

# Oracle Direct Seminar



# ORACLE®

## Web負荷テスト入門

～準備から実施、分析まで、これだけは押さえておこう～

日本オラクル株式会社

**Oracle** Direct

**Oracle**  
**Direct**  
**Seminar**



# Agenda

- Webアプリケーション負荷テストの基礎知識
- テストデータの作成
- 効果的な負荷テストの方法
- ボトルネックの切り分けと分析
- まとめ
- Appendix

## 無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
  - MySQLからの移行相談
  - PostgreSQLからの移行相談
  - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
- Oracle Developer/2000 Webアップグレード相談
  - パフォーマンス・クリニック
  - Oracle Database 構成相談
- Oracle Database 高可用性診断
  - システム連携アセスメント
  - システムセキュリティ診断
  - 簡易業務診断

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

# Agenda

- Webアプリケーション負荷テストの基礎知識
- テストデータの作成
- 効果的な負荷テストの方法
- ボトルネックの切り分けと分析
- まとめ
- Appendix

## 無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
  - MySQLからの移行相談
  - PostgreSQLからの移行相談
  - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
- Oracle Developer/2000 Webアップグレード相談
  - パフォーマンス・クリニック
  - Oracle Database 構成相談
- Oracle Database 高可用性診断
  - システム連携アセスメント
  - システムセキュリティ診断
  - 簡易業務診断

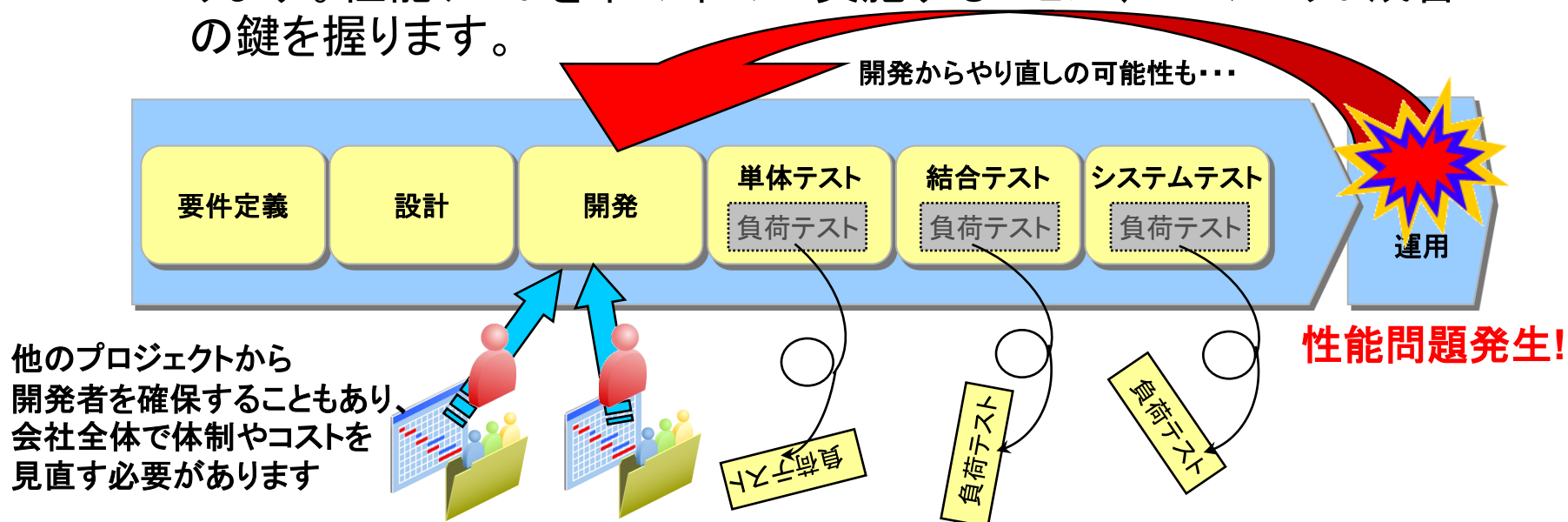
<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

# カットオーバー!と思ったら性能問題発生

- 負荷テストは早めに実施

- 性能問題が存在しないWebシステムはほとんどありません。運用開始後に予期せぬ障害を招かないよう、負荷テストは必須です。
- プロジェクトの進捗に比例して性能問題を改善するコストも多くなります。性能テストを早め早めの実施することが、プロジェクト成否の鍵を握ります。



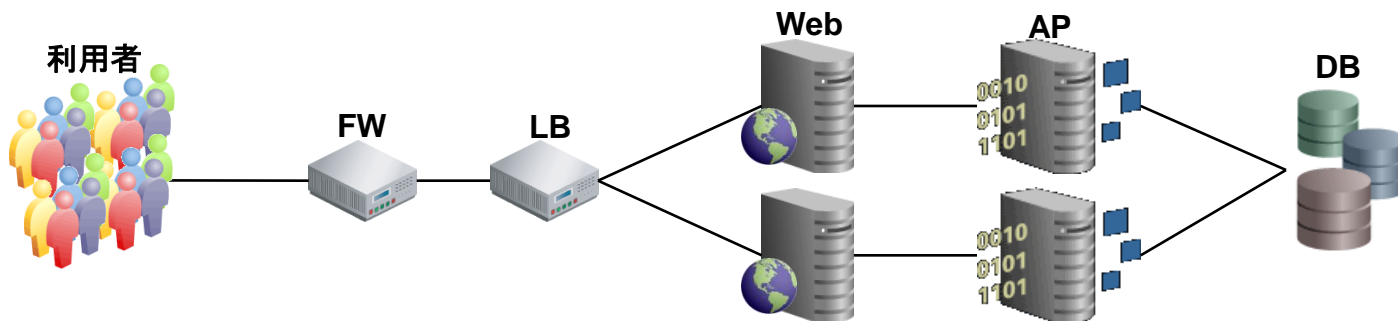
# 負荷テストを実施しなかったために・・・

- そのデータ、私のじゃない (某金融機関)
  - 営業支援系のWebシステム
  - 単体テストや結合テストでは大きな問題もなくクリア
  - 運用開始すると他の人の売上データが表示される
  - 変数のスコープが適切でなく、複数アクセスがあるとデータ不整合発生
- “地雷”と呼ばれたそのリンク (某通信業)
  - 社内イントラのWebシステム
  - 総合テストまでテスト用のダミーデータを利用
  - 本番データ移行後、データ件数が著しく増えSQLの性能劣化
  - あるリンクをクリックすると12時間ほど、応答が返ってこない・・・

# 性能は自分の目で確かめて

- Webシステムは複雑

- 様々なハードウェアやソフトウェアから構成されるWebシステム。それらの性能は誰も分かりません。メーカーから性能資料が提供されていても、複雑に構成されるWebシステムでそのまま適用できることは多くありません。
- 多くのWebシステムは既存のネットワーク上に構築されるため、ネットワーク機器などの影響も考慮する必要があります。
- 机上では計算できない性能は、実際にテストして自分の目で確かめるしかありません。



# テストしたいけど、人がいない

- これまで負荷テストの必要性をみてきましたが、現実的にテストができない(できてない)ことも少なくありません・・・
  - すでにコストやスケジュールが確定されており、今更組み込めない
  - 設計や開発、機能要件に対するテストに遅れが発生しているが、納期はズラせない
  - そもそも、性能テストを実施できる人がいない
- これらの問題をクリアするためにも、プロジェクト発足時から性能テストを必須として体制やスケジュールを検討する必要があります

# 負荷テストのサイクル

負荷テストを行う目的や目標を明確にし、スケジュールやリソースを検討する

計画

本番環境と同等の試験データやテストに必要な環境を確保する

準備

負荷テスト

分析・改善

策定した目標に到達しているか確認し、結果によってはチューニングなどを行い再テスト有無を判断する

実施

計画および準備に基づき、負荷テストを実施する



# Agenda

- Webアプリケーション負荷テストの基礎知識
- **テストデータの作成**
- 効果的な負荷テストの方法
- ボトルネックの切り分けと分析
- まとめ
- Appendix

## 無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
  - MySQLからの移行相談
  - PostgreSQLからの移行相談
  - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
- Oracle Developer/2000 Webアップグレード相談
  - パフォーマンス・クリニック
  - Oracle Database 構成相談
- Oracle Database 高可用性診断
  - システム連携アセスメント
  - システムセキュリティ診断
  - 簡易業務診断

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

# テスト現場の課題

- アプリケーション (Web またはクライアント / サーバー) の様々な試験 (総合、性能、バッチ) を行うことが多いが、テストデータを準備するのはいつも大変な作業である。
- 本番に近いデータを使った方が正確なテストができるでしょうが、どうすれば本番に近づくでしょうか。
- 今まで、本番のデータを使っていたが、プロセスとしてはよくないし、もっと気軽にテストできるようにしたい。

セキュリティポリシーにより、Sler / テスト関係者には本番データが出せない。

テストする側としては本番データを扱いたくない

# 本番データベースと同等のデータで各種テストを実施できるようにしたい理由

正確な負荷テストを行うためには、以下の条件でテストする必要があります。

- ・本番と同様のデータ量／質
- ・本番と同様のデータベース構成(パラメータ、索引など)

