしたデータがサーバのメモリから掃き出されても、ストレージのディスクキャッシュにはデータが保持されたままである可能性があります。ストレージのディスクキャッシュにデータが保持されている場合は、ディスク読み込みを行うよりも高速にデータを読み込むことができます。

6. 検証モデル

本章では、検証に使用した業務モデル、データベーススキーマ、DB サーバへ実行されるクエリについての概略を説明します。

6.1. 業務モデル

本検証では、製造業における「経営の見える化」による企業経営の高度化と企業活動の最適化を目的として下記のような業務モデルを前提としています。ツールは計画立案支援パッケージである「GLOVIA/SCP」、基幹業務パッケージである「GLOVIA/Process C1」及び、各種情報分析ツールである「Oracle BI」を使用しています。図 6-1 は、計画～販売～分析における業務およびデータの流れを表しています。

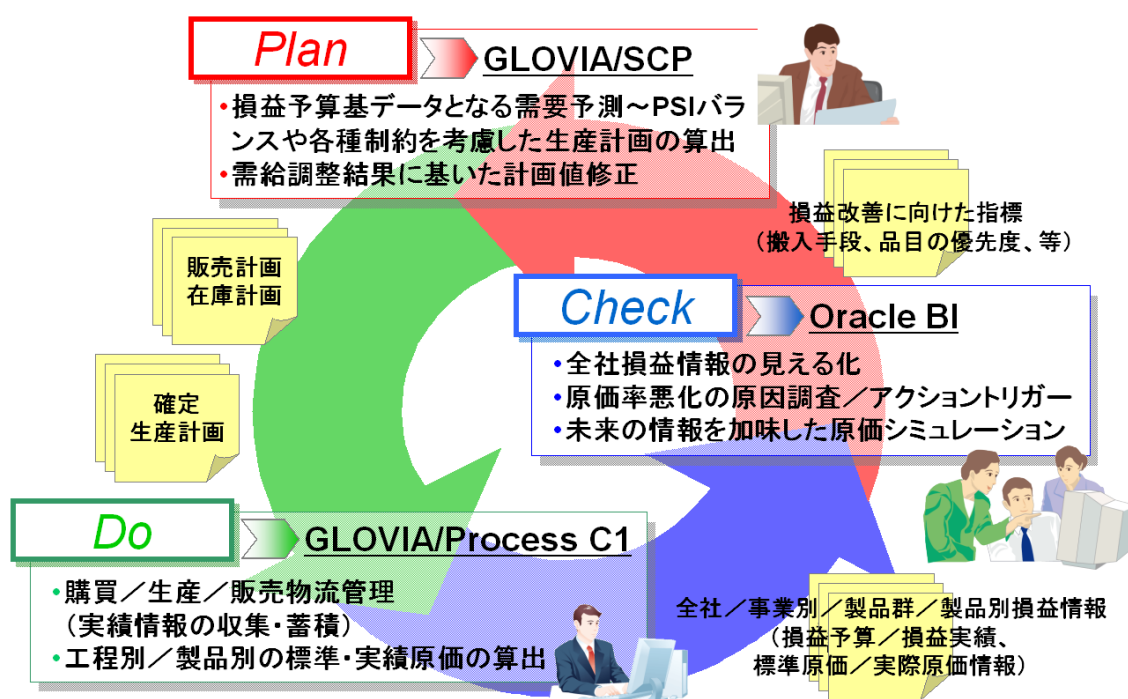


図 6-1 GLOVIA のデータの流れ

「実績データ」及び「生産設備制約」「物流ネットワーク」を加味した上で、将来の「原価要素の変動」や「物流輸送費」の損益シミュレーションを実施することによって、計画立案の精度向上・作業効率化を実現することができます。

システム連携イメージは図 6-2 のとおりです。

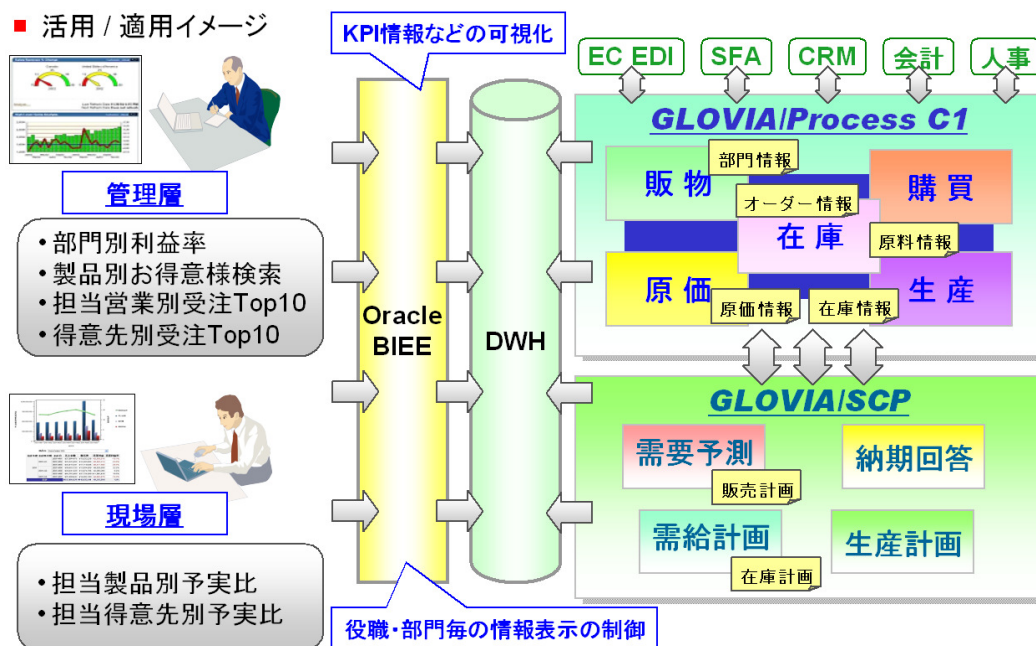


図 6-2 GLOVIA のシステム連携

GLOVIA/Process C1 及び GLOVIA/SCP からデータを抽出して DWH を作成し、Oracle BI を使用してそのデータの分析を行うことで PDCA サイクルの最適化を図ります。役職・部門毎に必要なデータは異なるため、Oracle BIEE を使用して表示する情報を制御します。

本検証では営業社員などの「現場層」、及び部門マネージャなどの「管理層」の検索モデルを想定しています。現場層においては多数のユーザが自分の担当している製品別や顧客別の予実比を確認します。管理層においては、次期経営戦略立案のため、各部門別の営業利益率、自部門の製品別お得意様、担当営業別、得意先別の受注額を確認し分析を行うシーンを想定しています。

6.2. 検証用モデル

[6.1. 業務モデル]で述べたとおり、本検証ではオラクル BI システムのユーザとして現場層、管理層に属す社員を想定しています。それぞれについて詳細を述べます。

6.2.1. 現場層による BI システムの使用傾向

現場層による BI システム使用傾向として、まず、同時にログインする人数が管理層と比較して多く、使用頻度が高いことが考えられます。これは管理層より現場層の方が絶対数が多いこと、また現場層の BI システム使途として、直近の取引明細の出力、毎時の生産量の確認など、後述する管理層の使途と比較すると使用頻度が高くなると考えられるからです。

また、現場層によるクエリーの検索範囲は、管理層のそれと比較して狭いことが考えられます。これはアクセス制御により、原則として現場層は自身の担当する、もしくは自分と関連のある限られたデータしか参照できないこと、また前述の通り直近のデータを参照

する傾向があると想定したからです。

以上から現場層の発行するクエリーは、狭い範囲のデータにアクセスすると仮定し、以降はこのクエリーを「小範囲クエリー」と呼びます。

6.2.2. 管理層による BI システムの使用傾向

管理層による BI システムの使用傾向として、検索範囲が広いことが想定されます。これは、管理層と位置づけた部門マネージャは、自分自身のみならず、部下のデータも参照し部門全体としての分析を行うと考えられるからです。経営層ともなれば、会社全体のデータを参照できなければ、全社を俯瞰した的確な経営判断材料を捻出できません。また、管理層による分析は、四半期ごと、年次ごと、それらを複数実行し行う時系列比較など、現場層と比較し長い期間を対象範囲とする傾向があると考えられます。以上の理由より、管理層が検索対象とするデータの範囲は、現場層の何倍にもなることが容易に考えられます。

一方で、一度にログインする人数、使用頻度は、現場層と比較して管理層の方が少ないと考えられます。これは、管理層はある程度の量の情報が蓄積された後に情報の参照、分析を行うと想定されるため、現場層よりはシステムの使用頻度が低いと考えられるからです。

以上から管理層の発行するクエリーは、広い範囲のデータにアクセスすると仮定し、以降はこのクエリーを「広範囲クエリー」と呼びます。

6.2.3. 小・広範囲クエリーによる各レイヤーのリソース消費

小範囲クエリーによるオラクル BI システムの使用では、アクセス頻度及び同時実行ユーザ数の多さから、リクエストの受付および結果の表示を担う PS サーバ、論理クエリーから物理クエリーへの変換を行う BI サーバ、すなわちこれら Oracle BIEE レイヤーのリソースを大量に消費するもの、と仮定しました。現場層による検索は検索範囲が狭いため、データベースから読み出すレコードは少なく、Oracle BIEE レイヤーのキャッシュの効果もあるため、後述する管理層によるシステム利用よりも、データベースレイヤーへの負荷は軽いものと想定しました。

広範囲クエリーによるオラクル BI システムの使用では、アクセス頻度及び同時実行ユーザ数については、現場層によるシステム利用よりも低いとしました。Oracle BIEE レイヤーのリソース消費はデータベースレイヤーに比べ少なく、かつ Oracle BIEE レイヤーのキャッシュが効かないと仮定しました。データベースレイヤーについては、検索範囲が現場層による小範囲クエリーと比較し広いため、ディスクから読み出す行数も多くその分負荷がかかると想定しました。

以上を図にまとめたのが図 6-3 になります。

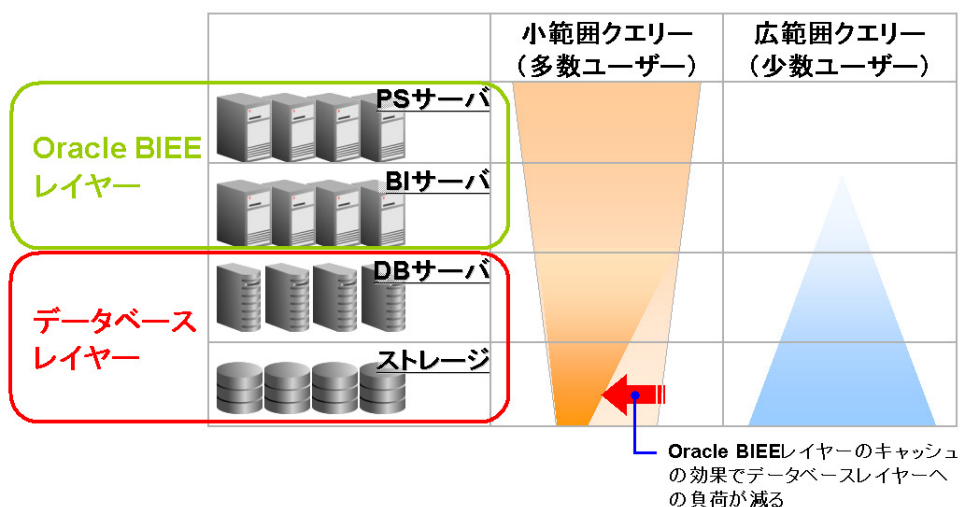


図 6-3 リソースの消費傾向の想定

エンタープライズ BI システムは大量のデータを扱う上に、ユーザ数も多くなります。そのため、各サーバレイヤーのハードウェアリソースの消費傾向を考慮しなければなりません。本検証で想定する小範囲クエリと広範囲クエリでは、Oracle BIEE レイヤーのキャッシュの効果とユーザ数の違いにより、負荷がかかるレイヤーが異なります。これらハードウェアリソースの消費傾向の異なる複数のクエリを使用し、BI システムの広い用途に対応するために各レイヤーのスケーラビリティを検討します。

