

ORACLE®

*JaSST'13 Tokyo セッション D3-1*

# コンサルタントが語る DBシステムのテストの現場

日本オラクル株式会社  
テクノロジーソリューションコンサルティング統括本部  
マネージングプリンシパルコンサルタント  
大塚 信男

2013/01/30



# Agenda

1. 講師紹介
2. オラクルコンサルティングサービス紹介
3. 企業システムにおけるDBの位置付け
4. DBレイヤにおけるテストの目的
5. 性能テストの落とし穴
6. よくあるDBテストの失敗例
7. 性能テストをさらに有用なものにするには
8. 適切な性能テストのための課題
9. DBシステム安定稼働に向けて
10. オラクル社テスト製品のご紹介

# 講師紹介

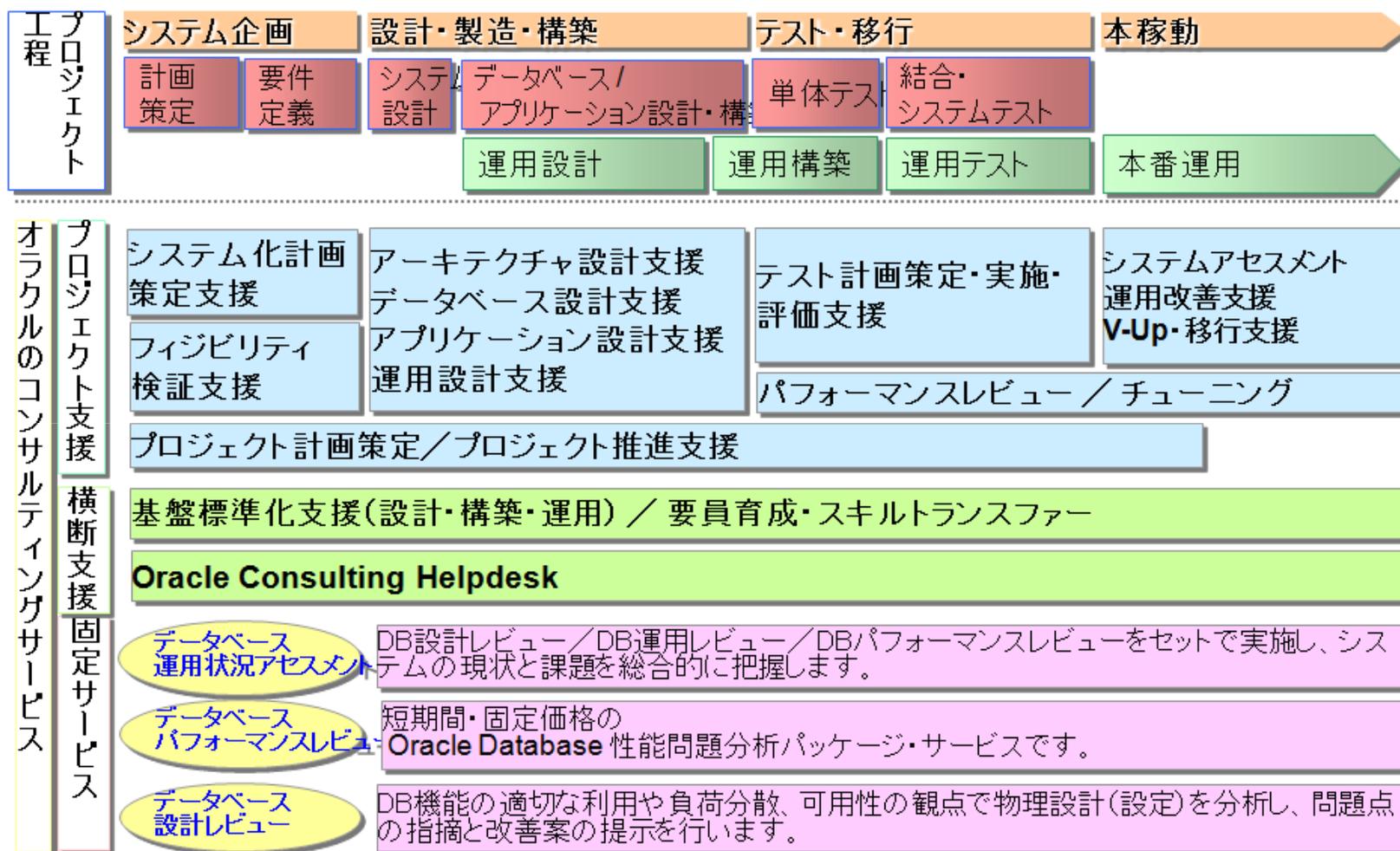
## ■ 大塚信男

- 96年日本オラクル新卒入社。Oracle Applicationsマーケティング技術担当として配属。Sunプラットフォームでのサイジングテスト、ベンチマークテストなどを担当
- 99年サポートサービス本部へ異動。社内向けデータベース内部動作解説トレーニングコースの開発および講師を担当。