- システム開発環境
  - ・ 新システム開発時に、既存システムと同等のデータを利用したい
  - ・ ラッシュテスト時などに本番のデータを利用したい
- 影響度検証テスト環境
  - ・ サブシステム追加時に、既存システムへの影響を知りたい
  - ・ パッチ適用時に、既存システムへの影響を知りたい

# 質の良いテストデータを準備する方法は？

## テストデータを準備する手段

アプリケーションのクライアントを通して手動で登録
バッチプログラムでデータベースに直接登録
クライアントの操作をツールで自動化してスクリプトを使って登録
本番データベースを使用

## 現場の発言

「非常に工数がかかる」
「以前やってみただけど、アプリケーションとの整合性はなかなか取れない」
「これ自体もテストみたいなもので、準備に工数かかるし、自動化するのも上手くいかない」
「多くの人が扱うので心配」 「テストに使うのはどう考えてもいけないよね」

データの量/質は本番からずれるため正確な試験ができない可能性があります。

# テストに本番データを使うことによるリスク

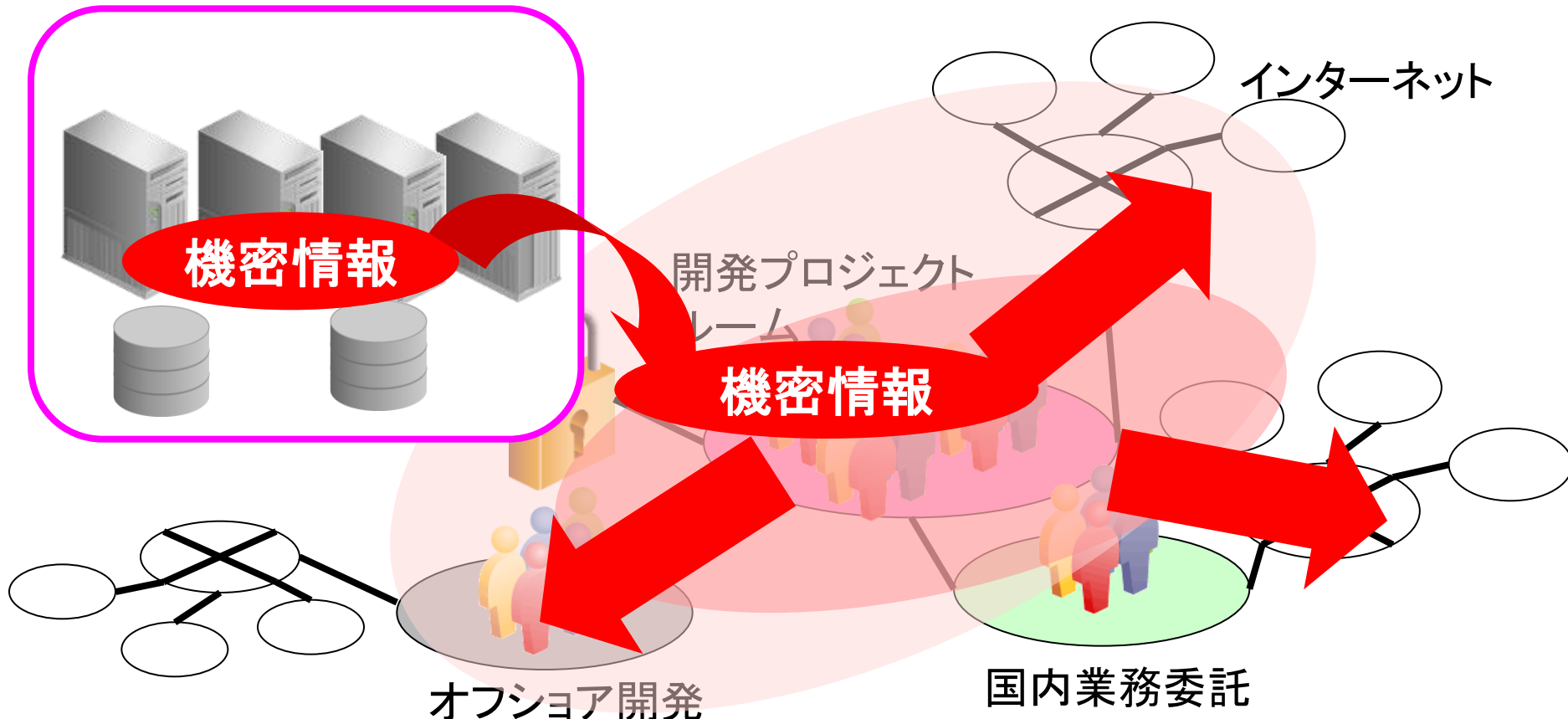
- ユーザーにとって
  - 機密情報／個人情報の漏洩
- 開発側にとって
  - 機密情報／個人情報の漏洩
  - 予想外の本番データ利用

こんなこともあり？（実際に起きたケース！）

- ワークフローシステムの試験に本番データを使い、処理完了メールが顧客に送信された！
- NDA を結んでも、本番データをテスト準備に使っているのがテスト関係者にきちんと伝わらず、実際に商品がお客様に出荷された！

# テストに本番データを使うことによるリスク

- ✓ 情報漏洩を発生させない為の鉄則は、本当に必要な場合以外、情報を外部に利用可能にさせない事である。
- ✓ 一度、出てしまった情報の機密性を取り戻す事は難しい。



# プライバシー・ポリシーに準拠したデータの共有

## ~ Oracle Data Masking Pack ~

開発環境、テスト環境およびステージング環境で機密情報をマスキングすることによって、組織がプライバシーおよび機密保護法を順守できるように支援します。



[表データの置換]

マスキング

ID	CARDNUMBER
1	7488-2984-1736-740
2	4033-6177-0089-0101
3	6141-5126-0475-0101
4	1139-4145-6222-0101
5	8337-6263-1608-0101
:	:

ID	CARDNUMBER
1	5870-2967-9149-5700
2	9634-7334-4874-2301
3	8430-8214-6445-1102
4	1573-9537-1503-5503
5	0606-3321-6271-8304
:	:

### 製品の特徴

- ・主キー/一意/参照整合性制約に違反することなくデータを置換
- ・置換の定義情報を残し、同じマスク処理を繰り返し実行可能
- ・マスキングはXML形式でエクスポート/インポート可能
- ・マスキング・フォーマットの充実
- ・高パフォーマンス

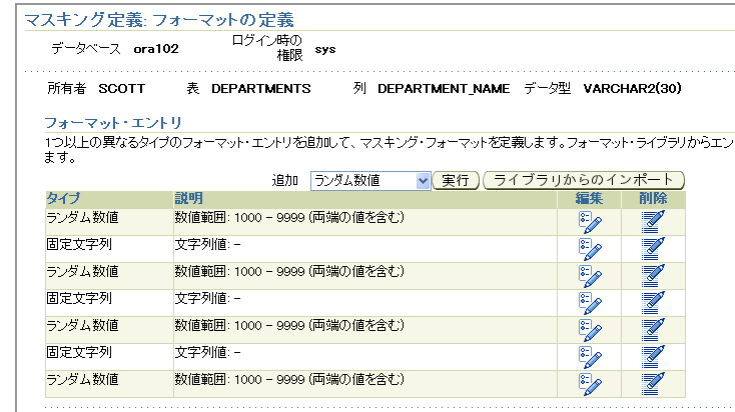
### 主な用途

- ・もっと簡単にテストデータを準備したい
- ・新システム開発時に、既存システムと同等のデータを利用したい
- ・負荷テストなどに本番データを使用したい
- ・サブシステム追加時に、既存システムへの影響を知りたい
- ・障害調査を行うために開発会社で本番データが必要だが、個人情報が多く含まれるため提供できない

# Oracle Data Masking Pack

センシティブなデータに対して多様なマスキングを容易に実現

- マスキングを簡単に定義、実行
  - GUIからの操作
  - マスキング定義を保存、再利用可能
- 多様なマスキングをサポート
  - 固定値、ランダム値、シャッフル、切り出し、など
  - 条件付きマスク、複数の列にまたがるマスク、など
- Oracle Databaseに最適化
  - 主キー / 一意 / 参照整合性制約を自動検知
  - マスク後、統計情報を自動リフレッシュ
  - カーディナリティの維持
  - パラレルDMLによる高速化 など



NAME
SMITH
ALLEN
JONES
CLARK
:

NAME
Aaafeh
Aafhed
Aaaafhe
Bodofa
:

ORACLE



# Oracle Data Masking Pack

## -多様なマスキングをサポート-

- 固定数値
- 固定文字列 (※1)
- ランダム桁数
  - 指定した範囲内の桁数でランダムの数値を生成
- ランダム数値
  - 指定した範囲内の数値をランダムに生成
- ランダム文字列(※2)
  - 指定した範囲内の文字数でランダムの文字列を生成
- ランダム日付
- 配列リスト (※1)
  - あらかじめ定義した文字列や数値の中からランダムに選ばれたものを使って元のデータを置換
  - (例)「東京、ロンドン、ニューヨーク、パリ」の中から選択して置換
- シャッフル (※1)
  - ほかの行のデータとランダムに置換
- 部分文字列 (※1)
  - もとのデータから指定した範囲を切り出して置換