6.2.4. データベース・スキーマ

本検証で使用したテーブル、および各テーブルの関係は図 6-4 のとおりです。

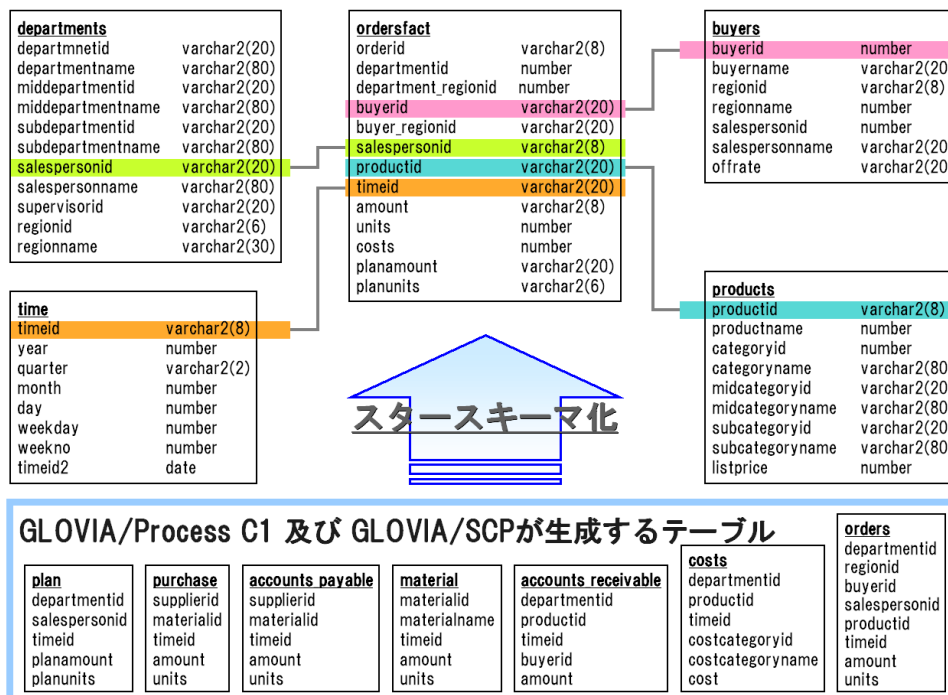


図 6-4 本検証で用いたテーブルとテーブル間の関係

GLOVIA/Process C1 及び GLOVIA/SCP にて生成されるレコードに対して ETL 処理を施し、ordersfact テーブルを中心としたスター・スキーマへ格納しました。各要素の詳細はディメンション表のキーを基に参照され、ordersfact 表は各ディメンションのキー列を含有します。本検証で用いたテーブルの概要及び ordersfact 表の詳細は次のとおりです。

表 6-1 本検証で用いたテーブルの概要

表名	キー	説明
departments	salespersonid	担当営業とその部署
time	timeid	年、月、日など日付情報
buyers	buyerid	顧客名、割引率など
products	productid	製品名、製品カテゴリ(大・中・小)
ordersfact	orderid	担当営業、売上年月日、顧客、品名、売上個数等のサマリー
		2004 年 1 月から 2009 年 12 月までの 6 年分のデータを格納 - 表サイズ約 245G(1 日あたり 100 万件、1 ヶ月あたり 2800 万件(約 3.4G)) 1 ヶ月単位でレンジパーティション化し、各パーティションを 4 つにハッシュ・サブパーティション化したレンジ-ハッシュ・コンポジット・パーティション表

6.2.5. 業務シナリオと BI 画面

本検証において想定した業務シナリオは次のとおりです。

1. ログイン画面へアクセス
2. ID とパスワードを入力してログイン
3. 検索条件の異なるグラフを複数回表示する

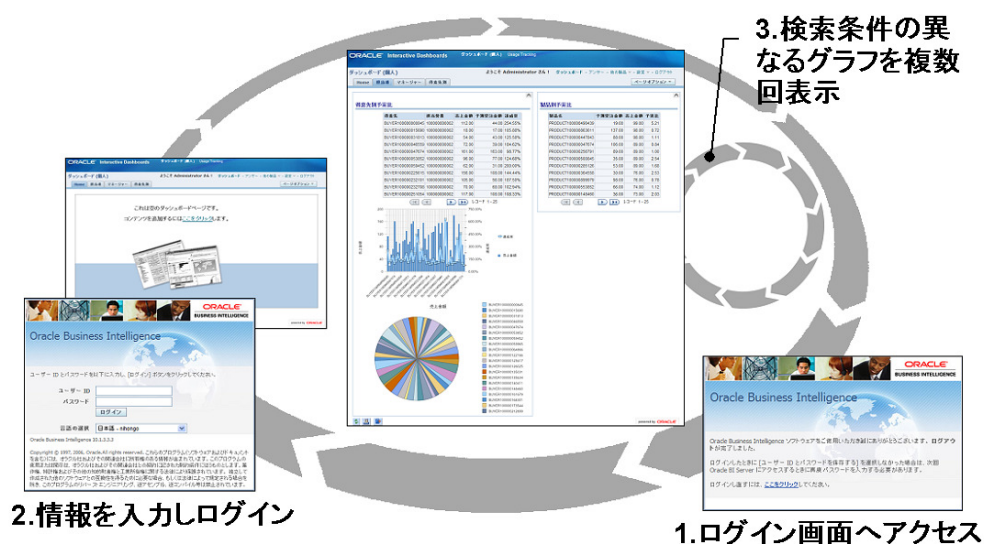


図 6-5 業務シナリオと BI 画面

6.2.6. BI サーバから発行される SQL

ユーザが小範囲検索用の BI 画面へアクセスすると、BI サーバから次の 2 つの SQL がデータベースへ発行されます。<任意の日>等の記述になっている箇所は、実際 BI から発行される際リテラル値が入ります。

小範囲クエリー (1)

```
SELECT distinct D1.c3 as c1, D1.c1 as c2, D1.c2 as c3,  
D1.c2 / NULLIF(D1.c1, 0) as c4  
FROM (  
    SELECT SUM(T862.PLANAMOUNT) as c1,  
    SUM(T862.AMOUNT) as c2, T882.PRODUCTNAME as c3  
    FROM PRODUCTS T882, TIME T898, ORDERSFACT T862  
    WHERE (T862.PRODUCTID = T882.PRODUCTID  
    AND T862.SALESPERSONID = <任意の社員番号>  
    AND T862.TIMEID = T898.TIMEID  
    AND T898.DAY = <任意の日>  
    AND T898.MONTH = <任意の月>  
    AND T898.YEAR = <任意の年> )  
    GROUP BY T882.PRODUCTNAME) D1  
ORDER BY c3 desc;
```

小範囲クエリー (2)

```
SELECT distinct D1.c9 as c1, D1.c3 as c2, D1.c2 as c3,  
D1.c1 as c4, D1.c2 / NULLIF(D1.c1, 0) * 100 as c5,  
D1.c4 as c6, D1.c5 as c7, D1.c6 as c8,  
D1.c8 / NULLIF(D1.c7, 0) * 100 as c9  
FROM (  
    SELECT D1.c1 as c1, D1.c2 as c2, D1.c3 as c3, D1.c4 as c4,  
    D1.c5 as c5, D1.c6 as c6,  
    SUM(D1.c1) over (partition by D1.c9) as c7,  
    SUM(D1.c2) over (partition by D1.c9) as c8,  
    D1.c9 as c9  
    FROM (  
        SELECT SUM(T862.PLANAMOUNT) as c1,  
        SUM(T862.AMOUNT) as c2,  
        T812.SALESPERSONID as c3,  
        T898.YEAR as c4, T898.MONTH as c5,  
        T898.DAY as c6, T812.BUYERNAME as c9  
        FROM TIME T898, BUYERS T812, ORDERSFACT T862  
        WHERE (T812.BUYERID = T862.BUYERID  
        AND T812.SALESPERSONID = <任意の社員番号>  
        AND T862.TIMEID = T898.TIMEID  
        AND T898.DAY = <任意の日>  
        AND T898.MONTH = <任意の月>  
        AND T898.YEAR = <任意の年> )  
        GROUP BY T812.BUYERNAME, T812.SALESPERSONID,  
        T898.DAY, T898.MONTH, T898.YEAR) D1) D1;
```

また、広範囲検索用の BI 画面へアクセスすると、BI サーバから次の 1 つの SQL がデータベースへ発行されます。

広範囲クエリー

```
SELECT D1.c1 as c1, D1.c2 as c2
FROM (
  SELECT D1.c2 as c1, D1.c1 as c2,
  CASE WHEN D1.c2 is not null then Rank() OVER (
    ORDER BY D1.c2 DESC NULLS LAST) end as c3
  FROM (
    SELECT SUM(T862.AMOUNT) as c1, T812.BUYERNAME as c2
    FROM TIME T898, BUYERS T812, ORDERSFACT T862
    WHERE (T812.BUYERID      = T862.BUYERID
           AND T862.TIMEID   = T898.TIMEID
           AND T898.MONTH    =<任意の月>
           AND T898.YEAR     =<任意の年>)
    GROUP BY T812.BUYERNAME) D1) D1
  WHERE (D1.c3 <= 10)
ORDER BY c2 desc;
```

7. ノード追加による BI システムのスケラビリティ

[1. 検証目的]でも述べたとおり、BI システムは複数のレイヤーに渡ってリソースを消費します。今日までの検証では、特定のレイヤーに着目して行われることがほとんどでしたが、本検証では PRIMERGY ブレードサーバおよびストレージシステム ETERNUS の高い拡張性を生かし、全レイヤーにおいてスケールアウトによって処理性能が向上するのか確認しました。

7.1. Oracle BIEE のスケラビリティ

7.1.1. 検証目的

本節の検証では、小範囲クエリー(現場層向け)を想定した BI システムに対し Oracle BIEE レイヤーへノード追加を行います。本節の検証の目的は、Oracle BIEE レイヤーに負荷がかかる場合において、上述のノード追加によって性能が向上すると確認することです。

まず、PS サーバと BI サーバがそれぞれ 1 台の構成で、基準となる処理性能(最適 VU、スループット、レスポンスタイム等)を測定しました。しかしユーザがどのようなページ(データ)を頻繁に表示するかによって、PS サーバのキャッシュ、BI サーバのキャッシュやデータベース・バッファ・キャッシュのキャッシュ・ヒット率が異なり、処理性能が大きく異なることが予測されました。データベース・レイヤーのリソースを飽和させず、Oracle BIEE レイヤーのリソース追加の影響を調べるために小範囲クエリーを用いました。

以上より基準となる処理性能の測定では、Oracle BIEE レイヤーに着目しました。また PS サーバ、BI サーバの両方に存在するキャッシュについて、最も単純な構成を評価するために BI サーバのキャッシュを無効にし、PS サーバのキャッシュ・ヒット率に着目して処理性能の傾向を評価しました。

次に、図 7-1 に示すように、基準となる処理性能とその構成に対してユーザ数を増大させるとともに Oracle BIEE レイヤーをスケールアウトさせることでスケラビリティを評価します。

以降、検証方法と検証結果について順に報告します。

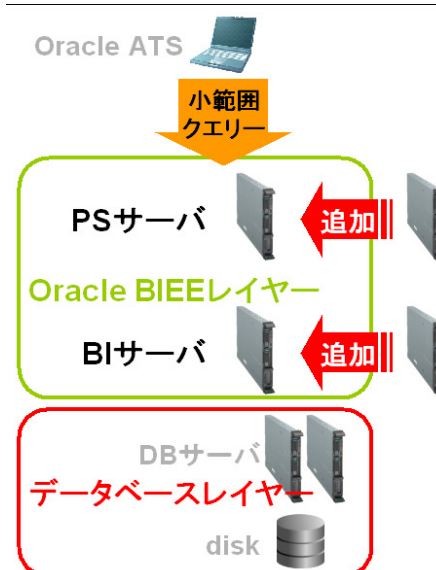


図 7-1 BI サーバ・レイヤーのスケールアウト

7.1.2. 検証方法

現場層に属す多数の社員が BI システムにログインし任意のデータを参照するという一連の動作をシミュレートするために、本節の検証では Oracle Application Testing Suite (以降、Oracle ATS) を活用しました。

[4.3. Oracle Application Testing Suite] で示したとおり、Oracle ATS 製品群には、画面遷移記録用ツール「Oracle Functional Testing for Web Applications」と、その記録した画面遷移を利用して負荷を生成する「Oracle Load Testing for Web Applications」があります。

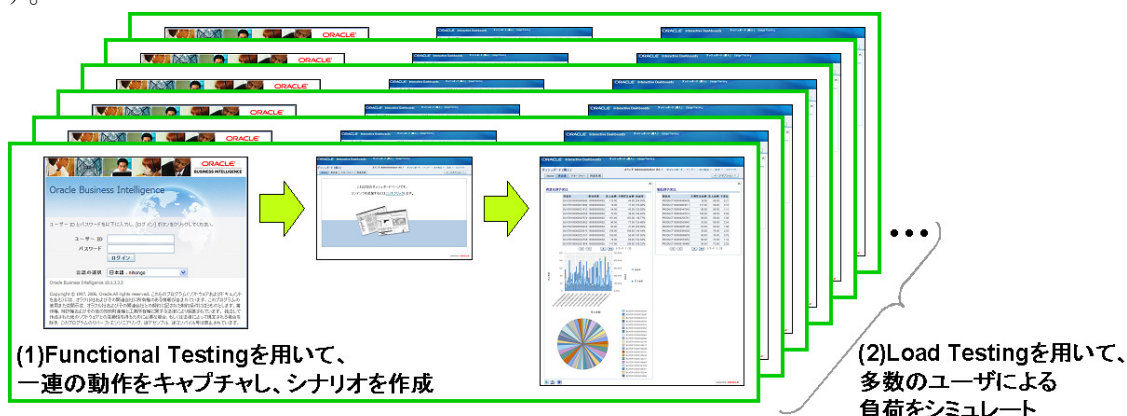


図 7-2 Oracle ATS のシナリオ作成と負荷生成

7.1.2.1. Functional Testing を用いたシナリオの作成

まず、現場層に属す 1 人の社員が BI 画面を操作する際の一連の動作を次のように仮定し、Oracle Functional Testing for Web Applications を用いて記録しました。記録したものを「シナリオ」と呼称します。この作業は図 7-2 の(1)に該当します。