- 02年コンサルティングサービス本部へ異動。DBコンサルタントとしてサービスデリバリを担当。
  - 主なプロジェクト：
    - 某自治体税務システム
    - 某証券取引所情報系システム
  - その他、性能改善プロジェクト、運用状況アセスメント等多数
- 08年よりDBコンサルティングチームのマネージャを務める
- 著書（共著）：
  - 「門外不出のOracle現場ワザ」（翔泳社）
  - 「新・門外不出のOracle現場ワザ～エキスパートが明かす運用・管理の極意」（翔泳社）

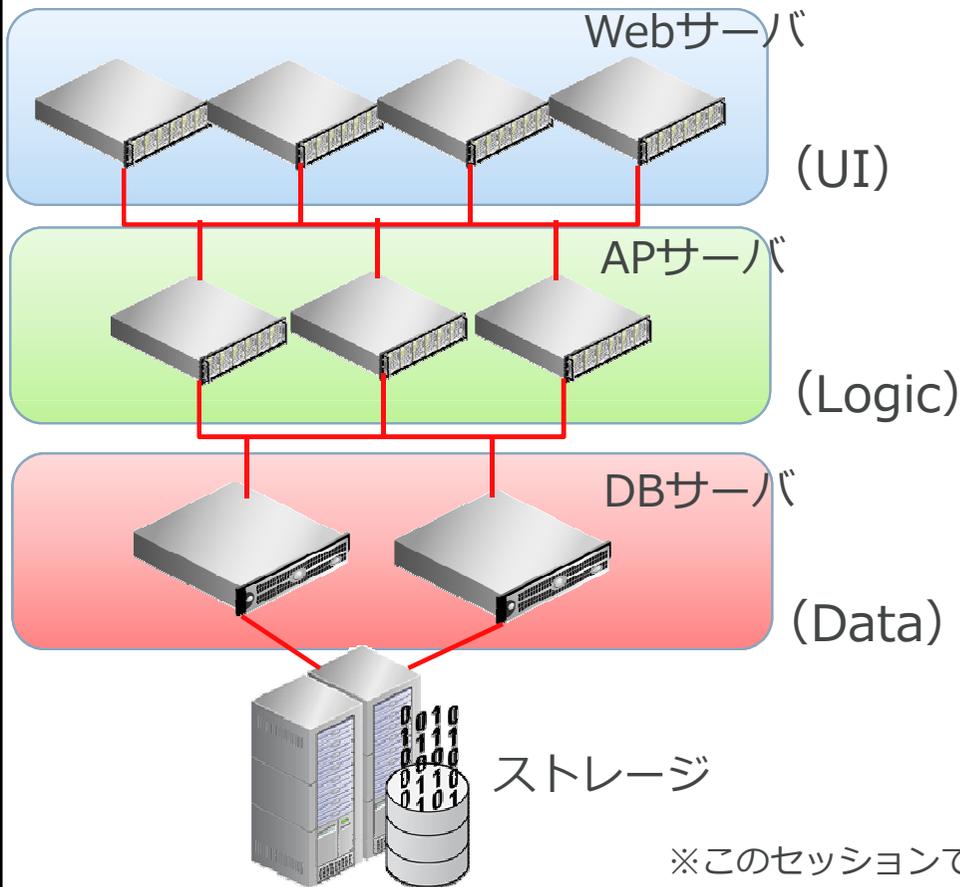


ORACLE

# オラクルコンサルティングサービス紹介



# 企業システムにおけるDBの位置付け



## ■ DB(RDBMS)

- データの永続化
- 同時処理要求に対する矛盾の無いデータの保存と取出
- 大規模データへの高速アクセス
- 障害発生時には、DBを共有する複数業務が停止
- 設計/サイジングによっては、システム全体における重大な性能ボトルネックとなりうる
  - 低速なHDD (RAMと比較して) への書込み/読込みによる性能ペナルティ
  - メモリ上の排他制御機構による競合発生
  - 不適切な索引設計など、SQLチューニング不足によるSQL処理遅延