※1 日本語対応

※2 日本語非対応

ORACLE

# Oracle Data Masking Pack

## -多様なマスキングをサポート-

- 削除
  - 行を削除(条件付マスクと共に使用)
- NULL値
  - NULL値を挿入(条件付マスクと共に使用)
- 元のデータの保持
  - 元のデータを保持(条件付マスクと共に使用)
- 切捨て
  - データの切捨て(truncate)
- 置換
  - 指定した表の指定した列の値をもとにハッシュベースの値を作成してマスク
- 表の列 (※1)
  - 指定した表の指定した列からランダムに値を取り出し置換
- ユーザー定義関数
  - ユーザーの作成した任意のPL/SQLスクリプトを使用してマスキングを実行

# Agenda

- Webアプリケーション負荷テストの基礎知識
- テストデータの作成
- 効果的な負荷テストの方法
- ボトルネックの切り分けと分析
- まとめ
- Appendix

## 無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
  - MySQLからの移行相談
  - PostgreSQLからの移行相談
  - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
- Oracle Developer/2000 Webアップグレード相談
  - パフォーマンス・クリニック
  - Oracle Database 構成相談
- Oracle Database 高可用性診断
  - システム連携アセスメント
  - システムセキュリティ診断
  - 簡易業務診断

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

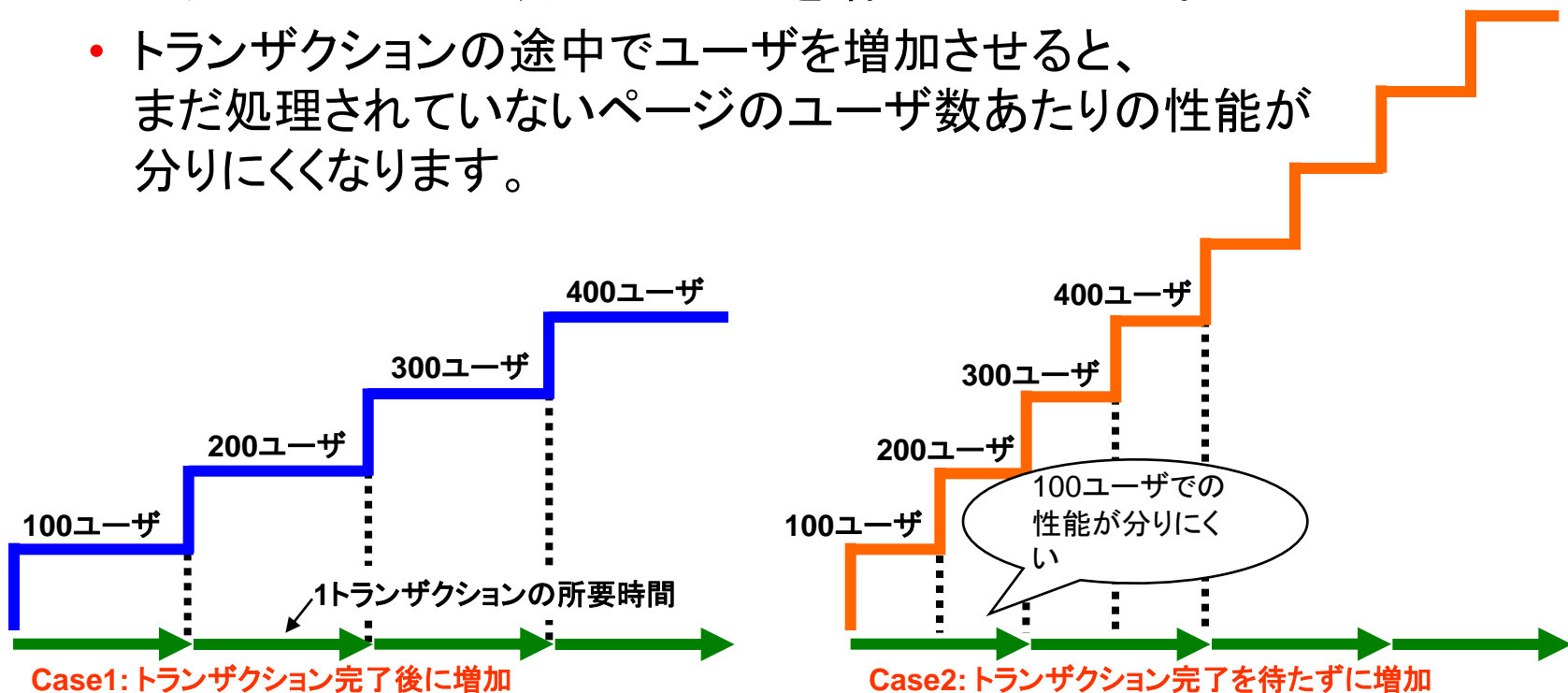
# 性能テストの手法 - テストも色々 -

- 性能テスト
  - 目標スループットの負荷を与えた時のパフォーマンス、システムの振る舞いを評価し、限界値(限界ポイント)までテストを続け、限界値を見極める
- ボリュームテスト
  - 大量のスループットもしくはコネクションを負荷装置より発生させ、ネットワーク機器・Webサーバの最大スループットや最大コネクション数の確認を行なう
- 限界テスト
  - システムの性能限界値を測定し、将来のサイジングや運用時の対応を予測・検討する
- 耐久テスト
  - 長時間にわたり負荷を与え続け、システムの挙動を確認する

# パフォーマンス検証の手法

## - ユーザの増加 -

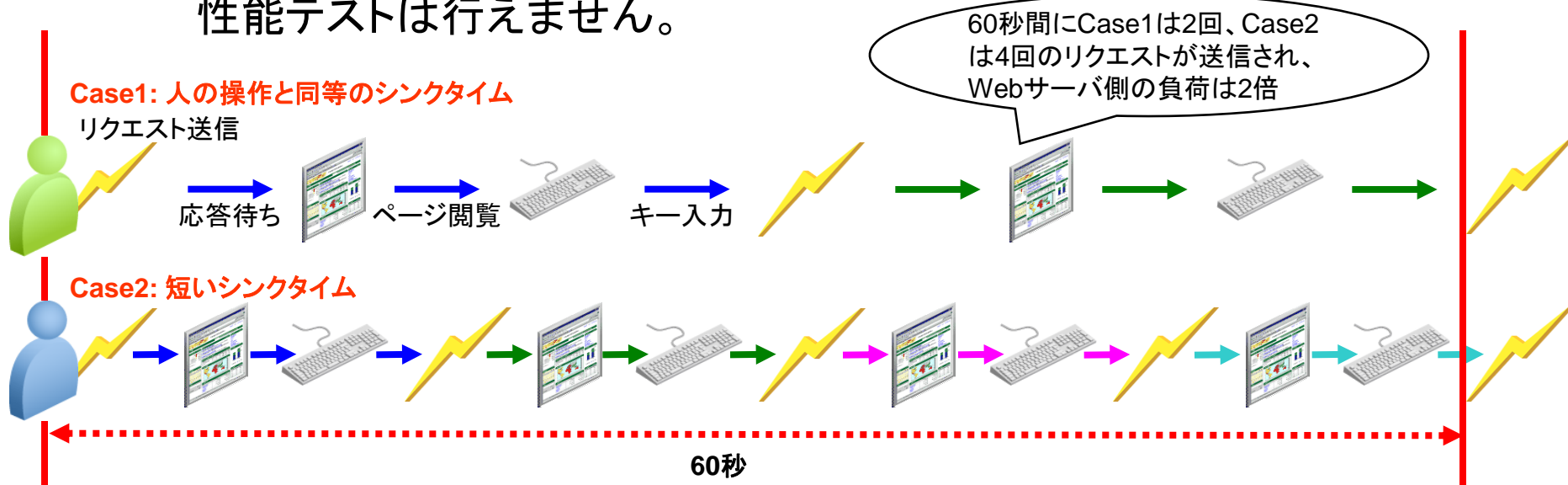
- ユーザの増加は余裕をもって
  - ユーザ数ごとに性能を計測するには、全ページ(トランザクション)が処理されてから、次のユーザを増やしてください。
  - トランザクションの途中でユーザを増加させると、まだ処理されていないページのユーザ数あたりの性能が分りにくくなります。



# パフォーマンス検証の手法

## - シンクタイム -

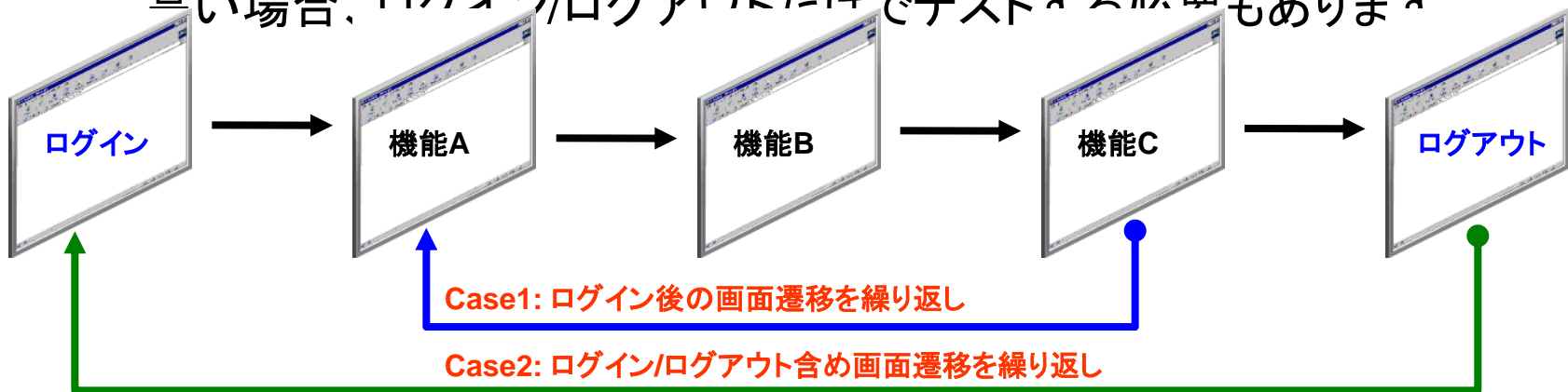
- ページ閲覧とキー入力の時間も考慮する
  - Webのページが表示されると、コンテンツを閲覧したりキーボードを入力する時間を経て、サーバにリクエストを送信します。
  - 性能テストでは、この時間(シンクタイム)を考慮する必要があります。シンクタイムが短かったり長かったりすると、運用を想定した性能テストは行えません。