- (1) ログイン画面にアクセスしユーザ名とパスワードを入力
- (2) 担当営業と日付を条件指定してダッシュボードを表示
- (3) (2)を、担当営業と日付の組み合わせを変更し、終了またはエラー発生まで繰り返し

ダッシュボードとは、ビジネスに関する様々な指標をグラフなどビジュアル的な要素を用いて画面上に表示する機能です。ダッシュボードに表示する情報は各ユーザが恣意的に設定できますが、本検証では担当営業と日付を含む 2 つのグラフを表示するよう固定しました。その結果[6.2.6. BI サーバから発行される SQL]が BI サーバによって生成されました。

次にシナリオを負荷テストに適した形へ編集します。具体的には、担当営業と日付を変数とし、外部データ(CSV ファイル)から逐次読み込むように設定します。この設定により、(3)を自動化できます。

7.1.2.2. シナリオと Load Testing による負荷生成、「最適」VU

編集したシナリオを Oracle Load Testing for Web Applications を用いて実行します。この作業は図 7-2 の(2)に該当します。この際、多数のユーザのログインをシミュレートしますが、このシミュレートされたユーザを仮想ユーザ(Virtual User、以降 VU)と呼びます⁵。

本検証では 10 秒経過毎に 5VU ずつ VU を追加し、レスポンスタイムが急激に悪化した時点の VU 数を「最適」VU 数と定義しました。

負荷テストのレスポンスタイムと VU 数のグラフ例を図 7-3 に示します。この時点の数値を採用したのは、レスポンスタイムが悪化したということは、何かしらのリソースが飽和し始めたか、もしくはリソース競合が発生し始めたと考えられるからです。ただし本検証において、この最適 VU 時のレスポンスタイムが 3 秒超になることはほとんどありませんでした。お客様のシステム要件によりませんが、3 秒超のレスポンスタイムが許容されるのであれば、より多くの VU をログインさせることが可能です。その意味で「最適」VU は「最大」VU ではありません。図 7-3 から、レスポンスタイムは悪化しますが最適 VU 以上のユーザがシステムを利用可能であることが分かります。

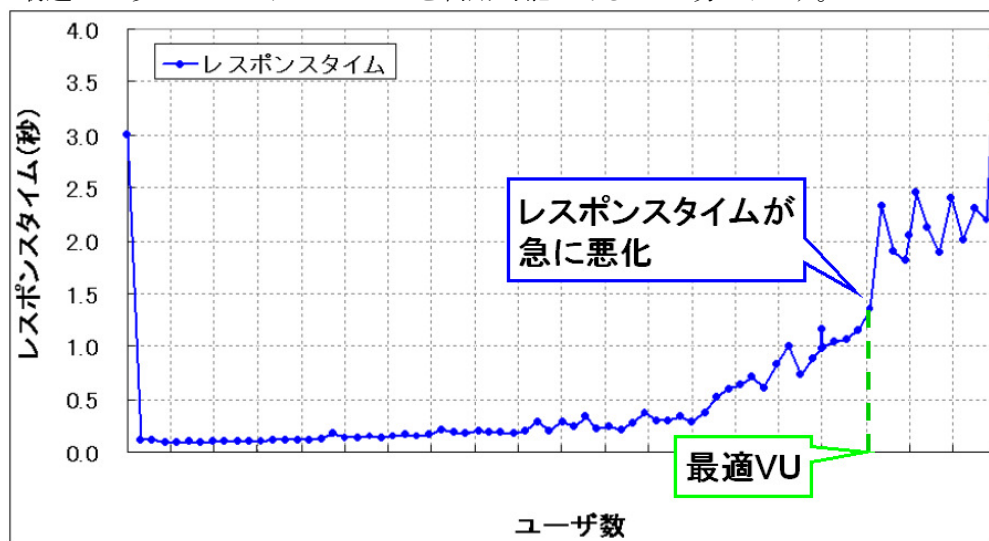


図 7-3 最適 VU

⁵具体的には、Oracle Load Testing for Web Applications のエージェントマシンの JVM 上に複数スレッドを起動し、それぞれがユーザ 1 人の動作を多重並列処理で実行することによって、負荷を生成します。

7.1.3. PS サーバのキャッシュ・ヒット率毎の性能

各ユーザが同一のページ(データ)を表示する頻度を調整することによって、PS サーバのキャッシュ・ヒット率は変化します。本節では、PS サーバのキャッシュ・ヒット率と処理性能の関係を測定した結果について述べます。なお、本項の検証では、PS サーバが 1 台、BI サーバが 1 台、DB サーバが 2 台の構成を対象としています。

まず最適 VU (Users) に対するスループット (TPS) を比較すると、図 7-4 にあるように、キャッシュ・ヒット率が高い場合には最適 VU とスループットが大きいという結果が出ています。

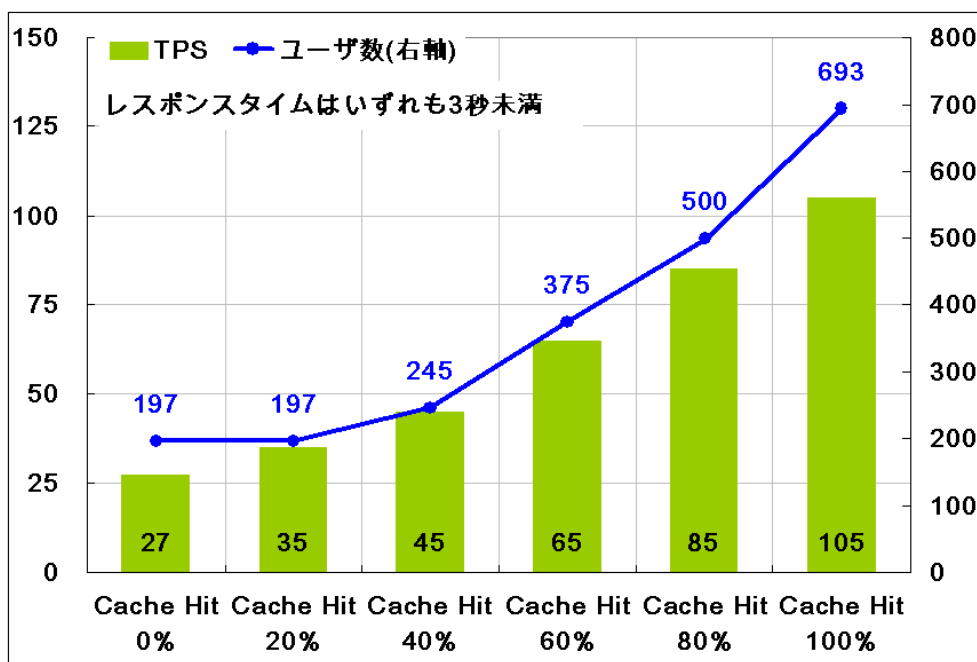


図 7-4 キャッシュ・ヒット率別のユーザ数対スループット

では、図 7-4 の処理性能を出すために BI システムを構成する各サーバのリソースはどの程度消費されたのか、スループットに大きく影響する CPU 使用率の傾向を図 7-5 で確認します。

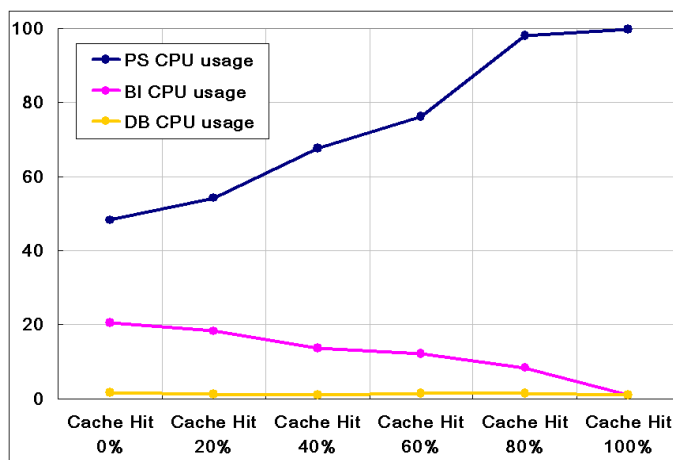


図 7-5 PS サーバのキャッシュ・ヒット率と CPU 使用率

全体的な傾向として、PS サーバのキャッシュ・ヒット率によらず、BI サーバの CPU 使用

率(BI cpu usage)と DB サーバの CPU 使用率(DB cpu usage)は低く、リソースに余裕があります。一方で PS サーバの CPU 使用率(PS cpu usage)は高い傾向にあります。

PS サーバのキャッシュ・ヒット率ごとの PS/BI/DB サーバの CPU 使用率に着目すると、キャッシュ・ヒット率が高いほど PS サーバの CPU 使用率が高く、逆にキャッシュ・ヒット率が低いほど BI サーバの CPU 使用率が高くなっています。

キャッシュ・ヒット率の高低による動作の違いを図示すると図 7-6 のようになります。キャッシュ・ヒット率が高いと、PS サーバのみでリクエストの結果を返せる割合が高くなるため、PS サーバの CPU 使用率が上がります。他方でキャッシュ・ヒット率が低い場合は、BI サーバ及びデータベースレイヤーへ更にリクエストを発行するため、BI サーバの CPU 使用率が上がります。しかし PS サーバは、BI サーバからの結果の待機時間が長くなる分 CPU 使用率が下がります。

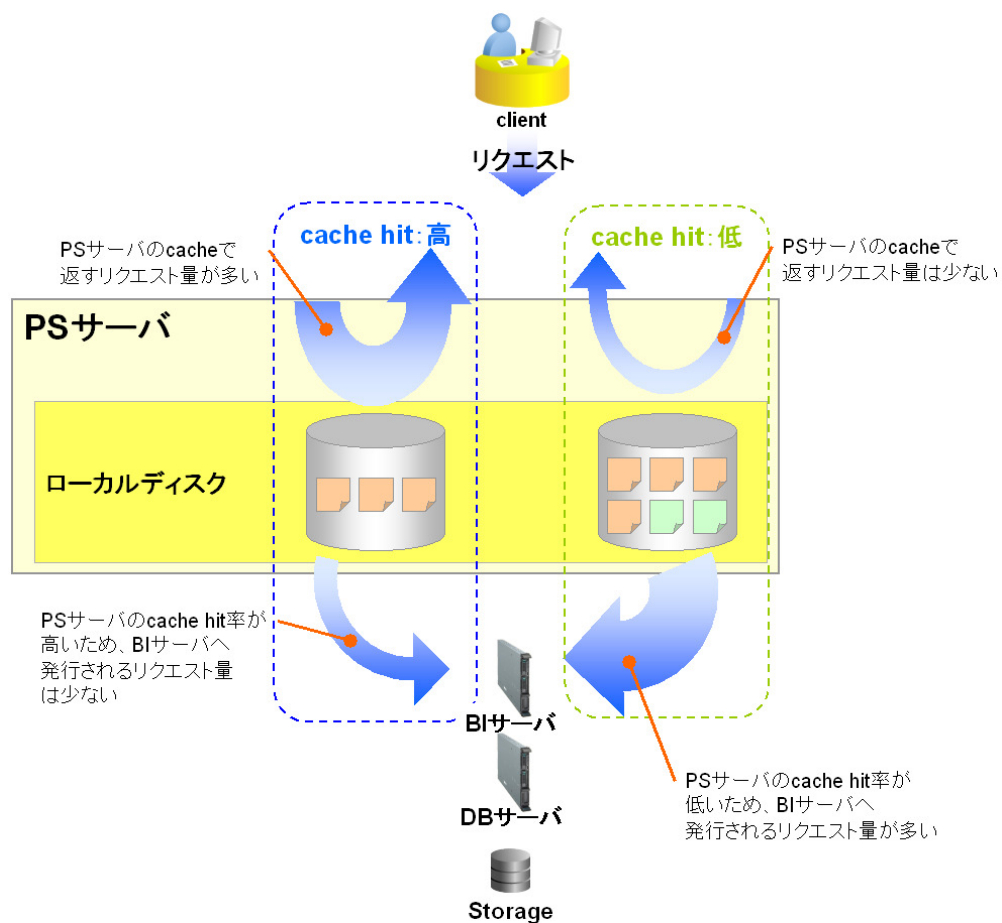


図 7-6 PS サーバのキャッシュ

以降の PS サーバおよび BI サーバのスケールビリティ検証では、CPU の使用率が約 80%となるキャッシュ・ヒット率 60%の負荷パターンを基準としました。