※このセッションでは、このようなDBを使用するシステムのことを『DBシステム』と呼んでいます。

# DBレイヤにおけるテストの目的

テスト種類	目的・内容
方式検討テスト	理論値ではなく実測値による比較や実現可能性の実証が必要な場合。
基盤テスト	インストールとセットアップの妥当性評価。業務開発側へ環境をリリースする前の最低限のテスト。起動・停止、接続確認テストなど。
障害テスト	可用性が重視されるDBに特徴的なテスト。各種コンポーネントに疑似障害を発生させ、自動復旧やアプリケーションへの影響を確認。フェイルオーバー構成の場合は、系切替時の停止時間や処理性能の縮退率などを確認。リストア・リカバリの所要時間測定など。
性能テスト	応答性能、スループット性能が要件を満たしていることを確認する。単体応答性能、負荷性能、限界負荷性能、長期走行テストなど。
運用テスト	バックアップ手順とリストア・リカバリ手順確認。各種定期メンテナンス（ログ・ローテーション、断片化解消処理など）手順確認。各種監視データの正常取得状況確認。
移行テスト	データ移行手順および移行処理性能の確認。

特にノウハウが必要とされる

ORACLE

# 性能テストの不備がもたらすトラブル例

- DB性能テストを実施していたとしても、システムのサービスイン直後、またはしばらく経ってから、下記のようなトラブルに見舞われることがあります
  - ▶ サービスイン直後、多数のユーザーが同時に使用し始めると、全ての画面応答が**著しく遅く**なり、業務にならない
  - ▶ リソース不足エラーが度々発生し、ユーザーにエラーが返されたり画面がフリーズするため、業務が**不安定**になる
  - ▶ サーバがダウンしたり、ハング状態になり、**全ての業務が停止**する
  - ▶ 負荷ピーク日に、バッチ処理がオンライン時間帯に突き抜け、オンライン開放時間（業務開始時間）が遅れる
  - ▶ ピーク日でもないのに、ある日突然、特定のSQLが性能劣化し、特定の画面のレスポンスが遅延する。場合によっては、これが原因でサーバのリソース枯渇を引き起こし、システム全体でスローダウンが起こる

問題を事前に把握するためには、適切な性能テストが不可欠

# 各種性能テストによる確認事項

- 単体テスト
  - 単体応答性能要件への適合性
- 負荷テスト
  - 同時実行性要件、スループット要件、サーバのキャパシティ状況の要件への適合性の確認
- 限界負荷テスト
  - ユーザ数やデータ量を増幅し、どこまで耐えられるか
  - システムの最大キャパシティと、限界状態で何が起こるかを事前に把握
- 長期走行テスト
  - 負荷テストを長時間（運用上定義されたサーバの定期停止間隔など）実行し、稼働状況の安定性（性能の安定性、リソース使用量の安定性）を確認

# 性能テストの落とし穴

- テスト方法の不備
- テストをしないことのリスク
- テストの限界
  - サービスイン前のテストのみによる性能保証の限界
  - サービスイン後の性能管理

# よくあるDB性能テストの失敗例

## テスト方法の不備

### ■ 負荷試験の失敗①

– データアクセスが局所化されたテストシナリオ

- たとえば、「ユーザーAが、自分の発注明細100件を参照する」という処理を単純増幅（秒間100回など）して投入する試験を行っている

テスト時

アクセスするデータが全てキャッシュヒットするため、高性能

アクセスする共有メモリ上のページが少量

本番稼動後

アクセスされるデータが広範囲にわたり、ディスクへのアクセスが増大。I/Oボトルネックが発生してスローダウン

アクセスされるデータが広範囲にわたり、共有メモリ上の多くのページを参照。それにともない、プロセス毎のPTE(Page Table Entry)メモリが肥大化して、OSメモリ枯渇が発生。クラスタウェアによって異常と判断され、最終的にサーバ停止。

ORACLE

# よくあるDB性能テストの失敗例

## テスト方法の不備

- 負荷試験の失敗②

- 精度の低い疑似データ

- たとえば、負荷試験用のテスト・データを、少量の疑似データを単純増幅することで準備している

テスト時

増幅された疑似データの値の分布に最適化されたSQL実行計画で処理される

本番稼働後

本番データの値の種類や分布がテストデータとは異なっているため、異なるSQL実行計画が選択される。一部のSQLが本番データに対して最適な実行計画が選ばれず、大幅な性能劣化が発生。