# パフォーマンス検証の手法

## - ログイン/ログアウト -

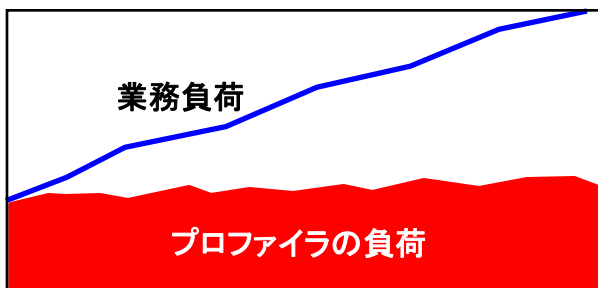
- ログイン認証も繰り返しますか
  - ログイン/ログアウトといった認証も繰り返す必要があるか、Webシステムの使われ方を想定して画面遷移を決定します。
  - グループウェアなどは認証後のセッション維持時間が比較的長い場合、業務機能のみを繰り返す画面遷移として性能を測定します。
  - 認証そのものやログイン後に表示されるポータル画面の負荷が高い場合、ログイン/ログアウトだけでテストする必要もあります。



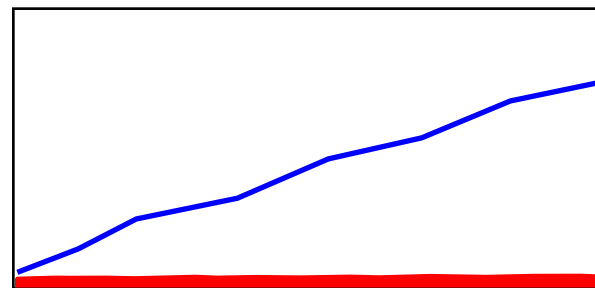
# パフォーマンス検証の手法

## - プロファイラーの活用 -

- プロファイラーが負荷をあげていませんか
  - アプリケーションコードの中で、どのクラス/メソッドのコストが高いか確認するのにプロファイラーは有効です。
  - しかしプロファイラーによっては詳細な情報を取得するため、サーバ側にも相応の負荷がかかってしまいます。
  - 利用する目的や環境などによって、最適なプロファイラーを選択してください。特に性能テスト時はなるべく負荷の低いプロファイラーをお勧めします。



Case1: プロファイラのCPU負荷が高い



Case2: プロファイラのCPU負荷が低い



# パフォーマンス検証の手法

## - データベースキャッシュ -

- その性能はキャッシュによるものでは？
  - 同じ操作、リクエストでは、キャッシュにより本来必要な負荷がかからない場合があります。
  - 実運用時のキャッシュヒット率を想定し、リクエストを動的に変更したテストが必要です。

応答時間も早いし、  
負荷も低く性能要件  
を達成したぞ！



AP

キャッシュを利用

結果

クエリー

Select \* from A where name = "oracle"  
Select \* from A where name = "oracle"  
Select \* from A where name = "oracle"



(SGA)

必要な負荷がかかっていないため  
リソースは軽微

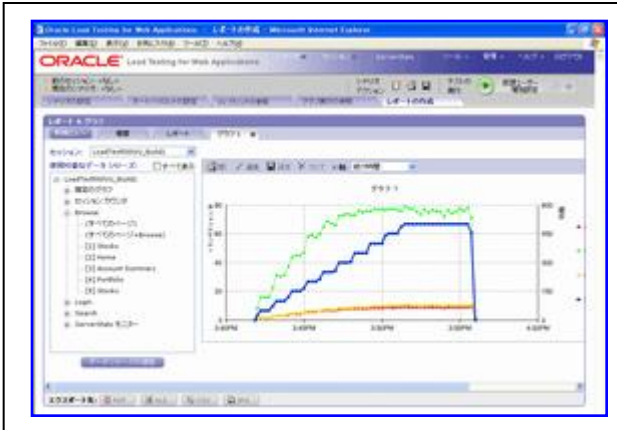


DB

# Webアプリケーションに対する負荷テスト

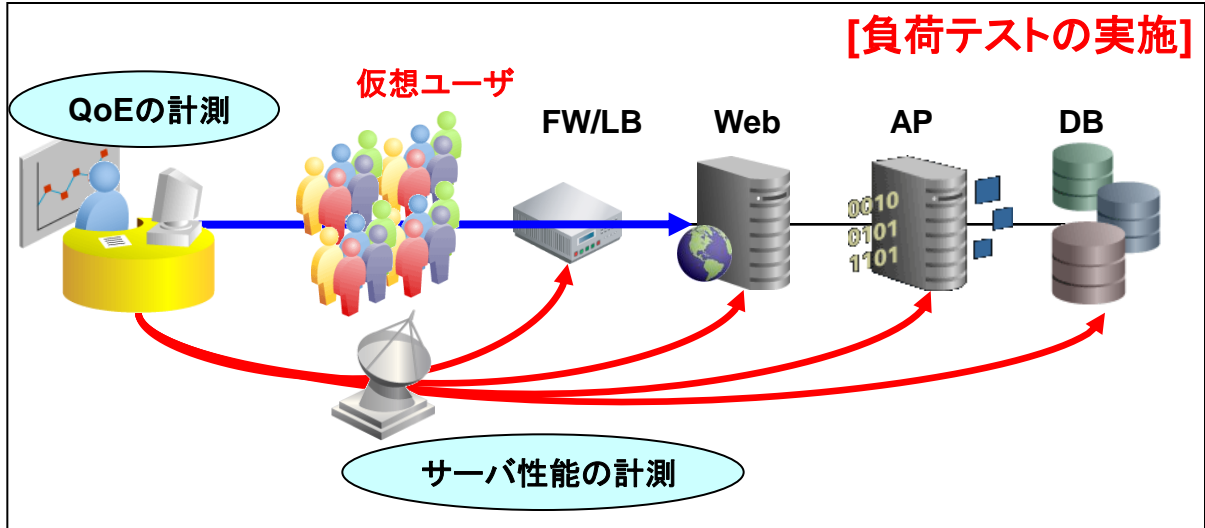
## ～ Oracle Load Testing ～

多数のユーザからのアクセスを擬似しWebアプリケーションの性能検証を実現。運用を開始する前に、アプリケーションの性能問題をあぶりだし、Quality of Experienceの向上を支援します。



### 製品の特徴

- ・GUIによる簡単なスクリプト作成
- ・CookieやHTML内のセッションデータも自動パラメータ化
- ・ユーザ視点のエラーチェック
- ・OS/AP/Network等の性能データをエージェントレスで収集
- ・見やすい分析グラフ
- ・HTTP(S)/SOAPに対応



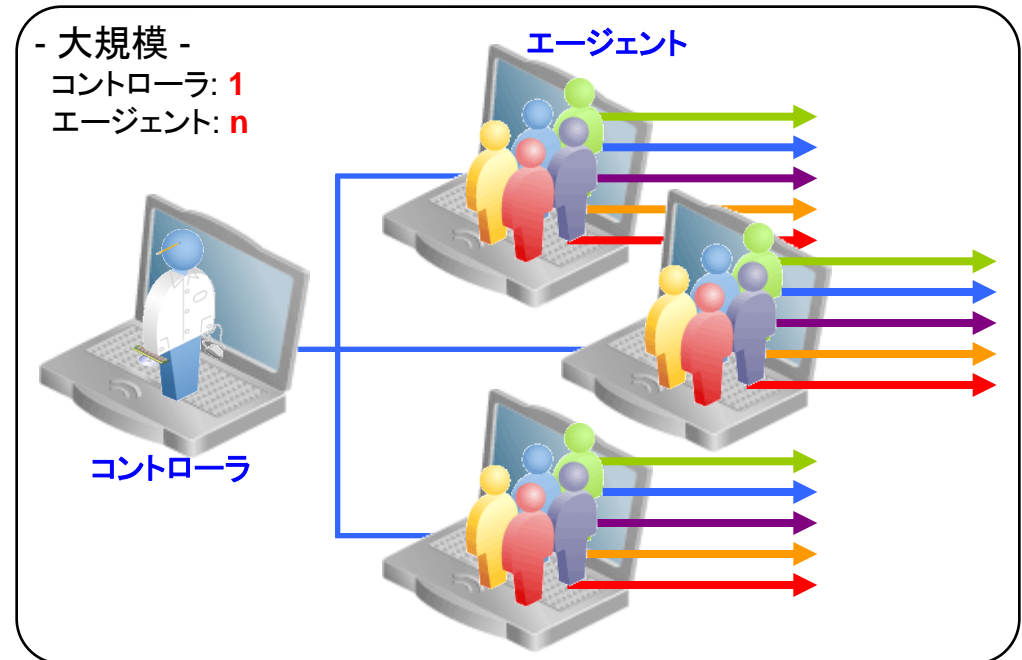
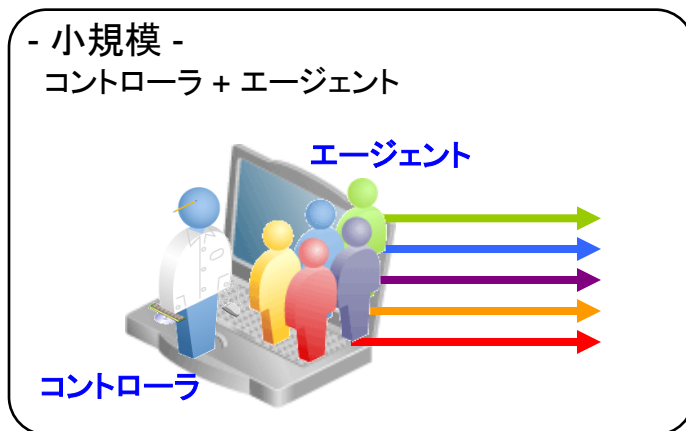
### 主な用途