7.1.4. PS サーバのスケールビリティ

本節では、PS サーバのキャッシュ・ヒット率を 60% に調整した構成を基準としてスケールビリティを検証した結果を述べます。なお、本項の検証では BI サーバが 2 台、DB サーバが 2 台の構成を対象とし、PS サーバの台数を変更して比較しています。

まず最適 VU (Users) に対するスループット (TPS) を比較すると、図 7-8 にあるように PS サーバを増設するにつれて、最適 VU とスループットが増加すると分かります。

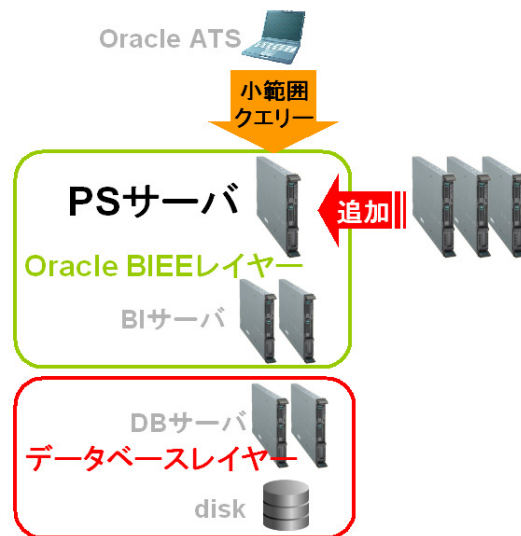


図 7-7 検証イメージ

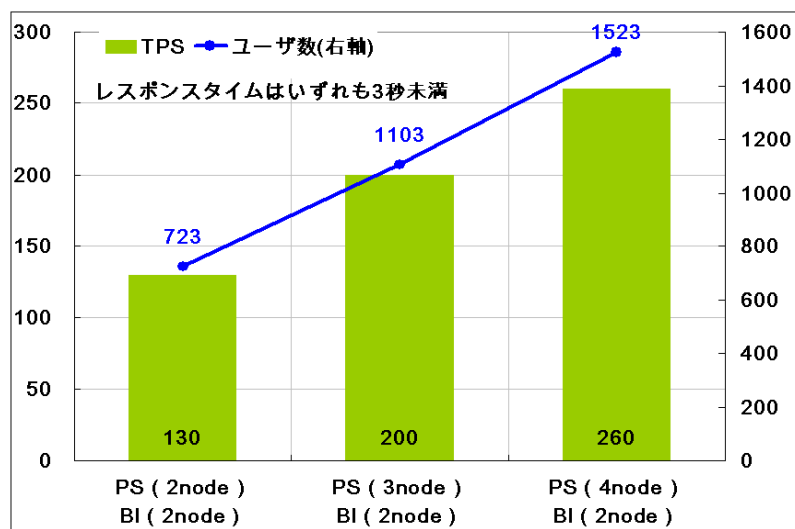


図 7-8 PS サーバノード追加とユーザ数/スループット

では、実際にこのような処理性能を出すために、BI システムを構成する各サーバの OS リソースはどの程度消費されたのか、スループットに大きく影響する CPU 使用率の傾向を図 7-9 で確認します。

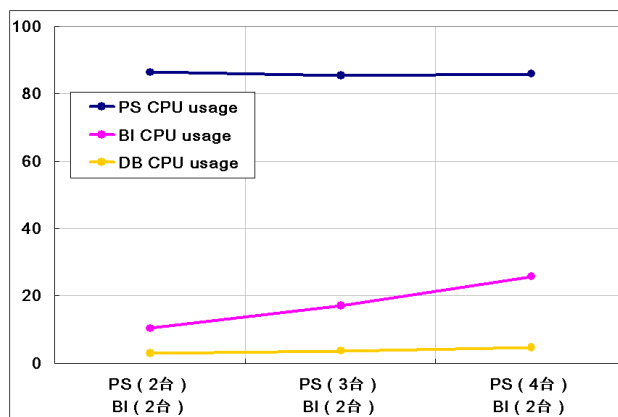


図 7-9 PS サーバノード追加とCPU使用率

全体的な傾向として、PS サーバの CPU 使用率(PS cpu usage)はほぼ 80%から 90% 程度を安定して推移しています。BI サーバの CPU 使用率(BI cpu usage)と DB サーバの CPU 使用率(DB cpu usage)についてはノード数が一定のため、PS サーバの増設にともない CPU 使用率が増加している傾向が見られます。

上記の結果として、オラクル BI システムを構成する BI サーバおよびデータベース・レイヤー上にボトルネックが発生しておらず、PS サーバがボトルネックの場合には、PS サーバのスケールアウトによる処理性能の高いスケールビリティが確認できました。

7.1.5. BI サーバのスケールビリティ

本節では、PS サーバのキャッシュ・ヒット率を 60%に調整した構成を基準として BI サーバが 1 ノードの場合と 2 ノードの場合を比較し、ユーザ数を増大させたときのスケールビリティを検証します。なお、本項の検証では DB サーバが 2 台の構成を対象としています。

まず最適 VU (Users) に対するスループット (tps) を比較します。BI サーバが 1 ノード構成の場合は図 7-11 のとおり、PS サーバを 4 台に増設した段階で期待した処理性能まで増加していないことが分かります。



図 7-10 検証イメージ

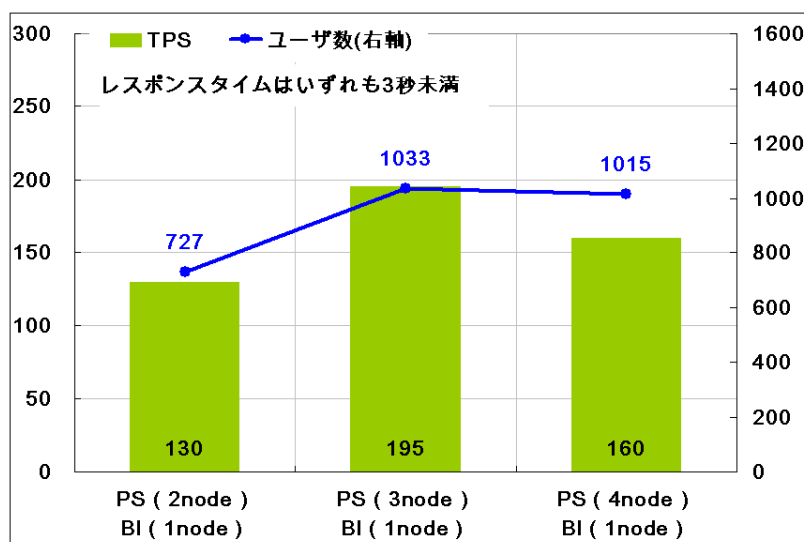


図 7-11 PS サーバの増設とユーザ数/スループット (BI サーバ 1 ノード)
※レスポンスタイムについてはどのパターンも 3 秒以内の性能

他方、BI サーバが 2 ノード構成の場合は、図 7-12 のとおり期待した処理性能まで増加しています。

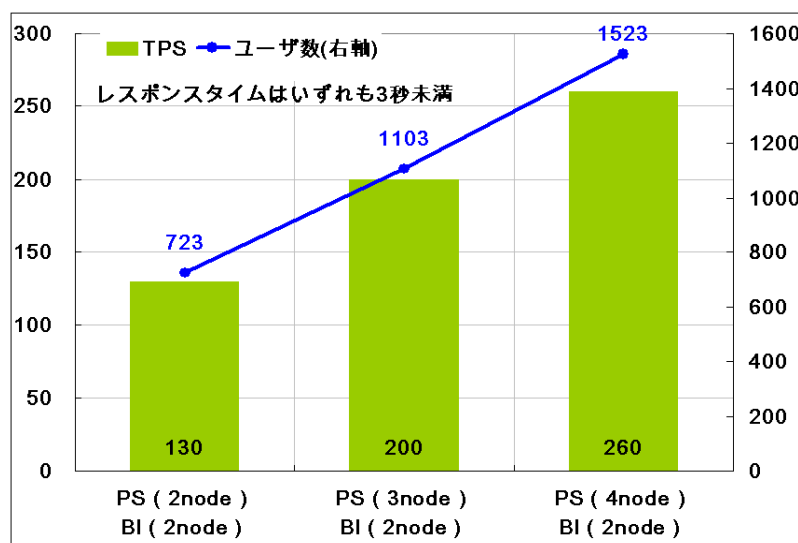
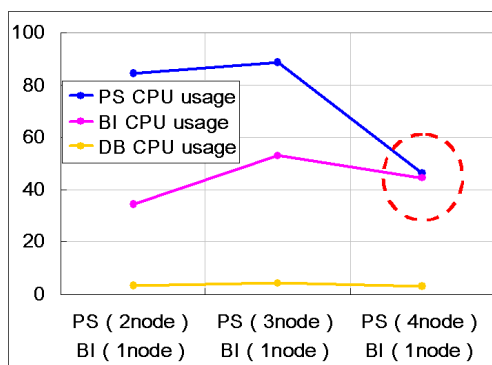
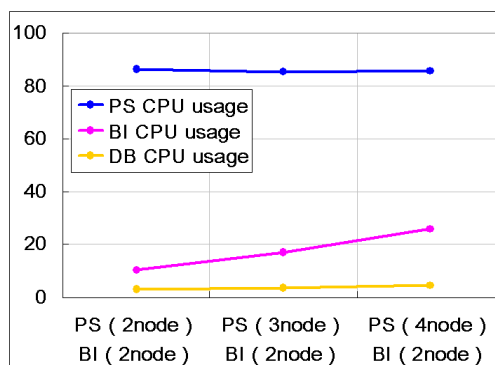


図 7-12 PS サーバの増設とユーザ数/スループット(BI サーバ 2 ノード)

では、実際にこのような処理性能を出すために BI システムを構成する各サーバの OS リソースがどの程度消費されていたかについて、スループットに大きく影響する CPU 使用率の傾向を図 7-13 と図 7-14 で確認します。

図 7-13 PS サーバの増設と
CPU 使用率(BI サーバ 1 ノード)図 7-14 PS サーバの増設と
CPU 使用率(BI サーバ 2 ノード)

BI サーバが 1 ノード構成の場合は、図 7-13 のとおり、DB サーバの CPU 使用率(DB cpu usage)はほとんど変化なく、PS サーバが 3 台までは期待した CPU 使用率(PS cpu usage)になっています。しかし PS サーバを 4 台に増設した段階で BI サーバの CPU 使用率(BI cpu usage)が 50%から増加しておらず、PS サーバの CPU 使用率も低下しています。

翻って BI サーバが 2 ノード構成の場合、図 7-14 のとおり、PS サーバが 4 台の構成においても PS サーバの CPU 使用率は安定しており、BI サーバの CPU 使用率も PS サーバの増設にともない増加していることが分かります。また、図 7-13 と図 7-14 の BI サーバの CPU 使用率を比較すると、BI サーバのノード数を 1 から 2 に増加することで、BI サーバの負荷が分散され CPU 使用率が減少していることも分かります。

上記の結果から、BI サーバが 1 ノード構成の場合には PS サーバ 4 ノード分のユーザ

数まで増加させると BI サーバ内で何らかのリソース競合が発生しており、BI サーバが 2 ノード構成になることによってリソース競合が解消され処理性能が向上したと考えられます。このことから、BI サーバのスケールアウトによる処理能力向上が確認できたといえます。

Oracle BIEE のスケールビリティについて、[7.1.4. PS サーバのスケールビリティ]および [7.1.5. BI サーバのスケールビリティ]の検証結果から、ユーザ数の増大に応じてオラクル BI システムを構成する PS サーバと BI サーバをスケールアウトすることで、処理能力を向上可能なことが確認できました。

7.2. Oracle RAC のスケールビリティ

7.2.1. 検証目的

前節までの検証で用いた多数のユーザによる小範囲クエリーでは、ユーザー数を増加させていくと、データベース・レイヤーのリソースが枯渇する前に PS サーバおよび BI サーバ、つまり Oracle BIEE レイヤーのリソースが枯渇する傾向があると確認できました。それは DB サーバの OS 統計と PS・BI サーバの OS 統計を比較しても明らかでした。

しかし少数のユーザで膨大な量のデータを読み込む広範囲クエリーでは、Oracle BIEE レイヤーよりも、データベース・レイヤーへ負荷がかかると予測されます。そこで本節の検証では、データベース・レイヤーのリソース追加により性能向上が可能なのかを確認めます。

7.2.2. 検証方法

[6.2.6. BI サーバから発行される SQL]で述べた広範囲クエリーとはほぼ同等のクエリーを、Java のカスタムアプリケーションを用い、直接 Oracle RAC に JDBC 接続して実行しました。

使用ディスク本数の増減は [5.3.2. ASM ディスクグループとユーザ]で述べた BIUSER ユーザ、および BI4R ユーザの切り替えにより再現しました。