※データの量や分布といった条件により、OracleのSQL実行計画が変化することに対する認識が低いケース。テストデータでは、表の列Aの索引を使用したアクセスが高速でも、本番データでは、列Bを使うべきで、その列には索引が作られていなかったため、表のフルスキャンが起きる、といった問題が起きる。

# よくあるDB性能テストの失敗例

## テスト方法の不備

### ■ 負荷試験の失敗③

- スループット目標達成のためのテストを、シリアル実行でテストしていた
  - たとえば、「秒間200件の注文入力」という要件を、シリアルに投入（単一スレッドによる逐次実行）

テスト時

1件あたり5ms未満の応答時間で処理され、要件を達成

本番稼働後

複数スレッドが同時に注文入力処理を実行したとき、表の行値を使用した採番の実装が原因で行ロック待ちが発生。また、単調増加する値のINSERTが原因で、Right-growing Indexの状態となり、ブロック競合が激化。平均応答時間が大幅遅延。

※同時実行性による影響を評価する、という本来の負荷試験の目的を理解せずにテストが実行されているケース

ORACLE

# よくあるDB性能テストの失敗例

## テスト方法の不備

- 長期走行試験の失敗

- 運用上定義された連続運転時間と同等の長期走行試験を行っていなかった
  - たとえば、「週次再起動」の運用要件であったが、48時間程度の長期走行試験を実施

テスト時

試験期間を通し、性能要件は満たしていた。  
リソース消費量の推移には注目していなかった。

本番稼働後

Oracleが使用する共有メモリの領域の一部が、徐々に肥大化し、72時間程度経過したところで上限値に達し、メモリ割当てエラーが発生。さらにメモリ再利用のための内部処理のオーバーヘッドが増加し、内部ロック競合による大幅な応答遅延が発生。

※運用要件と全く同等の期間のテストではなくても、長期走行試験時のリソース消費量の推移を分析することで、潜在的なリスクを検出することができます。

ORACLE

# よくあるDB性能テストの失敗例

## テストをしないことのリスク：あらゆる変更はリスクである

- 一見、影響がないと思われる変更が引き起こした障害例：
  - CPUネックのシステムにCPUを追加したら余計にCPU使用率が高騰し、バッチ処理が大幅遅延
    - 複数のCPUキャッシュ間のデータ整合性維持のオーバーヘッド。CPU数が多い方が、一度あたりのオーバーヘッドが大きくなる
    - 同じ表にアクセスする同時セッションが多数ある場合、Oracleが頻繁にアクセスするメモリ領域があることが原因（現在は修正済みの製品不具合）
  - 表データの入替え処理の高速化のため、データロード処理をパラレル化したら、表アクセスが大幅遅延
    - 設計上、パラレルロードを使用する予定だったが、実装時に有効化するためのコマンドが不足し、シリアルロードされていた
    - コマンドを追加し、パラレルロードを有効にしたところ、当該表にアクセスする問合せの一部が大幅遅延
    - 表のHWMまでデータをスキャンするFull Table Scan処理が遅延した
    - データのロード前にDELETEでデータを消す運用をしていたが、DELETEの場合、表データを空にしてもHWMを引き下げない。シリアルロードはHWMより下にデータをロードするが、パラレルロードはHWMより上にデータをロードするため、データの入れ替えの度に表が肥大化していった