- ・開発の早い段階から手軽に負荷テストを実施したい。
- ・応答時間の遅延の原因となるサーバを特定したい。
- ・想定していないエラー画面を見落としたくない。
- ・PCだけでなく携帯や専用端末のアプリケーションもテストしたい。
- ・サーバにモジュールを導入することなく性能を計測したい。
- ・機能テストで作成したスクリプトを活用したい

# Oracle Load Testing

## -大規模負荷テストへの対応-

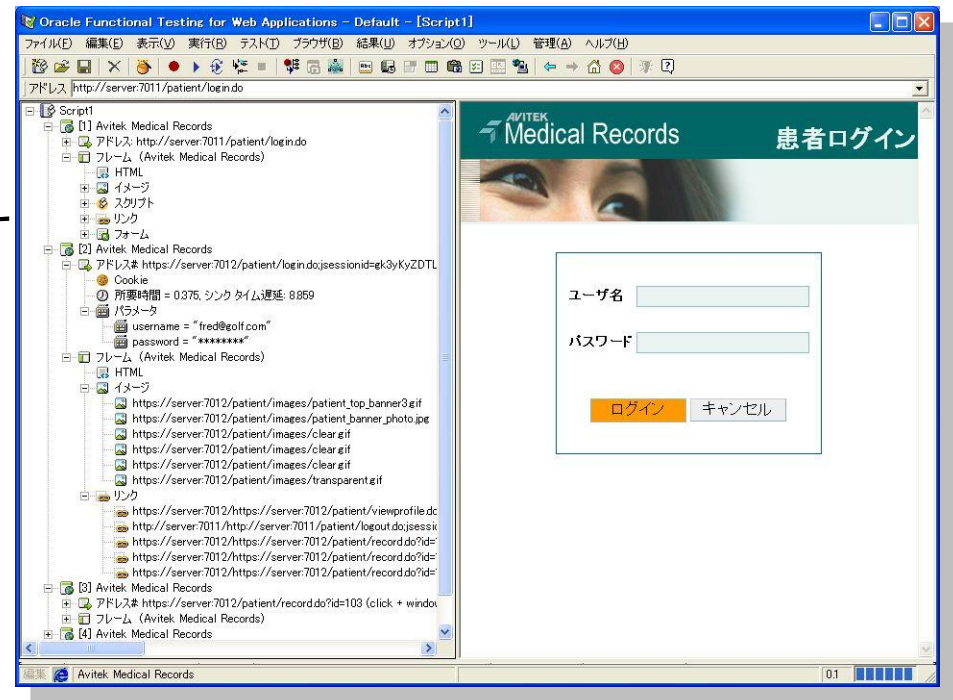
- コントローラとエージェントを負荷条件により柔軟に配置
  - コントローラ ... エージェントの制御,性能情報の取得,レポートの作成
  - エージェント ... Webサーバへの同時アクセス



# Oracle Load Testing

## -GUIによる簡易なテストスクリプトの記録-

- スクリプトの作成にプログラミングは不要
  - テストを行いたい画面遷移をブラウザと同じ感覚で操作するだけで、テスト用のスクリプトが記録されます
  - テストスクリプトには、URL、クエリースtringやPOSTデータ、Cookieなどの情報が自動的に保存されます



# Oracle Load Testing

## -HTTPリクエストのセッションを自動処理-

- 常に最新のセッションIDを処理してテストを実施
  - Webアプリケーションはセッションによる管理を行っており、複数ユーザーが同じセッションIDを利用したり有効期限が切れるとエラーになります
  - Load TestingではCookie/URL/HTML埋め込みなどのセッションIDを自動的に処理し、常に正しいリクエストを送信します
  - Weblogic、Microsoft ASP/ASP.NET、WebSphere、Cold Fusionなど、数多くの一般的なWeb開発環境に対して最適化されています



# Oracle Load Testing

## -データドリブン型の負荷テスト-

- 異なるデータに同時にアクセスするシナリオをエミュレート
  - CSVファイルに定義したデータを用い、仮想ユーザーごとに異なるユーザー名、パスワード、口座残高などを使うことができます

データバンク  
(CSV形式)

```
UserID, Passowrd  
fred@golf.com, as98asds  
larry@bball.com, op2dg87d  
page@fish.com, qw9fd95q
```

患者情報: Winner, Fred I | 生年月日: 03/26/1965 | ログアウト

ホーム

訪問

日付	訪問理由	医師
07/05/1989	Twisted knee while playing soccer.	Dr. Phil B Lance
05/30/1993	Sneezing, coughing, stuffy head.	Dr. Kathy E Wilson
06/18/1999	Complains about chest pain.	Dr. Phil B Lance

処方箋

開始日	薬	服用量	頻度	補充	説明
05/30/1993	Codeine	10 oz	1/6hrs	1	No instructions
06/18/1999	Advil	100 tbls	1/4hrs	0	No instructions
06/18/1999	Drixoral	16 oz	1tspn/4hrs	0	No instructions

仮想ユーザー 1

患者情報: Parrot, Larry J | 生年月日: 02/13/1959 | ログアウト

ホーム

訪問

日付	訪問理由	医師
06/20/2001	Overjoyed with everything.	Dr. Phil B Lance

処方箋

開始日	薬	服用量	頻度	補充	説明
06/20/2001	Valium	50 pills	1/day	3	No instructions

仮想ユーザー 2

患者情報: Trout, Page A | 生年月日: 02/18/1972 | ログアウト

ホーム

訪問

日付	訪問理由	医師
01/13/1987	Blurred vision.	Dr. Phil B Lance
09/09/1990	Sore throat.	Dr. Mary J Oblige
04/01/1991	Drowsy all day.	Dr. Mary J Oblige

処方箋

開始日	薬	服用量	頻度	補充	説明
薬は処方されていません。					

仮想ユーザー 3

# Oracle Load Testing

## -ユーザー視点によるエラーチェック-

- コンテンツの内容を検証しエラーを判定
  - ページの応答時間やサーバーの性能に問題がなくても、ユーザーの期待していないコンテンツが表示されている可能性があります
  - サーバーの負荷が高くなると、“ただいま混み合っています”などSorry Serverへ転送されてしまう場合もあります
  - Load TestingではWebサーバーのレスポンスコード(4xx,5xx)に加え、コンテンツ(HTML)が正しいかユーザー視点でのチェックを行います
  - エラーコンテンツはリアルタイムに確認でき、問題追跡を容易にします