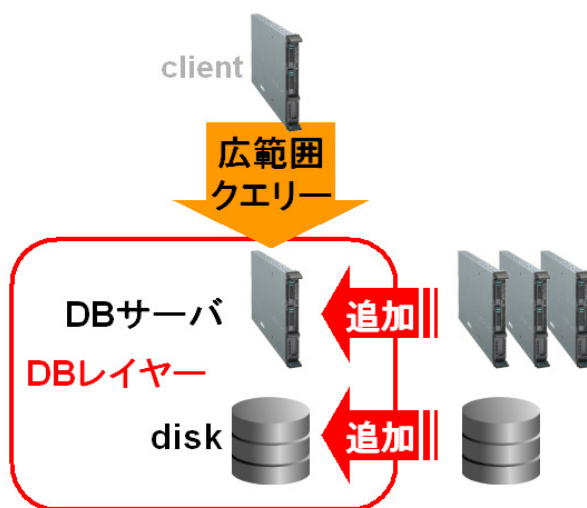


図 7-15 Oracle RAC スケールビリティの検証方法

7.2.3. 検証結果: Oracle RAC ノード追加による性能向上

BIUSER ユーザで広範囲クエリーを実行した際の結果は図 7-16 のとおりです。

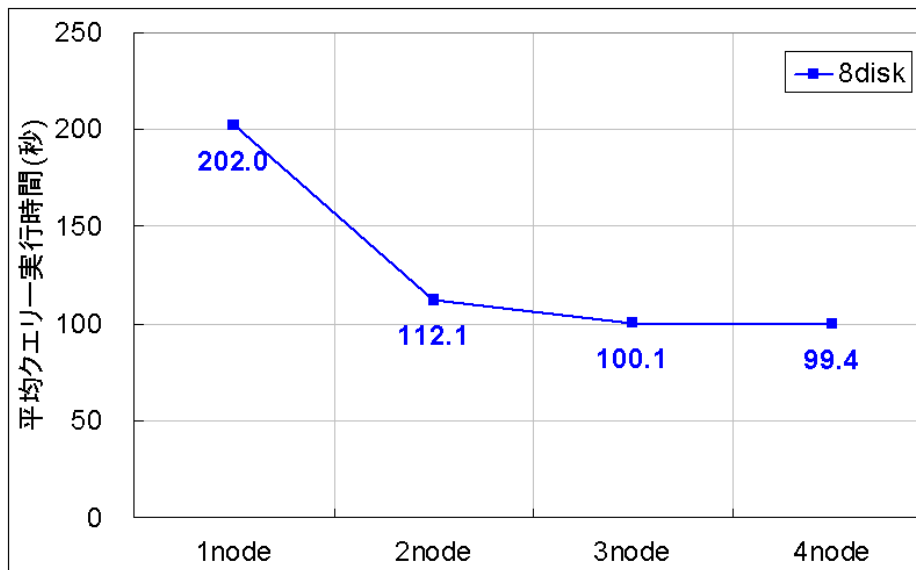


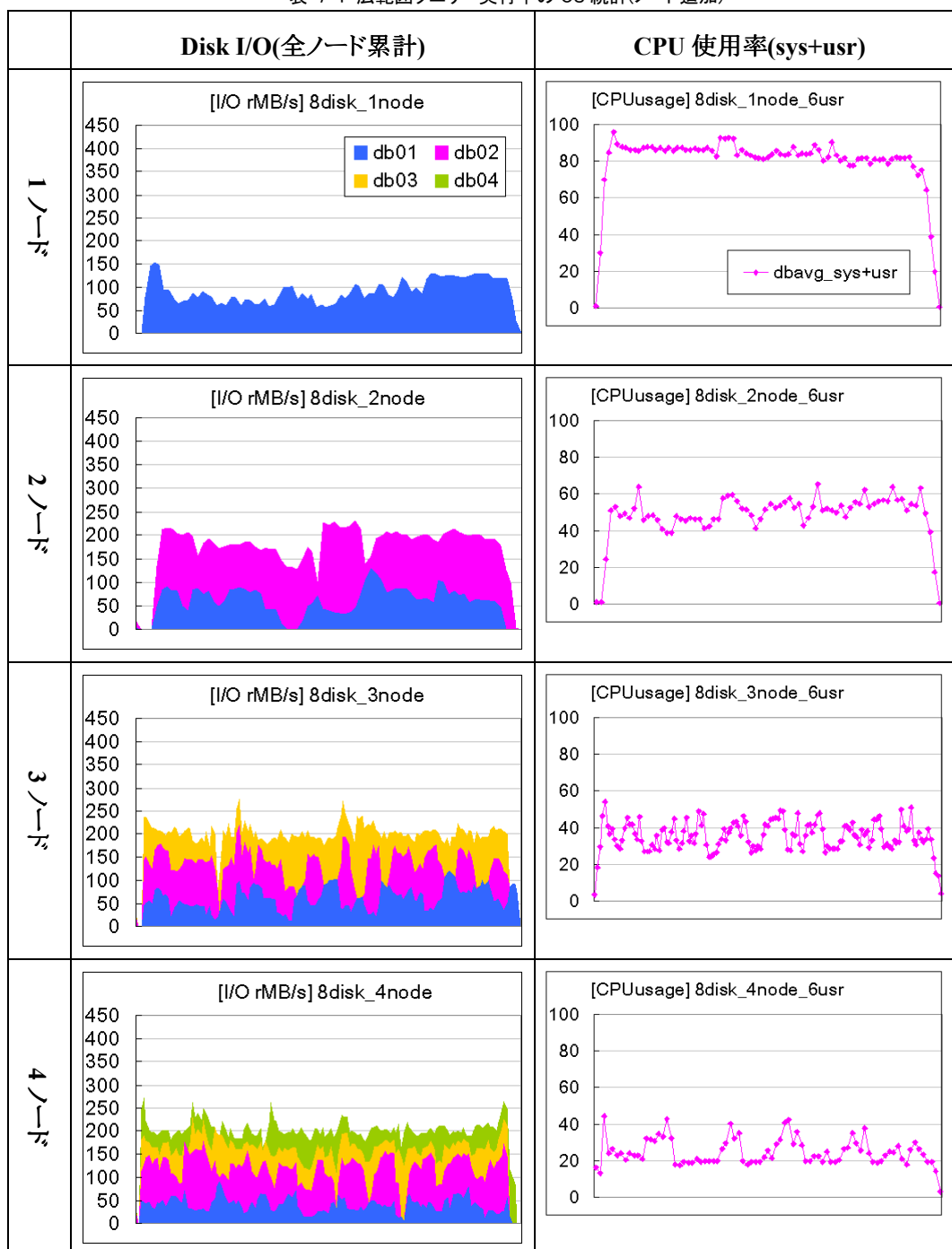
図 7-16 クエリー1 回あたりの平均実行時間相対値

図 7-16 のとおり、DB サーバ 1 ノードから 2 ノードへノードを追加することにより、平均実行時間 (SQL レスポンスタイム) が改善していることが分かります。従って、1 ノードの場合は、CPU リソースのボトルネックが存在したと推測されます。

他方で、2 ノードの Oracle RAC に 1 ノード追加し 3 ノードの Oracle RAC 構成にした場合も、若干の性能向上が見られますが、3 ノードの Oracle RAC に 1 ノード追加して 4 ノードの Oracle RAC 構成にした場合はほとんど変化しませんでした。これより、3 ノードの時点で既に CPU リソースはボトルネック要因でなかったと推測されます。

次に、各ノード数における Disk I/O と CPU 使用率の時系列グラフを示します。Disk I/O は全ノードの累計、CPU 使用率は全ノードの平均の値を採用しています。

表 7-1 広範囲クエリー実行中の OS 統計(ノード追加)



1 ノード時の CPU 使用率グラフに着目すると、使用率の高さからボトルネックは主に CPU リソースであったことが分かります。Disk I/O については 100 M バイト/秒前後を推移しています。

レスポンスタイムにあまり違いのない 2、3、4 ノード時の CPU 利用率グラフに着目すると、使用率は 20%～50% で推移しており、CPU リソース以外がボトルネックとなっていることが推測されます。Disk I/O のグラフに着目すると全ノードの合計 Disk I/O が 225 M バイ

ト/秒で頭打ちとなっていることが読み取れます。

事前に OS コマンドの dd を用いて、BIUSER スキーマを格納した ASM ディスクグループ+BI DATA_DG を構成するディスクに対し、ディスクの読込み性能を測定した結果、秒間 220-250MB が性能限界と推測されました。これより、3 ノード及び 4 ノードの構成では、ボトルネックが CPU リソースでなく Disk I/O にあったと考えられます。

なお、2 ノードの場合も CPU 使用率が 50%前後で推移しているのに対し、Disk I/O は 225 M バイト/秒近い値を断続的に出していることから、CPU リソースよりも Disk I/O の方がボトルネックの要因として大きな割合を占めていたと推測されます。

7.2.4. 検証結果:ディスク追加による性能向上

前項より、BIUSER ユーザの環境ではこれ以上 DB サーバのノード追加による性能向上が望めないことが分かりました。また Disk I/O の性能が飽和していたことも分かりました。

そこで次に、DB サーバ 4 ノードの場合に、BI4R ユーザで広範囲クエリーを実行しレスポンスタイムが改善するのかを確認しました。BI4R ユーザは[5.3.2. ASM ディスクグループとユーザ]で述べたとおり、ASM ディスクグループ+BI DATA_DG の倍のディスク本数で構成される ASM ディスクグループ+BI4R_DG 上にスキーマを持ちます。

結果は図 7-17 のとおりです。

図 7-16 では、3 ノードから 4 ノードへのノード追加を行っても性能向上が見られなくなりましたが、図 7-17 に示すとおり、ディスクを追加することでより性能が向上することが確かめられました。

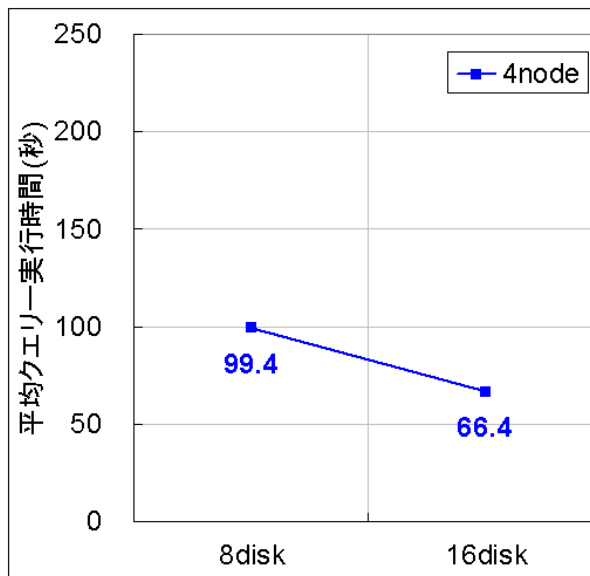
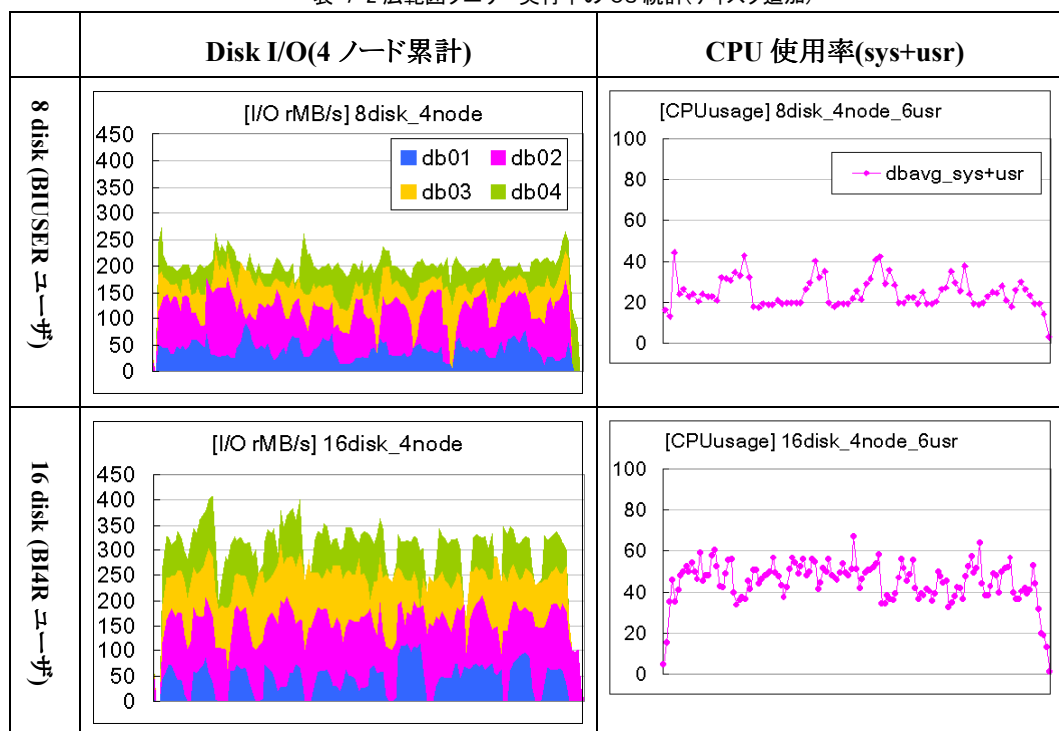