# サービスイン前のテストのみによる性能保証の限界

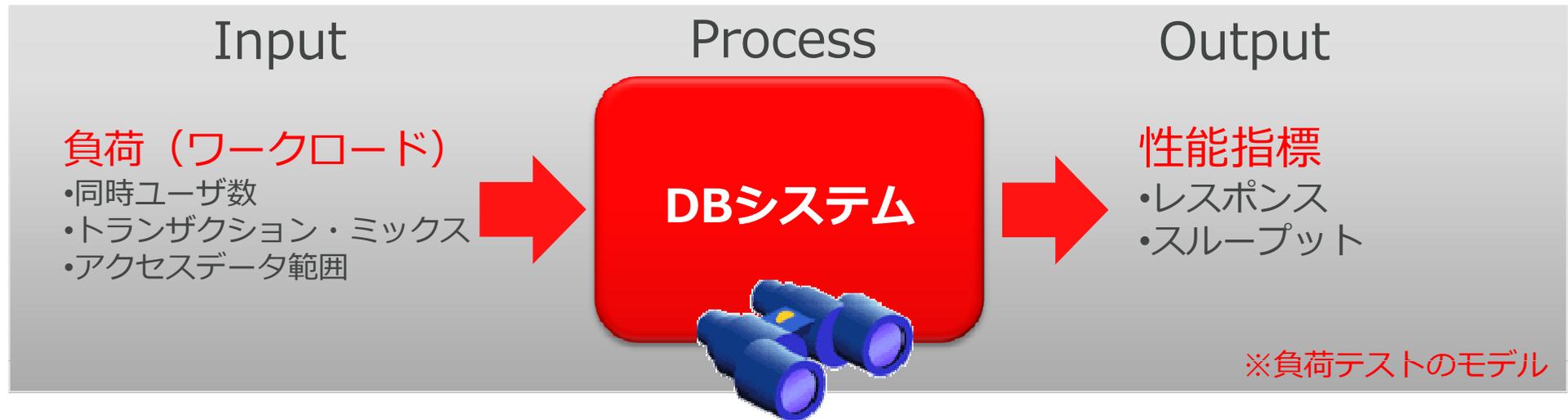
- テストで「想定外の問題が無いこと」を保証はできない（テストの一般原則）
  - 想定したシナリオの範囲内で問題がないことを確認できるのみ
- 一般的に企業システムは、サービスイン後に**変化し続ける**
  - データ量、データ特性、ユーザー数、ユーザーの利用特性（よく利用する機能、1日当たりのトランザクション数など）
  - サービス改善や追加のためのAP修正、新規APリリース
  - ワークロードが変化すれば、それまでの性能テストによる確認は保証にならない
- 変化する状態に適応する機能
  - 自動最適化機能（SQLオプティマイザ、自動メモリ管理、など）が『常に』変化に対して最適なアウトプットを出せるかどうかを、テストで確認することはできない（全数テストは不可能）

**テストで確認すべきことと、その後の運用管理の中で  
マネジメントしていくことを、分けて考える必要がある**

# サービスイン後の性能管理

- 継続的なモニタリングと分析（キャパシティ管理）
  - 予兆を捉え、対策を事前に実施
  - Oracle Databaseは、予兆を捉えるための稼働統計情報を豊富に提供している
  - どの統計をどのような閾値で監視するか、また、複数の統計から、DB内で起きていることを推測し、適切な対応を取るにはノウハウが必要
    - Oracle Enterprise Manager自動診断機能によるレポートニング
    - コンサルティングサービス
- 定期メンテナンスの実施
  - 断片化解消（時間の経過にともなう性能劣化の解消）
  - オプティマイザ統計収集（時間の経過にともなう性能劣化の解消、変化への対応）
  - 計画的なパッチ適用（今起きていない問題が、今後もし起きないとは限らない）

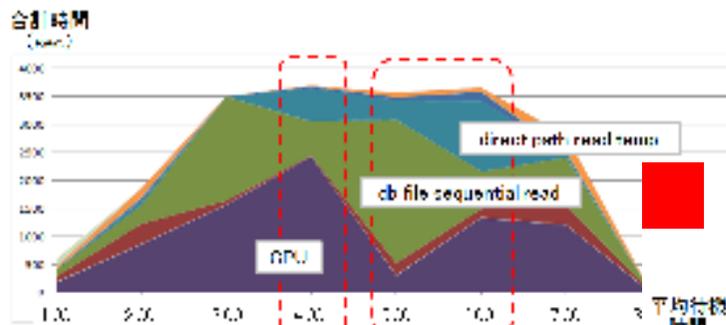
# 性能テストをさらに有用なものにするには



- テストのアウトプットである性能指標の目標達成状況を確認するだけでは、**テスト自体の誤り**や、本番環境で起こり得る**問題の予兆**を見落とす可能性がある
- プロセスをブラックボックスとせず、その中で起こっていること (**テスト時のDB稼働状況**) を適切に分析することが重要