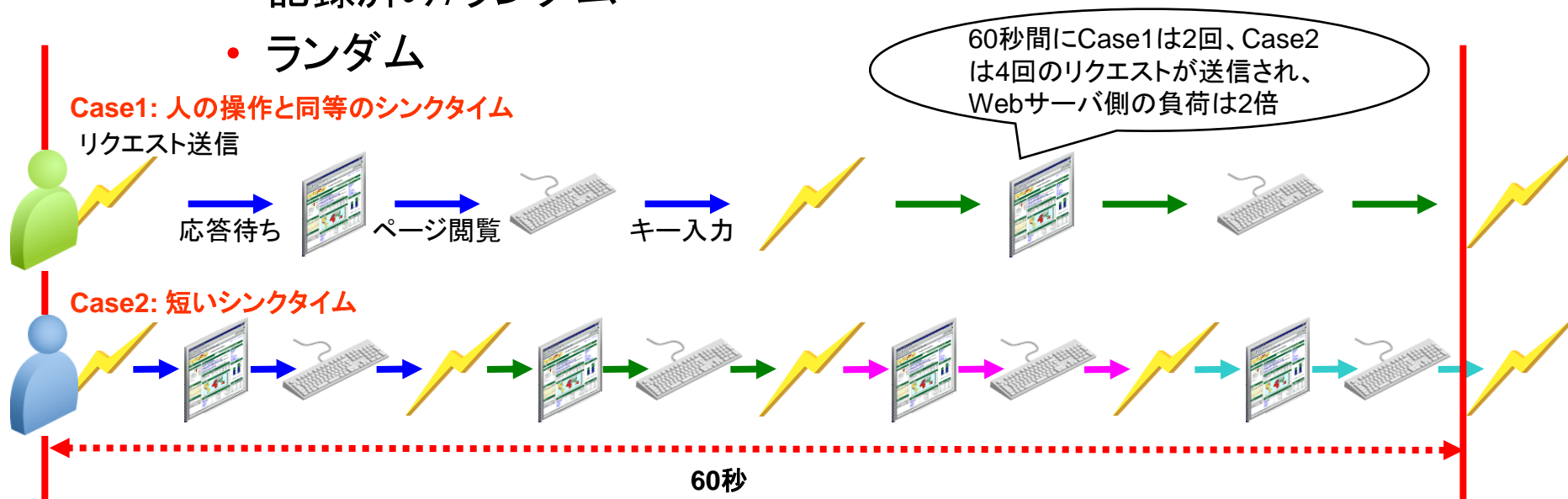
患者登録が失敗しました。システム管理者にお問い合わせください。

Eメール	<input type="text" value="loadtest01@oracle.com"/>	住所	
パスワード (8-10 chars)	<input type="password" value="●●●●●●"/>	番地	<input type="text" value="2-5-8"/>
名	<input type="text" value="Load"/>		<input type="text"/>
ミドルネーム	<input type="text"/>	市	<input type="text" value="Minato"/>
姓	<input type="text" value="Test"/>	州	<input type="text" value="NY"/>
性別	<input type="text" value="男"/>	郵便番号	<input type="text" value="1070061"/>
生年月日 (mm/dd/yyyy)	<input type="text" value="01/01/2000"/>	国	<input type="text" value="Japan"/>
ソーシャルセキュリティ番号 (xxxxxxxxxx)	<input type="text" value="000000001"/>		
電話 (xxx-xxx-xxxx)	<input type="text" value="001-001-0001"/>		

# Oracle Load Testing

## -思考遅延時間により正確なテストを実現-

- コンテンツの閲覧やキー入力に必要な時間を設定
  - ページが表示されてからリクエストを送信するまでの時間(思考遅延時間/シンクタイム)を柔軟に定義することができます
    - 記録済み時間
    - 記録済み/ランダム
    - ランダム

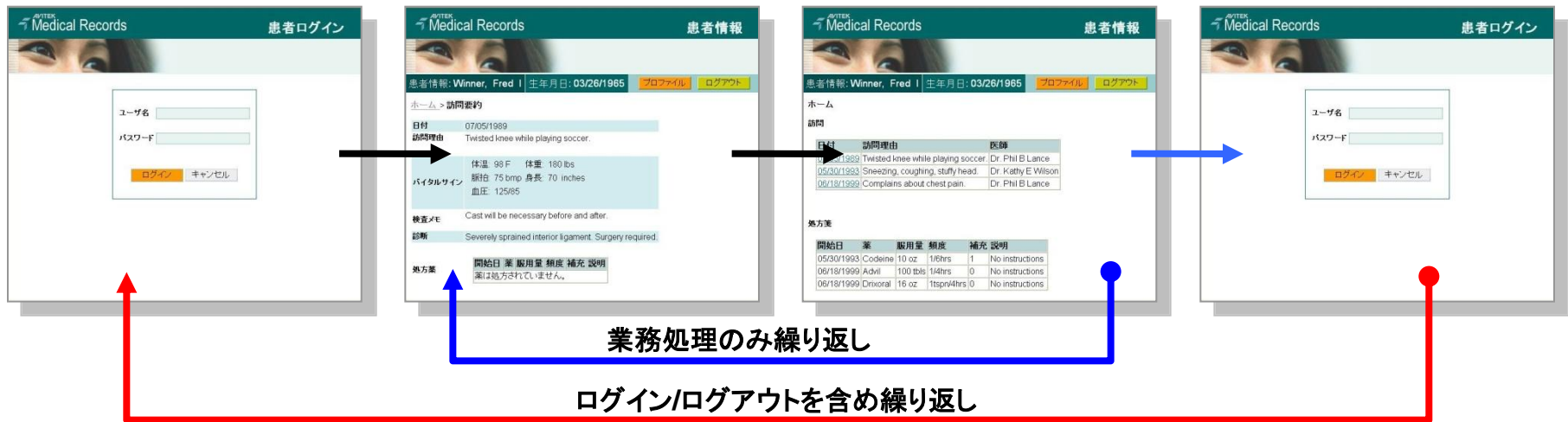




# Oracle Load Testing

## -複数スクリプトの組み合わせ-

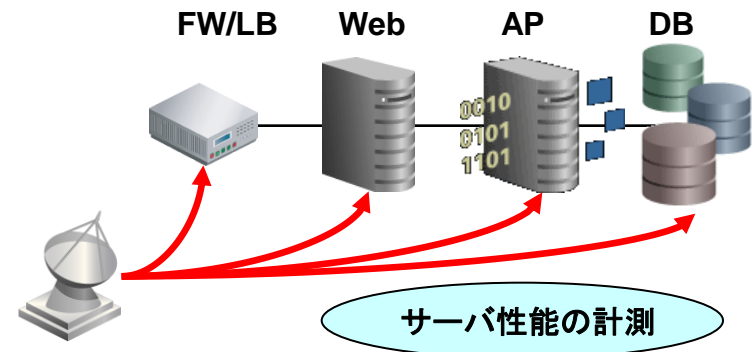
- ログイン後の処理のみ繰り返し
  - グループウェアなど認証後のセッション維持時間が比較的長いアプリケーションは業務機能を繰り返す画面遷移として定義できます
  - 認証そのものやログイン後に表示されるポータル画面の負荷が高い場合、ログイン/ログアウトだけでテストする必要もあります
  - すべての仮想ユーザーが同時に処理する同期ポイントも設定できます



# Oracle Load Testing

## -サーバー性能監視-

- エージェントレスによるリアルタイム監視
  - さまざまなアプリケーション、データベース、システム、ネットワーク機器などのリソース情報を監視することができます
  - 対象システムにエージェント等を導入する必要はありません
  - 監視可能なリソース
    - Windows OS (Perfmon)
    - Solaris/Linux (Telnet/SSH)
    - Application Server (JMX/SNMP)
    - Network Device (SNMP)
    - Database (SQL)
    - Web Page (URL)
    - Ping, COM+ ...



# Oracle Load Testing -テスト結果の容易な分析-

- テスト結果をリアルタイムで参照
  - パフォーマンスデータは、スクリプト、ページ、ページ上のオブジェクトなど、さまざまなレベルで蓄積できます
  - グラフおよびレポートには、レスポンス時間、エラー発生率、ユーザー数、およびヒット数/秒やページ数/秒などの統計情報が表示されます



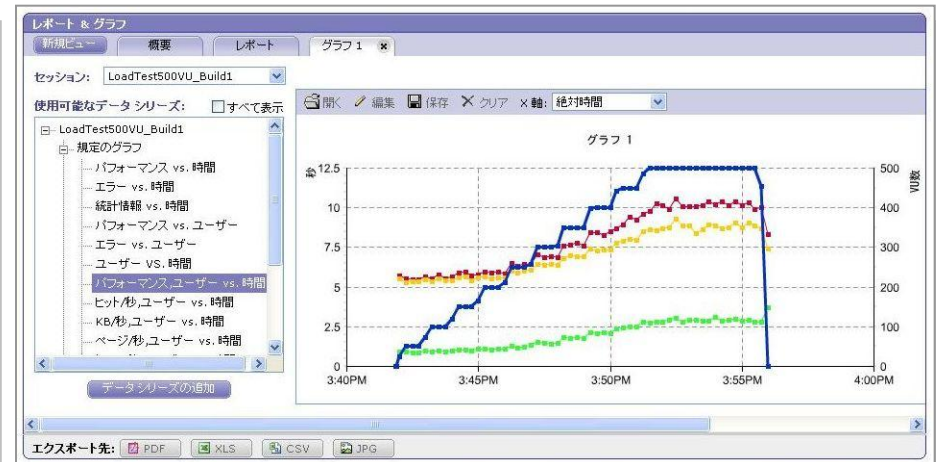
レポート & グラフ

セッション: LoadTest500VU\_Build1 レポート: セッションレポート

プロフィール及びリソースごとの応答時間 \*

名前	最小	最大	平均	成功	失敗	標準偏差	90th %
Browse	5.483	10.532	7.923	5865	0	1.873	10.24
Browse.[1] Stocks	0.004	0.721	0.275	5865	0	0.241	0.603
Browse.[2] Home	0.041	1.522	0.742	5865	0	0.518	1.341
Browse.[3] Account Summary	0.026	1.076	0.549	5865	0	0.376	1.015
Browse.[4] Portfolio	0.019	1.081	0.543	5865	0	0.376	1.002
Browse.[5] Stocks	0.009	0.863	0.336	5865	0	0.309	0.734
Login	0.873	3.672	1.948	53038	0	0.837	2.929
Login.[1] Stocks	0.01	0.809	0.306	53038	0	0.259	0.638
Login.[2] Home	0.048	1.439	0.761	53038	0	0.526	1.389
Search	5.305	9.304	7.113	6481	0	1.395	8.888
Search.[1] Stocks	0.005	0.631	0.257	6481	0	0.229	0.568
Search.[2] Home	0.028	1.48	0.741	6481	0	0.53	1.408
Search.[3] Ticker List	0	0.689	0.252	6481	0	0.233	0.563