図 7-17 クエリー1 回あたりの平均実行時間相対値

表 7-2 に各ディスク 8 本、16 本の各パターンにおける Disk I/O と CPU 使用率の時系列グラフを示します。Disk I/O は 4 ノードの累計、CPU 使用率は 4 ノードの平均の値を採用しています。

表 7-2 広範囲クエリー実行中の OS 統計(ディスク追加)



ディスクの本数が倍になったことにより、CPU 使用率が全体的に高いことが分かります。これは、ディスクの本数を増やしたことでストレージのデータ供給能力が増強され、その結果 DB サーバの CPU が Disk I/O を待機する時間が減少し、CPU をより効率的に使えるようになったため、と推測されます。

Disk I/O(4 ノード累計)に着目すると、BIUSER ユーザの場合は上限 200-220 M バイト / 秒程度だったのに対し、BI4R ユーザでは 337 M バイト/秒前後まで引き上がっているのが分かります。

7.2.5. Oracle RAC スケーラビリティのまとめ

本節の検証では、広範囲クエリーを用いて Oracle RAC のリソース追加によるスケールビリティを確認しました。図 7-18 で示すとおり、ボトルネックとなったリソースを追加すると、他のリソースがボトルネックとなることがあります。その場合でも、新たなボトルネックに対してリソースを追加すれば、更に性能向上が可能であると確認しました。

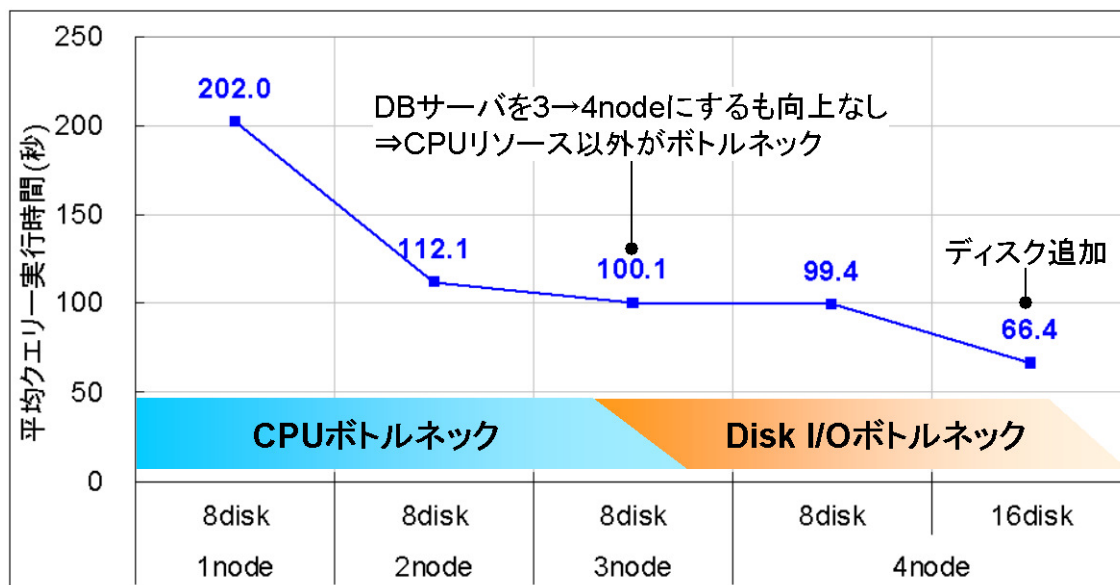


図 7-18 ボトルネックの変移

8. 複数レイヤーでのリソース消費

8.1. 検証目的

前章までの検証では、小範囲クエリーと広範囲クエリーを独立して試行しました。小範囲クエリーは、Oracle BIEE レイヤーのキャッシュの効果により、データベース・レイヤーの負荷が下がるため、広範囲クエリーとはリソースの消費傾向が異なります。

本章では、リソース消費傾向が異なる小範囲クエリーと広範囲クエリーを同時に実行し、相互の性能への影響を確認します。

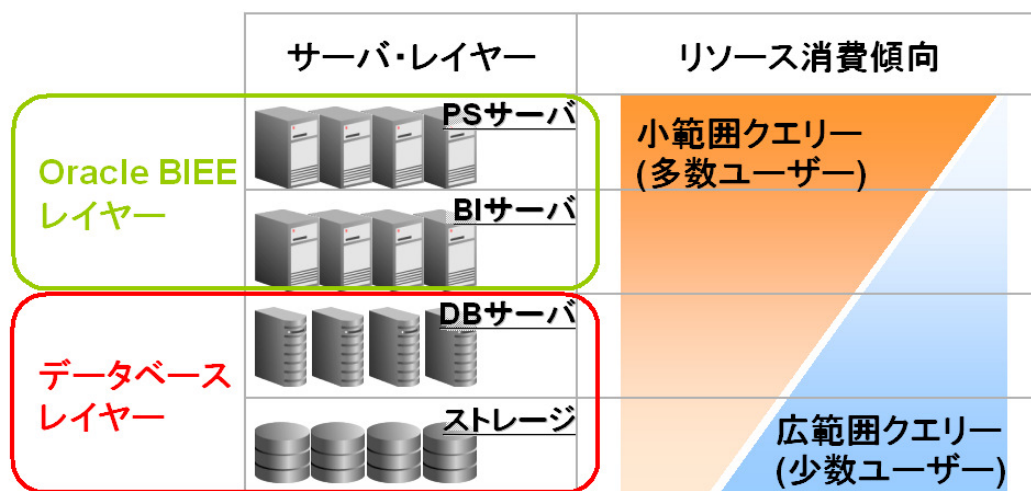


図 8-1 リソースの消費傾向

8.2. 検証方法

前節の検証目的で述べたとおり、リソース消費傾向の異なる小範囲クエリーと広範囲クエリーを同時に実行します。具体的には、小範囲クエリーを Oracle Load Testing で実行している最中に、広範囲クエリーを手動で実行します。広範囲クエリーを手動実行するタイミングは、小範囲クエリーを独立して試行した際の最適 VU 数が 1600 だったため、小範囲クエリーが 1600 に達し TPS が安定したタイミングとします。

8.3. 検証結果

図 8-2 に Load Testing で実行した小範囲クエリーの TPS と、DB サーバの OS 統計を示します。Disk I/O として vmstat の bi(ブロックイン発生数)、CPU 使用率として vmstat の cpu (us) を用いました。

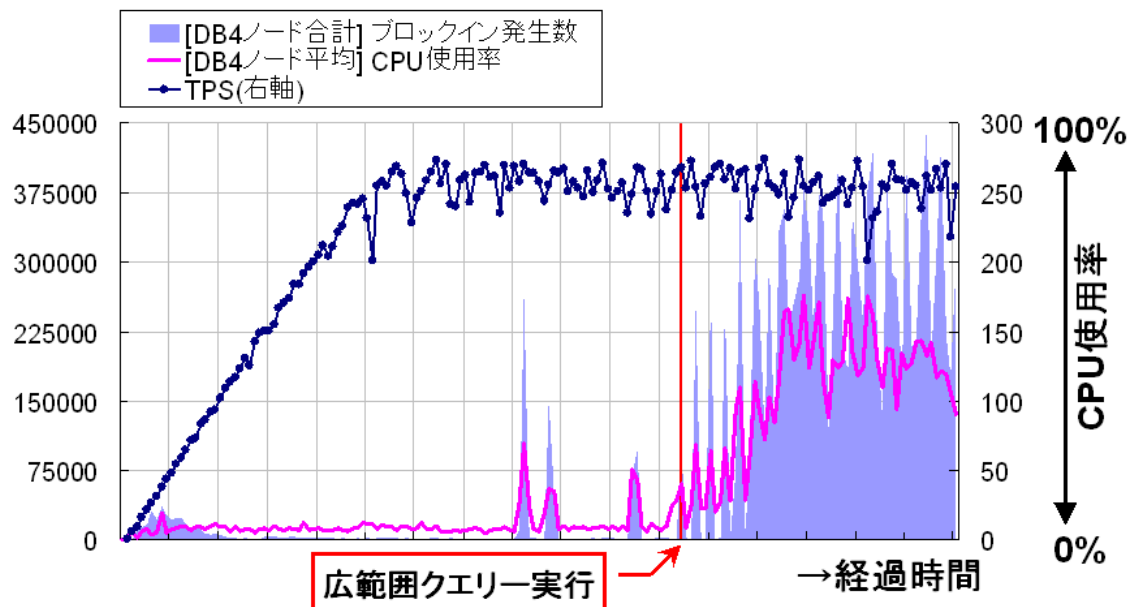


図 8-2 小範囲クエリーTPSとDB サーバの OS 統計

広範囲クエリーを実行したのは図 8-2 内で示した時点です。広範囲クエリー実行に伴い、DB サーバの CPU 使用率および Disk I/O が増加していますが、小範囲クエリーの TPS については目立った劣化がないことが分かります。

本検証で使用した小範囲クエリーは、Oracle BIEE レイヤーでのキャッシュの効果により、データベース・レイヤーへの負荷が下がり、相対的に Oracle BIEE レイヤーの負荷が高くなる傾向があります。これに対し広範囲クエリーはデータベース・レイヤーの負荷が高くなる傾向があります。本検証では、このようにリソース消費傾向の異なる複数の処理を同時に実行しても、性能への影響は小さいことを確認しました。

8.3.1. PS サーバのチューニング・ポイント

検証の際、BI サーバ上で広範囲クエリーを実行中であるにも関わらず、その結果が PS サーバへ返らないうちに、広範囲クエリーが PS サーバから BI サーバへ繰り返し発行される、という問題が発生しました。問題の詳細は下図のとおりです。

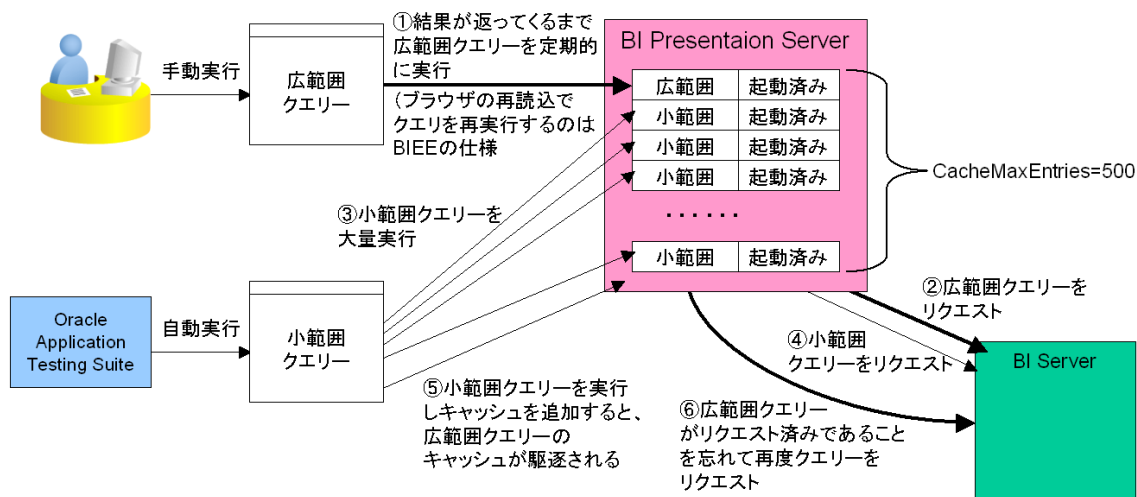


図 8-3 PS サーバの動作

PS サーバ上で保持している各クエリーのキャッシュには、クエリーのステータスが保持されています。一方、CacheMaxEntries というパラメータによって PS サーバにロードされるキャッシュのレコード数が決まります。この CacheMaxEntries の値が小さすぎると、PS サーバ上にロードされるレコードが、次々と発行される小範囲クエリーでいっぱいになり、広範囲クエリーのキャッシュが追い出されてしまいます。キャッシュにはクエリーのステータスが保持されているため、広範囲クエリーのキャッシュが追い出されてしまうと、ステータスがまだ起動中であることが認識されなくなり、その結果広範囲クエリーが繰り返し発行されてしまいます。

本検証においては、CacheMaxEntries の値を 500 から 3000 へ変更することによってこの問題は解決しました。

9. ノード追加手順に関する検証

本章では、Systemwalker Resource Coordinator Virtual server Edition (RCnr VE)のクローニング機能による、サーバブレード設置後の作業の削減効果、およびOracle RACのノード追加の検証結果について述べます。

物理的にサーバブレードを設置した後の作業として、OSのインストール、OSのパッチ適用、ネットワークパラメータの設定などが挙げられます。以降これらの作業を「OS導入」と総称します。