# テスト時のDB稼働状況分析

## DB処理時間分析

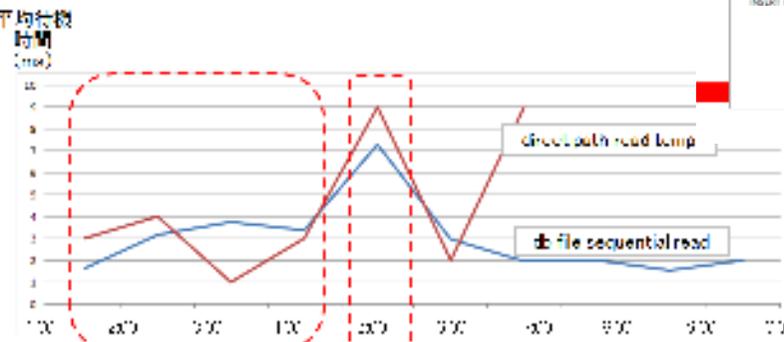


この時間帯は特定のSQL実行の集中による、CPU使用率が高くなる。もしくは並列実行するSQLが複数ある。

この時間帯は特定のSQL実行の集中による、CPU使用率が高くなる。もしくは並列実行するSQLが複数ある。

•リソースを大量消費しているのはどの処理か？原因は？  
•応答性能が問題になりそうな処理はどれか？

## I/Oキャパシティ状況 (平均応答ms)

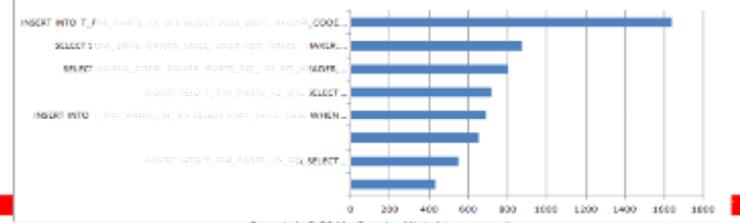
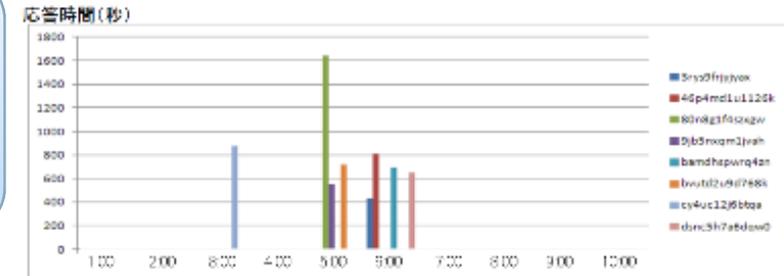


この時間帯はCPU使用率が低下しており、十分な余裕あり。

この時間帯はCPU使用率が低下しており、十分な余裕あり。

•処理時間の制約 (ボトルネック) となっている要因は何か？  
•待機要因と競合の構造は？  
•リソースキャパシティには余裕があるのか？

## 1実行が長時間かかるSQL



•もう少し負荷が増えたらどんなことが起こりそう？  
•継続して監視が必要な項目と監視閾値は？  
•稼働状況が健全すぎるのは、テスト方法のミスではないか？

# 適切な性能テストのための課題

- システムの性能に対して楽観的すぎる姿勢
  - 「だろーう運転」：現実を自分に都合よく解釈してリスクに目を向けない
  - プロジェクトの短納期／低予算の要望が背景
  - 動くシステム（機能要件を満たす）を作ることので精一杯。性能は運任せ、もしくはユーザーに内緒で先送り
- インフラ・アーキテクチャ（SW/HWの構成や内部構造）に対する知識不足、テストの詳細設計のスキル不足、テスト結果の妥当性を評価するスキルの不足、DBの稼働状況の健全性を評価するスキルの不足

# 性能とテストの原則

- 性能は設計してシステムに作りこむもの
- 必要な性能のシステムが都合よく自然にできあがったりはしない
  - HWの進歩や低価格化、およびそれをうまく利用するSW技術の進歩（Oracle Exadataなど）により、今までボトルネックとなっていたものが、大幅に緩和されてきているのは事実
  - しかしながら、これらのアドバンテージを利用して、複数システムを統合したり、Big Dataを活用するサービスの開発など、さらに大規模で複雑なシステムが作られるようになり、新しいボトルネックが生まれている