エクスポート先: CSV 印刷用画面: 表示 プロファイル タイマーシンク タイムを含める



# Agenda

- Webアプリケーション負荷テストの基礎知識
- テストデータの作成
- 効果的な負荷テストの方法
- ボトルネックの切り分けと分析
- まとめ
- Appendix

## 無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
  - MySQLからの移行相談
  - PostgreSQLからの移行相談
  - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
- Oracle Developer/2000 Webアップグレード相談
  - パフォーマンス・クリニック
  - Oracle Database 構成相談
- Oracle Database 高可用性診断
  - システム連携アセスメント
  - システムセキュリティ診断
  - 簡易業務診断

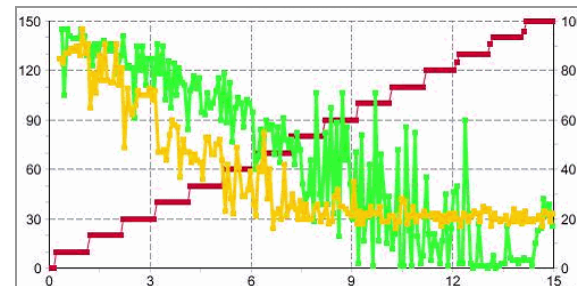
<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

# 単体テストはしっかりやったが・・・



アプリケーションの負荷テストに入ったけど、すぐにパフォーマンスが頭打ちになるぞ？



単体テストは十分やったんですが・・・



C/Oは二週間後だ、それまでによろしく

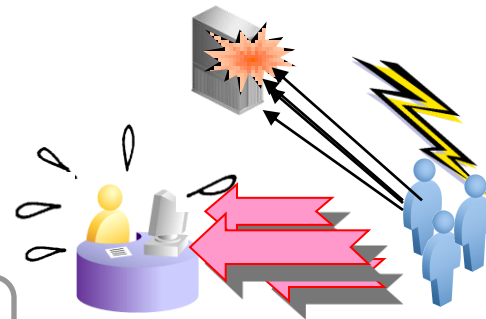


簡単に言うけど、ここまで大規模になると分析は大変。必要なデータを取るだけでもかなりの作業だ・・・



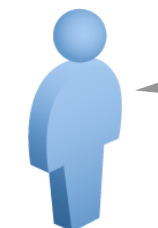
結合・総合テスト以降の問題分析が困難

# DBのパフォーマンスが悪い？



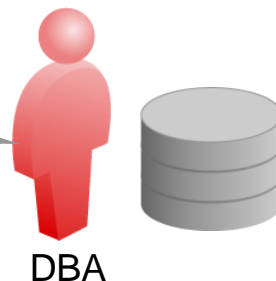
アクセスが増える時間帯になると「遅い」とクレームが来るんですが、DBを調べてもらえませんか？

・・・CPUもI/Oも特に異常は見られないし、コストの高いSQLもなさそうですけど



そんなこと言ってもなんとかしないとイケないんですよ

(問題はJavaのほうにあるのでは・・・)



DBA



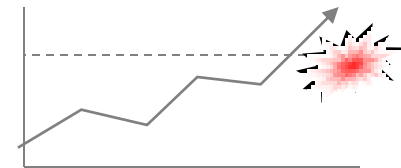
Java-DB間のボトルネック切り分けに手間取る

# メモリーリークの原因は？



開発者A

ヒープ使用率がどんどん上昇している・・・  
まさかメモリーリーク！？

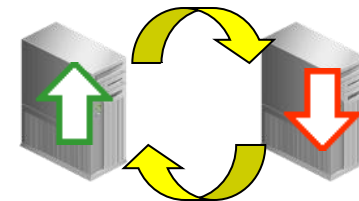


開発時にはみつからなかったのに・・・  
かといってどこでリークしているか調べるのは大変だ



開発者B

しかたがない、毎日夜間に再起動してもらおう  
しかし日中にOutOfMemoryにならないか心配だ・・・



本番環境に入ってからメモリーリークに悩まされる

# Javaはブラックボックス化する

ブラックボックス化したJavaアプリケーション内部のボトルネックをリアルタイムに分析することは困難

- システムのスループットやレスポンスにとって Javaアプリケーションのチューニングは、DBのチューニングと並んで重要な項目
- そのため、開発時にはプロファイラを使ってボトルネックを調査しチューニングすることが多いが、いったん運用に入ると塩漬け化、ブラックボックス化してしまうことがほとんど
- しかし本番環境でしか発生しない問題がある。開発環境に移しても再現しなかったり、時間がかかったりする
- DB同様に本番環境をそのまま分析できるのが一番良いが、既存のプロファイラではオーバーヘッドが大きく、現実的でない

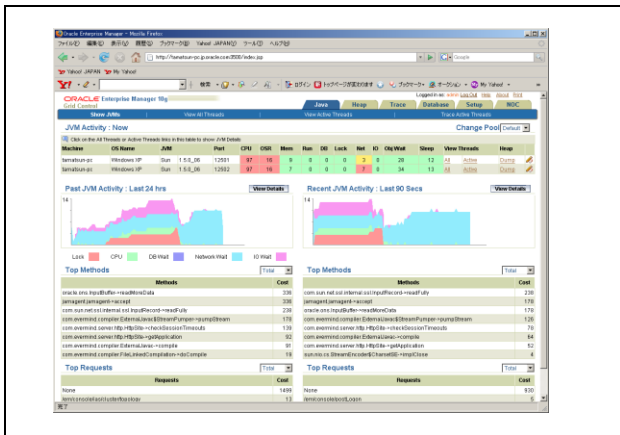




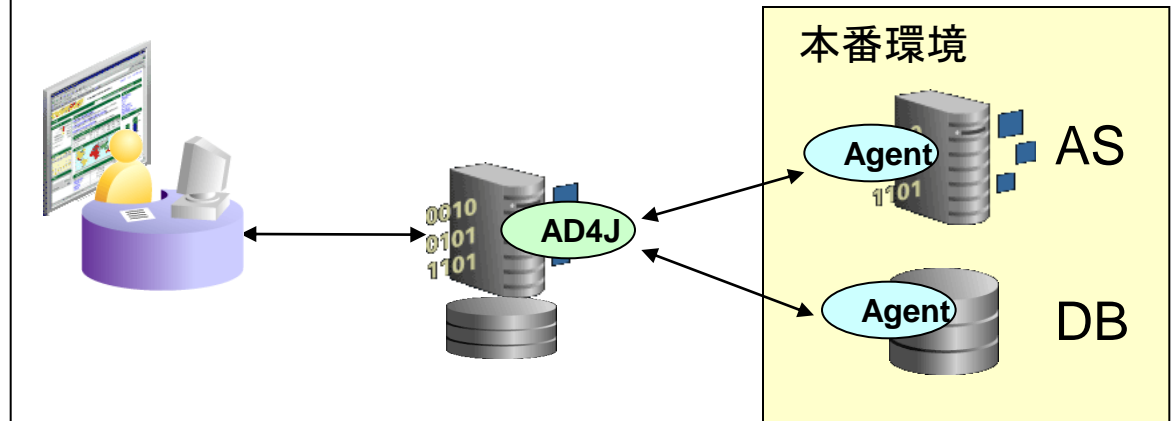
# Javaアプリケーションの詳細な分析

## ～ Oracle Application Diagnostics for Java (AD4J) ～

AD4Jを使用することにより、管理者はJavaアプリケーションの状態をプロアクティブに監視し、問題発生時にボトルネックとなっているオブジェクトを素早く特定することができます。



### [Javaアプリケーションの監視]



### 製品の特徴

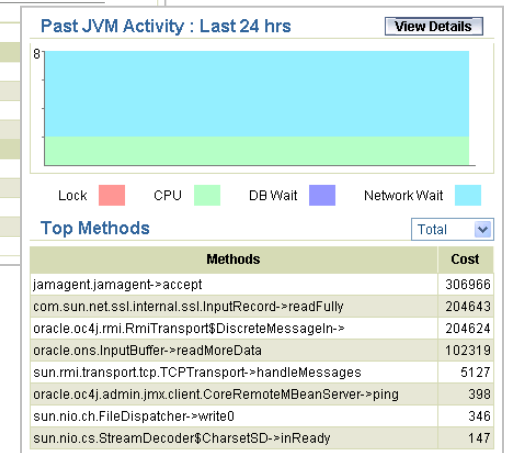
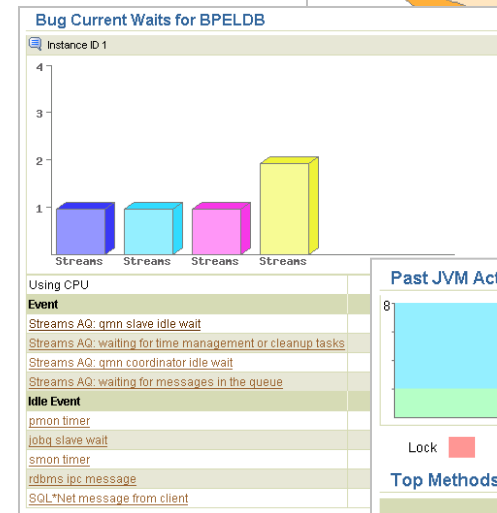
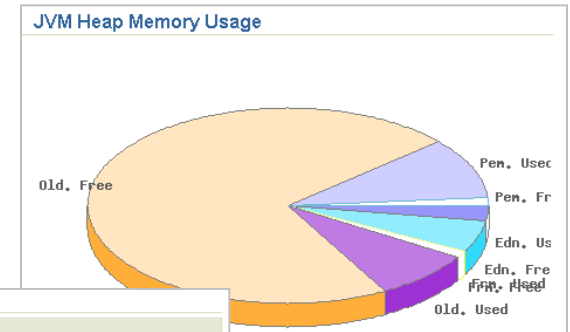
- ・負荷が軽く、本番環境で使用可。
- ・AS層、DB層にまたがるトランザクションを追跡
- ・ほとんどのアプリケーション・サーバーをサポート
- ・インストール/セットアップが簡単。
- ・Agentデプロイ時に再起動の必要なし。