OS導入は、従来は各サーバブレードごとに行う必要がありました。しかしRCnr VEのクローニング機能を用いると、最初の1台にOS導入を行えば、以降は導入済みのOSイメージ(以降、マスタと呼称)を、各サーバブレードへクローニング(配付)することにより、OS導入の作業コストを削減できます。

マスタの採取、および配付はRCnr VE管理サーバで実行します。マスタイメージを保存しておけば、システム稼動前の環境構築時だけでなく、システム稼動後新規にサーバを増設する際もOS導入作業を削減できます。

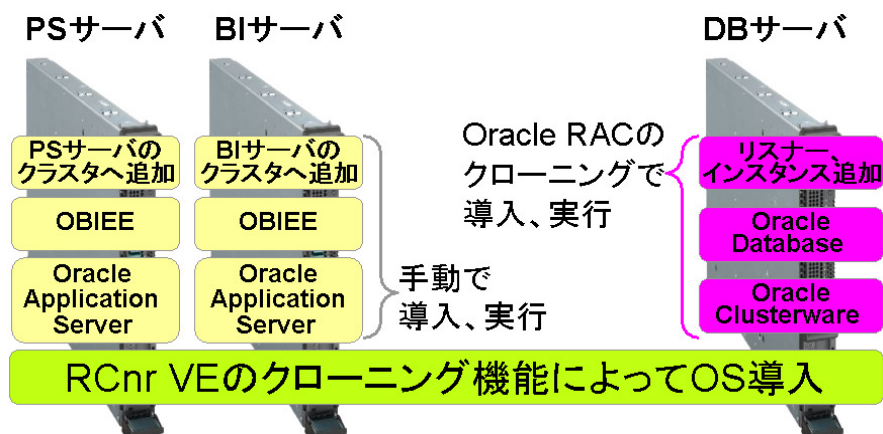


図 9-1 RCnr VE による OS 導入とその後の作業

PS/BI/DB 各サーバ用のサーバブレードへ、RCnr VE のクローニングを用いて OS 導入を行った後の作業は、図 9-1 のようになります。

9.1. RCnr VE によるプラットフォーム追加手順検証

9.1.1. 検証目的

RCnr VE のクローニング機能を用いて OS 導入を行い、その有効性を確認します。

9.1.2. 検証方法

RCnr VE を用いて管理 LAN 経由で、クローニング(マスタ採取・配付)を行います。PS/BI サーバのマスタ採取は各 1 ノードずつ同時に、マスタ配付は各 2 ノードずつ同時に行いました。また、業務 LAN のネットワークパラメータ自動設定機能を使用しました。

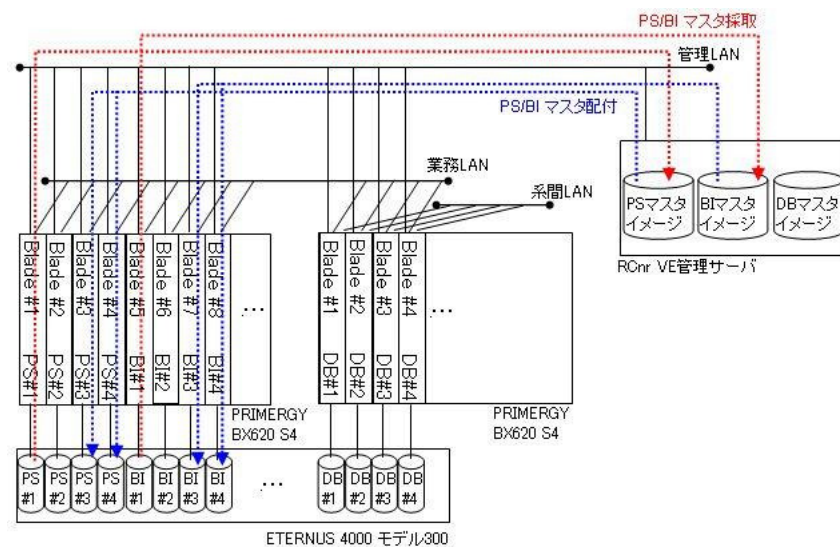


図 9-2 PS/BI のマスタ配付・採取

DB サーバのマスタ採取は 1 ノードで行い、マスタ配付は 3 ノード同時に行いました。

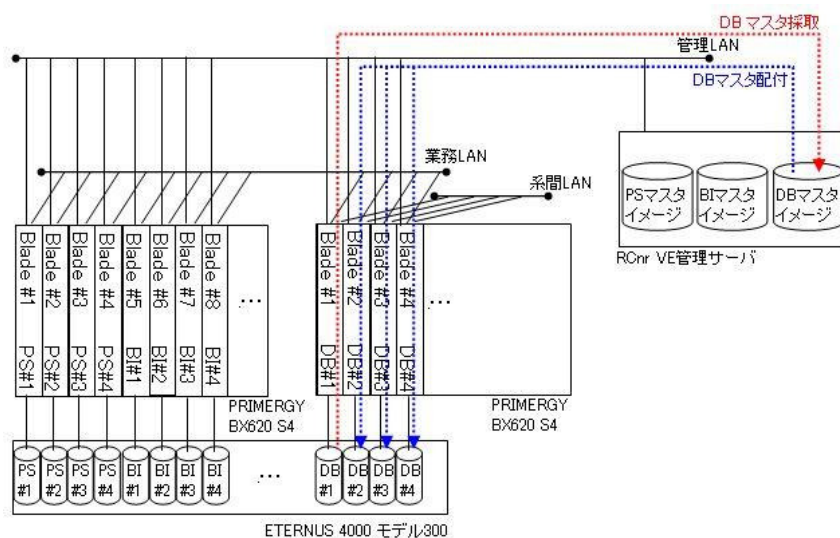


図 9-3 DB サーバのマスタ採取・配付

9.1.3. 検証結果

RCnr VE のクローニングの結果を表 9-1 に記載します。

表 9-1 RCnr VE のクローニング結果

対象サーバ	対象作業	同時実行サーバ数	ネットワークパラメータ設定	所要時間	作業短縮時間※1
DB	マスタ採取	1	無	0:39:37	8:28:20
	マスタ配付	3		1:22:03	
BI	マスタ採取	1	有	0:47:33	5:35:58
	マスタ配付	2		1:12:29	
PS	マスタ採取	1	有	0:39:02	5:44:20
	マスタ配付	2		1:12:38	

※1 作業短縮時間

《作業短縮時間》＝

《RCnr VE を使用しない場合の作業時間》－《RCnr VE を使用した場合の作業時間》

《RCnr VE を使用しない場合の作業時間》＝

《1 台の OS 手動構築時間》×《同時実行サーバ数》

《RCnr VE を使用した場合の作業時間》＝

《マスタ配付・所要時間》＋《マスタ採取・所要時間》

《1 台の OS 手動構築時間》＝

《OS インストール・パッチ適用・設定時間》＋《ネットワーク設定時間》

(但し「ネットワーク設定時間」はネットワークパラメータ設定有りの場合)

《OS インストール・パッチ適用・設定時間》は 210 分、《ネットワーク設定時間》は 18 分とした

DB サーバ用サーバブレードで約 8 時間半、PS/BI サーバ用サーバブレードでそれぞれ約 5 時間半の時間短縮となりました。これより、サーバブレード設置後の OS 導入のコスト削減に、RCnr VE のクローニング機能が大変有効であることが確かめられました。

9.2. Oracle RAC ノード追加手順検証

本検証では、前項で実施した RCnr VE による OS 導入後、サーバブレード追加の基盤となる環境として、図 9-1 に示したとおり PS/BI サーバ用の各サーバブレードごとに、手動で Oracle Application Server および Oracle BIEE のインストールを行いました。他方で、DB サーバは 3 台同時に Oracle RAC ソフトウェアのインストールおよびリスナー、インスタンスの作成を行いました。

この 3 ノード Oracle RAC 環境に、業務を模した負荷下での Oracle RAC ノード追加、および業務がない場合の Oracle RAC ノード追加の両方を実行し比較して、業務への影響を確かめました。

9.2.1. 検証目的

業務中に Oracle RAC のノード追加 (3 ノード Oracle RAC 環境に 1 ノードを追加) を行

う場合において、次の点を検証します。

- a) 業務が Oracle RAC のノード追加に与える影響
- b) Oracle RAC のノード追加が業務に与える影響

9.2.2. 検証内容

次の方法で検証します。

- 1) 業務アプリケーションを実行していない状態で Oracle RAC のノード追加時間を測定
- 2) 業務アプリケーションを実行中の Oracle RAC のノード追加時間を測定

「a」 業務が Oracle RAC のノード追加に与える影響」は 1)と 2)の時間差を見ます。

「b」 ノード追加が業務に与える影響」は 2)を実行しているときの業務 TPS の変化を見ます。

なお、業務アプリケーションとして以下を使用しました。

- 「小範囲クエリー」で使用している SQL とほぼ同等のものを使用
- BI サーバから直接 Oracle RAC に JDBC 接続し実行
- 各ノードで 125 多重ずつ実行
- 各ノードの CPU 使用率が 90%程度になるように実行

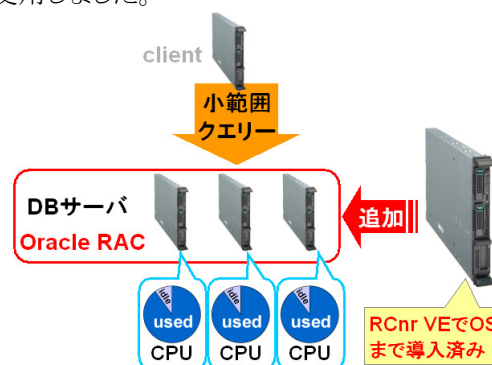


図 9-4 業務アプリケーション

なお、本文中では、3 ノード Oracle RAC 環境の 1 ノード目を「既存ノード」、追加するノードのことを「追加ノード」と記載しています。ノード追加作業の中で 3 ノード Oracle RAC 環境で実施すべき作業は、「既存ノード」で行いました。

9.2.3. 検証結果

a) 業務が Oracle RAC のノード追加に与える影響

Oracle RAC のノード追加手順と各手順に要した時間(業務)を表 9-2 に記載します。

表 9-2 Oracle RAC ノード追加手順と所要時間

手順番号	手順概要	手順詳細	作業 種別	所要時間 (業務無)	所要時間 (業務有)	備考
1	Oracle Clusterware の追加					
1-1	Oracle Clusterware の配付					
1-1-1	addNode.sh の実行と、 追加ノード情報の設定	既存ノードでOUIを以下の方法で実行 \$ cd \$ORA_CRS_HOME/oui/bin \$./addNode.sh 追加ノードの情報を設定	手	0:00:54	0:00:45	
1-1-2	ソフトウェアの配付と インベントリ更新		自	0:01:00	0:01:37	配付元のソフトウェアの サイズは約800MB
1-1-3	oraInstRoot.sh の実行	追加ノード上でrootで実行 # <Inventory>/oraInstRoot.sh	手	0:00:24	0:00:15	
1-1-4	rootaddnode.sh の実行	既存ノード上でrootで実行 # \$ORA_CRS_HOME/install/rootaddnode.sh	手	0:00:21	0:00:42	
1-1-5	root.sh の実行	追加ノードで上でrootで実行 # \$ORA_CRS_HOME/root.sh	手	0:00:45	0:00:39	
1-2	ONS 構成の追加	追加ノード上でoracle ユーザで実行 \$ cd \$ORA_CRS_HOME/bin \$./racgons add_config <追加ノード> <remote_port> <remote_port>:11gのデフォルトは6251 \$ srvctl start nodeapps -n <追加ノード>	手	0:01:07	0:00:45	
1-3	クラスタ整合性の確認	既存ノード上でoracle ユーザで実行 \$ \$ORA_CRS_HOME/bin/clusvfy comp clumgr -n all \$ \$ORA_CRS_HOME/bin/clusvfy comp clu \$ \$ORA_CRS_HOME/bin/clusvfy stage -post crsinst -n all	手	0:00:48	0:00:51	
合計時間				0:05:19	0:05:34	
2	Oracle Database の追加					
2-1	Oracle Database の配付					
2-1-1	addNode.sh の実行と 追加ノード情報の設定	既存ノード上でoracle ユーザで実行 \$ cd \$ORACLE_HOME/oui/bin \$./addNode.sh OUIが起動するので 追加ノードの情報を設定	手	0:00:42	0:00:51	
2-1-2	ソフトウェアの配付と インベントリ更新		自	0:03:12	0:05:15	配付元のソフトウェアの サイズは約4GB
2-1-3	root.sh	追加ノード上でrootで実行 # \$ORACLE_HOME/root.sh	手	0:00:24	0:00:18	
合計時間				0:04:18	0:06:24	
3	リスナーの追加					
3-1	netca の実行	追加ノード上でoracle ユーザで実行 \$ \$ORACLE_HOME/bin/netca 構成するノードを問われるので新規に追加したノードのみを 選択	手	0:00:58	0:00:33	
3-2	listener.ora の修正	VIPホスト名とVIPリスト名の入替	手	0:00:18	0:00:18	
合計時間				0:01:16	0:00:51	
4	インスタンスの追加					
dbca の実行						
4-1-1	dbca の起動と 追加ノード情報の設定	既存ノードで実行 \$ \$ORACLE_HOME/bin/dbca 追加ノードの情報を設定	手	0:01:30	0:02:12	
4-1-2	追加ノードの ・UNDO/REDO追加 ・インスタンス情報設定 ・インスタンス起動		自	0:06:22	0:11:22	追加ノードの ・UNDO 8GB ・REDO 8GB×3Group ⇒合計32GB
合計時間				0:07:52	0:13:34	
総合計時間				約19分 + α	約27分 + α	α (ハードウェア性能 +追加REDO/UNDOサイズ +作業者経験等)