**性能は設計してシステムに実装し、テストによって確認する、  
という原則は変わらない**

# DBシステム安定稼働に向けて

## ■ インテグレーションベンダー

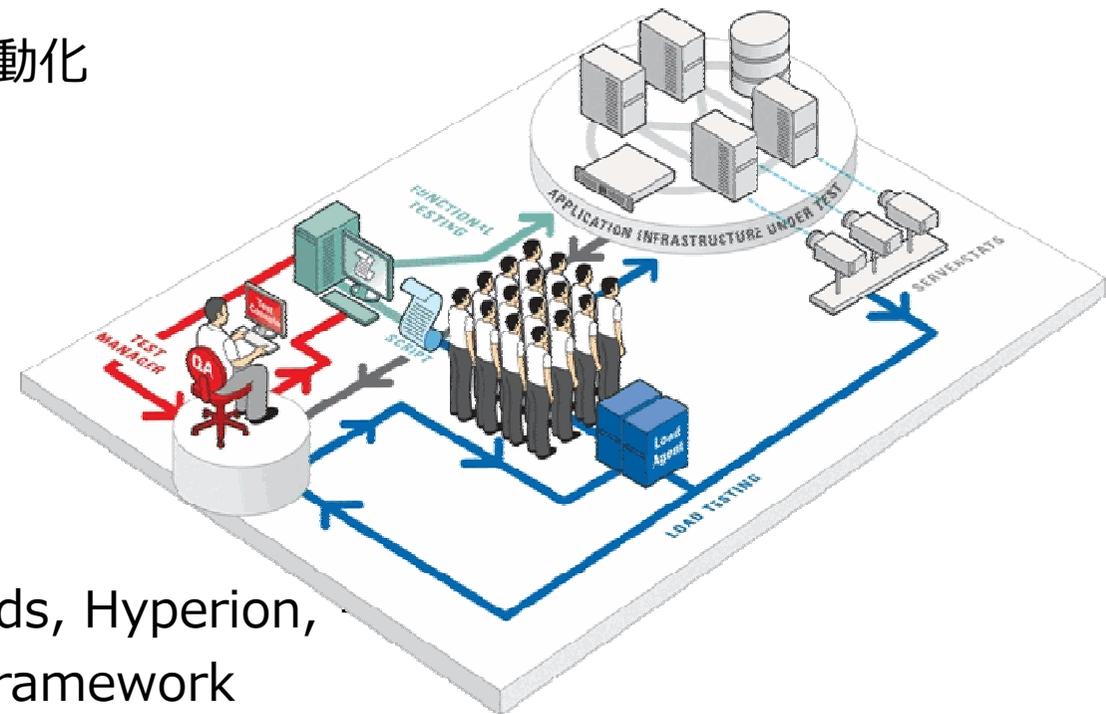
- 必要なテストの内容と重要性をユーザーに訴求し、**十分なテスト計画**をプロジェクトに盛り込むべき
- テストした範囲と残ったリスクをユーザーに正しく説明し、合意を得るべき
- テスト結果をアウトプットのみで評価するのではなく、内部の**稼働状況を評価**することで、テストの品質を向上し、テストからより多くの知見を引き出すべき

## ■ ユーザー（発注者）

- 性能に関するリスクを認識し、性能品質を得るには**投資が必要**であることを認識すべき
- ベンダーのテスト計画を精査し、サービスイン直前に、負荷テスト期間を1週間だけ置くようなプロジェクト計画を提出してきたベンダーは、信用できないと思うべき。性能管理がまるで分かっていないか、予算縮小のための辻褃合わせであり、サービスイン間際、もしくはサービスイン後に地獄を見るのは明らか

# Oracle Application Testing Suite

- ユーザー視点のテストを簡単かつ迅速に実現
  - Oracle Functional Testing
    - 機能/回帰テストやデータ投入を自動化
  - Oracle Load Testing
    - 負荷テストによる性能検証
  - Oracle Test Manager
    - テスト工程の管理
- Oracle製品群に対応
  - Oracle E-Business Suite
  - Siebel CRM, PeopleSoft, JD Edwards, Hyperion,
  - Oracle Application Development Framework



## EASY & SPEEDY

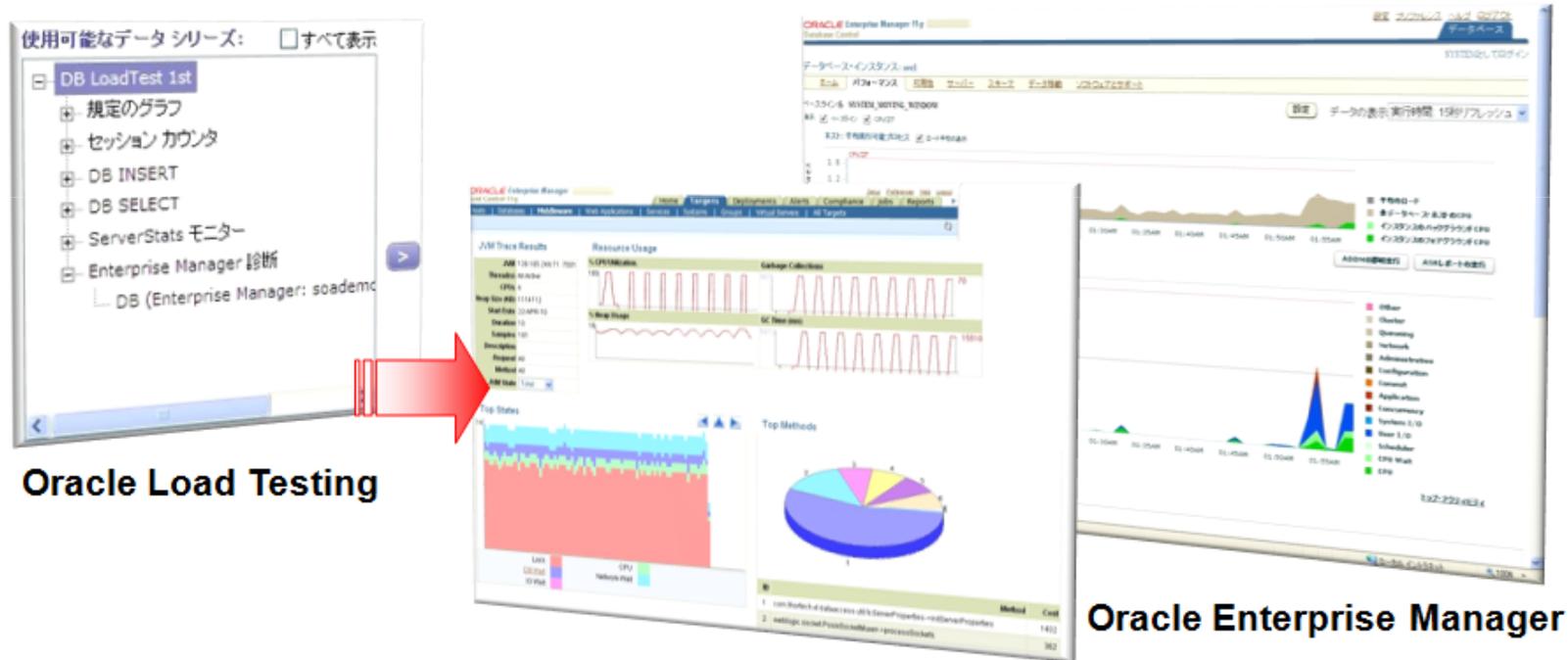
ORACLE



# Oracle Load Testing

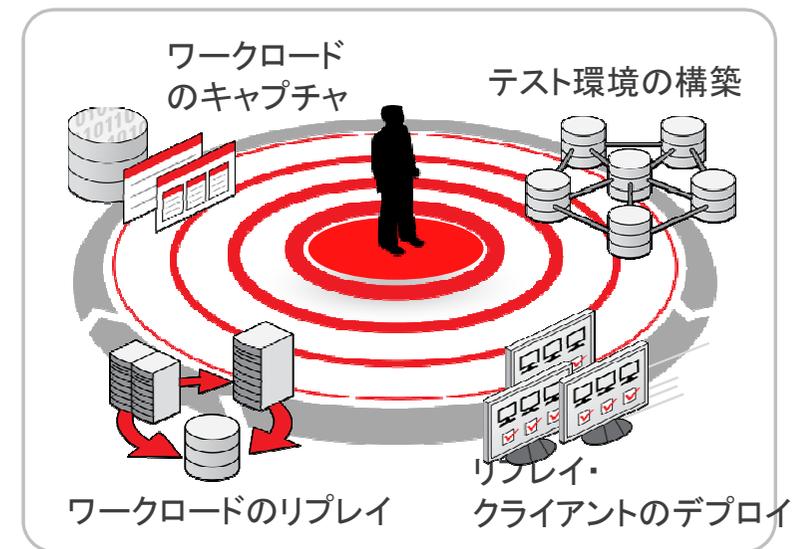
## Oracle Enterprise Manager との連携

- ボトルネック特定のためのパフォーマンス診断
  - 負荷テスト中にデータベースやJavaアプリケーションの詳細なパフォーマンス分析を行うOracle Enterprise Manager へのアクセスを提供



# Oracle Real Application Testing

- Oracle Database 11g からの新機能
- Enterprise Edition (EE) のオプション機能
- テスト品質の向上や工数削減、運用管理コストの削減を実現
- 以下の機能より構成
  - ✓ SQL Performance Analyzer (SPA)
  - ✓ Database Replay (DB Replay)
- より高度な分析には、オプションとして以下の Pack を併用
  - ✓ Diagnostic Pack for DB Replay
  - ✓ Tuning Pack for SPA



ORACLE

**Hardware and Software**

**ORACLE®**

**Engineered to Work Together**

**ORACLE®**