※1:この結果は本環境での値であり実際の作業時間は様々な要因により変化します

※2:「作業種別」の列の意味は次のとおりです

‘手’ - 手動作業。コマンド実行や画面遷移等の作業

‘自’ - 自動作業。インストーラや Configuration Assistant が自動的に実行

つまり、‘手’に要した時間は業務負荷による遅延以外にも作業者の手動作業時間誤差が含まれます。

業務を実行中の Oracle RAC ノード追加時間は約 27 分で、業務を実行していない場合と比較して、約 8 分(約 30%) 遅くなりました。この差は「2. Oracle Database の追加」(約 2 分の差)と「4. インスタンスの追加」(約 6 分の差)が主な原因です。さらに「2. Oracle

Database の追加」ではソフトウェア配付時間が主な遅延差で、「4. インスタンスの追加」では追加ノードの UNDO/REDO の作成時間が主な遅延差(3 分 30 秒ほど)でした。

これらの遅延の原因は、CPU ボトルネックです。業務なし、ありの場合の、既存ノード CPU 使用率を図 9-5 に示します。なお、図中の番号は各手順の番号です。

まず、業務を実行していない場合は、「2. Oracle Database の追加」と「4. インスタンスの追加」の際に CPU を 20%程度使用しています。

一方、業務を実行している場合は、業務に CPU を 90%以上使用しているため、ノード追加作業に割り当てられる CPU 時間は減少してしまいます。

この結果から、業務中にノード追加を行う場合は特に Oracle Database のソフトウェア配付や UNDO/REDO 作成を作業時間の変動要因として考慮する必要があります。

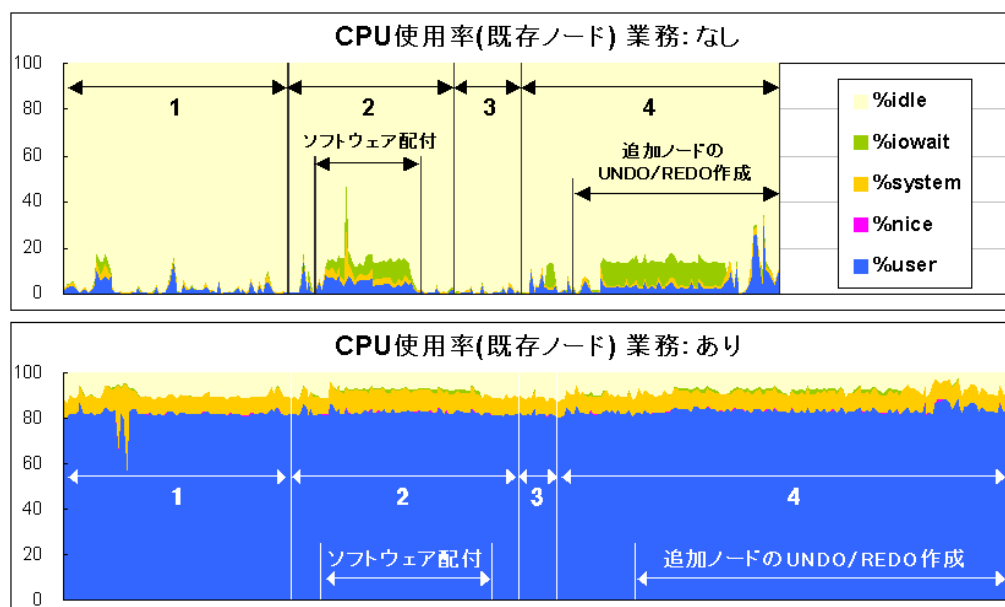


図 9-5 CPU 使用率(ノード1)業務なし・あり

b) Oracle RAC のノード追加が業務に与える影響

業務を実行している場合(業務有)の既存ノード・追加ノードの CPU 使用率を図 9-6 に示します。

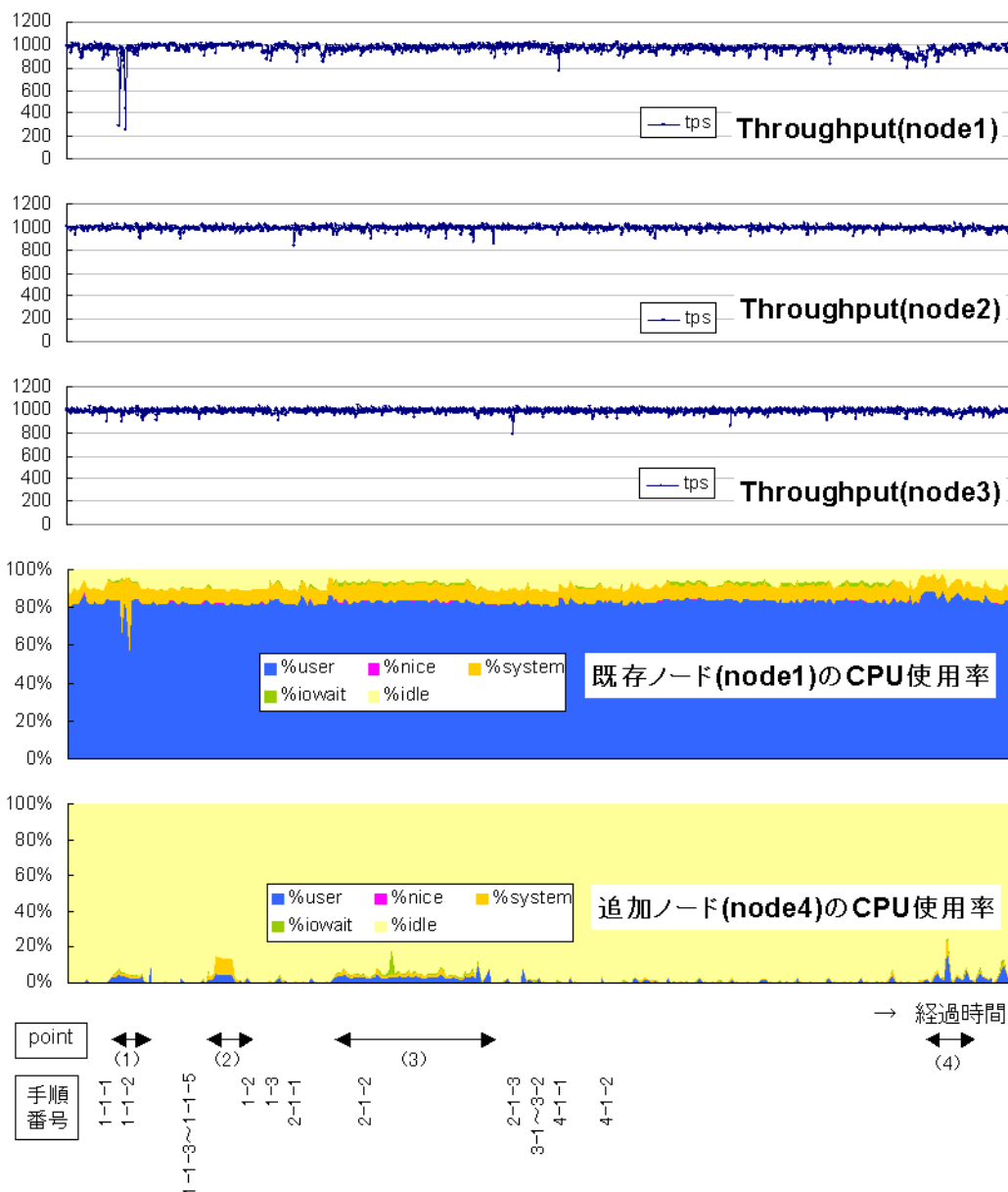


図 9-6 CPU 使用率と TPS(業務有)

【ポイント①】Oracle Clusterware のソフトウェアを配付。既存ノードではページアウトが発生し、数秒の間スループットが 20～30%ほど低下しています。⁶

⁶ ただし、本検証では Oracle Clusterware のソフトウェア配付中に、既存ノードでページアウトが発生してしまい、数秒の間スループットが 20～30%ほど低下しました。原因として、サイズの大きなソフトウェアをコピーしたことにより、OS のファイルシステムキャッシュが増大し、メモリを圧迫した可能性が挙げられます。このメモリの圧迫は、ファイルシステムキャッシュを制御するカーネルパラメータの調整によって回避できる場合がありますが、本検証においては検証期間の都合により追加調査できませんでした。

【ポイント②】この時点で追加ノードの ONS が起動しようとしています、ONS の構成が未設定のため、/var/log/messages にエラーが出力されます。この際に追加ノードで CPU を 10%～20%ほど使用しています。ONS の構成登録後はこの現象は発生しません。ただし、この現象は追加ノードで発生するため業務には影響を与えません。

【ポイント③】Oracle Database のソフトウェアを配付中ですが、業務にはほとんど影響していません。

【ポイント④】追加ノードの Oracle RAC インスタンスが起動。再構成が一瞬なのでほぼ業務に影響を与えていません。

以上より、全体的に Oracle RAC のノード追加は、ほぼ業務に影響を与えないという結果となりました。

10. 総合評価

本検証ではDBサーバやストレージも含めたBIシステム全体を意識し、各レイヤーにおいてスケールアウトによる処理性能のスケラビリティ検証を行いました。その結果、PSサーバ、BIサーバ、DBサーバ、ストレージの全レイヤーにおいてスケールアウトによる性能向上が可能であると確かめられました。

BI 使用ユーザ数として高い割合を占めると想定される、営業社員をはじめとする現場層をシミュレートした小範囲クエリーでは、PSサーバおよびBIサーバがボトルネックとなりがちでした。これは Oracle BIEE レイヤーのキャッシュの効果により、データベース・レイヤーへリクエストを発行しその結果を待機する時間が削減され、Oracle BIEE レイヤーの CPU を効率的に使用できたため、と推測されます。本検証によって、この Oracle BIEE レイヤーを構成する PS サーバ、BI サーバともにスケールアウトによる性能向上が可能であることが確かめられました。

他方、BI 使用ユーザ数として占める割合は少ないと想定される、管理層をシミュレートした広範囲クエリーでは、データベース・レイヤーでボトルネックが発生しました。クエリーの頻度が低いために Oracle BIEE レイヤーの負荷は低いものの、大量のデータの読み込み及び計算処理を行うデータベース・レイヤーがボトルネックとなります。しかしこちらも本検証により、ノード追加およびストレージのディスク追加によって性能を向上させられると分かりました。

サーバブレードの追加を行うことで性能向上可能であることは確かめられましたが、一般的にノードの追加は、新規ノード設置スペースの確保、電源の確保、OS のインストールおよびパッチの適用、ソフトウェアのインストール等様々な物的、人的コストが伴います。それらを解決するために富士通 PRIMERGY ブレードサーバは、ブレードサーバならではの省スペース性に加え、Systemwalker Resource Coordinator VE によるクローニング技術を擁し、ノード追加に伴うコストを軽減します。本検証では PS サーバ、BI サーバ、DB サーバの追加をこれらの機能を活用して行い、大幅な作業時間削減を成し遂げました。

今回の検証では 4Core CPU/5GB メモリの PS サーバ 4 台、4Core CPU/8GB メモリの BI サーバ 2 台、キャッシュ・ヒット率 60%、レスポンスタイム 3 秒以内、という条件で秒間 260 トランザクションを実現しました。実際の事例でも同等のパフォーマンスを実現しており、妥当な結果と言えます。

以上の結果により、Oracle Business Intelligence Suite Enterprise Edition は、適切なリソースの追加によってビジネス環境の変化に流されることなく常に優れた性能を提供できること、柔軟なリソース追加が可能なプラットフォームとして富士通 PRIMERGY ブレードサーバが大変適していることが確かめられました。



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

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

Copyright© 2009, Oracle. All rights reserved.

Copyright© 2009, FUJITSU LIMITED, 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。

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。

本書は、Oracle GRID Center 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、予告無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。

Intel、Xeon は、米国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

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 Torvals の商標です。

その他の各種製品名は、各社の製品名称、商標または登録商標です。」

本資料に記載されているシステム名、製品名等には、必ずしも商品表示((R)、TM)を付記していません。



Copyright© 2008, Oracle. All rights reserved.
Copyright© 2008, FUJITSU LIMITED, All rights reserved