### 主な用途

- ・異常を早く検知したい。  
(しきい値設定とメール/SNMPトラップの通知)
- ・メモリ・リーク時の原因オブジェクトを特定したい。
- ・デッドロックのオブジェクトを検出したい。
- ・パフォーマンス劣化の原因となっているオブジェクトを特定したい。
- ・定期的にJVMの状況を監視したい

# Oracle Application Diagnostics for Java

- オラクルのJava-DB診断ソリューション
  - Oracle Enterprise Managerのツール群の一つ
  - Java アプリケーションの詳細なパフォーマンス分析、ヒープ分析などの機能を本番環境で提供可能



Methods	Cost
jamagent.jamagent->accept	306966
com.sun.net.ssl.internal.ssl.InputRecord->readFully	204643
oracle.oc4j.rmi.RmiTransport\$DiscreteMessageIn->	204624
oracle.ons.InputBuffer->readMoreData	102319
sun.rmi.transport.tcp.TCPTransport->handleMessages	5127
oracle.oc4j.admin.jmx.client.CoreRemoteMBeanServer->ping	398
sun.nio.ch.FileDispatcher->write0	346
sun.nio.cs.StreamDecoder\$CharsetSD->inReady	147

Javaアプリケーションの詳細なパフォーマンス分析

DBにまたがる問題分析

JVMヒープ分析

分析時のオーバーヘッドが低い

# Oracle Application Diagnostics for Java

## -基本診断機能-

ORACLE Enterprise Manager 10g Grid Control

Java Heap Trace Database Setup NOC

Show JVMs View All Threads View Active Threads Trace Active Threads

JVM Activity : Now Change Pool Default

Click on the All Threads or Active Threads links in this table to show JVM Details

JVM Name	OS Name	JVM	CPU	OSR	Mem	Run	DB	Lock	Net	IO	Obj Wait	Sleep	View Threads	Heap
jpdelvm1.jp.oracle.com:as1013_1.home.default_group.1	Linux	Sun 1.5.0_06	20	7	19	0	0	7	4	0	82	12	All Active Dump	

JVM内の処理の内訳をグラフ表示  
(ロック待機/CPU/DB待機など)  
Java-DB間の切り分けを容易に実施

Recent JVM Activity : Last 90 Secs View Details

Lock CPU DB Wait Network Wait IO Wait

Auto Refresh  Every 15 secs

Top Methods

Methods	Cost
com.sun.net.ssl.internal.ssl.InputRecord->readFully	19196
org.springframework.samples.jpeteststore.dao.ibatis.SqlMapOrderDao->insertOrder	14596
oracle.ons.InputBuffer->readMoreData	9598
com.evermind.io.IOUtils->readCompressedInt	6160

Top Methods

Methods	Cost
org.springframework.samples.jpeteststore.dao.ibatis.SqlMapOrderDao->insertOrder	622

「待ち」の原因となっているメソッドを即座に判別

Top Requests

# Oracle Application Diagnostics for Java

## -基本診断機能-

System Activity : 11-AUG-09 22:48:00 - 11-AUG-09 22:49:30 mobacle.jp.oracle.com:as1310\_mobacle.OC4J\_BANKCRM.default\_group.1 1 CPU(s) 1152 MB Heap

**Active Threads** | **Heap Usage**

**CPU Utilization**

Legend: Lock (red), CPU (green), DB Wait (blue), Network Wait (cyan), IO Wait (magenta)

Top Methods		Top Requests		Top DBStates		Top SQL	
Methods	Cost	Requests	Cost	DBStates	Cost	SQL	Cost
Before_Servlet2->doPost	464	/BANKCRM/servlet2	464	CPU	316	1936493789	296
Before_Servlet1->doPost	318	/BANKCRM/servlet1	318	db file scattered read	106	1928436760	159
oracle.ons.InputBuffer->readMoreData	88	None	88	latch: cache buffers chains	91	2719912248	119
				null	86	0	86

**SQL Statement**

```
SELECT C.CIF_NO CIF_NO , C.NAME_SEI || C.NAME_MEI NAME
FROM RB_CIF C ,
     RB_CIFNAYOSE S
WHERE C.CIF_NO = S.CIF_NO
      AND S.SETCIF_NO = 9000001
```

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation.

DB待ちが発生している場合、発行しているSQL文まで特定可能

ボトルネックが発生しているページやメソッドをGUIから迅速に特定可能

# Oracle Application Diagnostics for Java

## -応用診断機能-

Javaの問題分析に必要な  
情報を本番環境で提供

どのコードが  
問題？



開発担当者

```
Stack Trace
View Totals By Percent
View Totals By Percent
View Lock By Percent
View CPU By Percent
View DB By Percent
View DB By Abs Numbers
View IO By Percent
View IO By Abs Numbers
View Net By Percent
0.35% java.lang.String->
0.35% java.lang.Object->clone
99.30% jamagent.jamrun->main
99.30% java.lang.reflect.Method->invoke
99.30% sun.reflect.DelegatingMethodAccessorImpl->invoke
99.30% sun.reflect.NativeMethodAccessorImpl->invoke
99.30% org.apache.catalina.startup.Bootstrap->main
81.12% org.apache.catalina.startup.Bootstrap->start
81.12% java.lang.reflect.Method->invoke
81.12% sun.reflect.DelegatingMethodAccessorImpl->invoke
81.12% sun.reflect.NativeMethodAccessorImpl->invoke
81.12% sun.reflect.NativeMethodAccessorImpl->invoke
81.12% org.apache.catalina.startup.Catalina->start
81.12% org.apache.catalina.core.StandardServer->start
72.38% org.apache.catalina.core.StandardService->start
8.74% org.apache.catalina.util.LifecycleSupport->fireLifecycleEvent
18.18%
```

ボトルネックを  
メソッドレベルまで追跡

問題なのは  
Javaコード？  
それともDB？



DBStates	Cost
enqueue	122
db file scattered read	30

SQL Statement

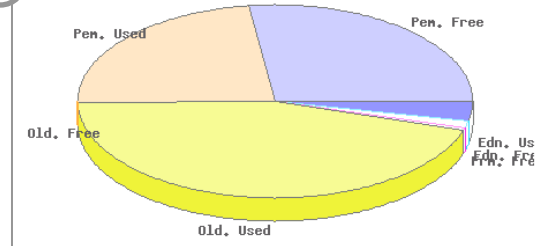
```
SELECT userid , status , email , firstname , lastname , addr1 , addr2 , city , state , zip , country , phone
FROM account
WHERE userid = :1
```

パフォーマンス低下の原因となっているSQL  
までを特定

メモリリークが  
発生！原因は  
どこ？



JVM Heap Memory Usage



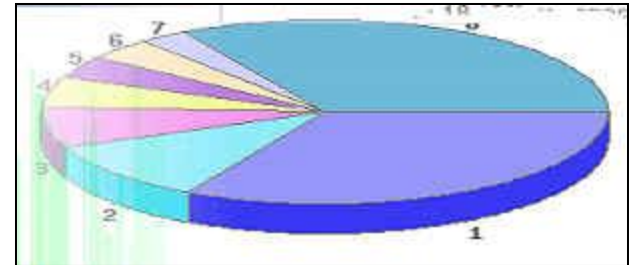
ヒープメモリの中身を分析し、メ  
モリリークの原因を特定

ORACLE

# Oracle Application Diagnostics for Java

## -スレッドトレース分析-

- リアルタイムにアクティブなスレッドをトレース
- ボトルネックになっているリソースを表示



### Stack Trace

```
100.00% jamagent.jamrun->main
  100.00% java.lang.reflect.Method->invoke
    100.00% sun.reflect.DelegatingMethodAccessorImpl->invoke
      100.00% sun.reflect.NativeMethodAccessorImpl->invoke
        100.00% sun.reflect.NativeMethodAccessorImpl->invoke0
          100.00% project1.JDBC_Test->main
            100.00% oracle.jdbc.driver.OracleStatement->execute
              100.00% oracle.jdbc.driver.OracleStatement->doExecute
                100.00% oracle.jdbc.driver.T4CStatement->execute
                  100.00% oracle.jdbc.driver.OracleStatement->execute
                    58.12% oracle.jdbc.driver.T4CStatement->execute
                      58.12% oracle.jdbc.driver.T4CStatement->execute
                        58.12% oracle.jdbc.driver.OracleStatement->execute
                          58.12% oracle.jdbc.driver.T4CStatement->execute
                            58.12% oracle.jdbc.driver.T4CStatement->execute
                              58.12% oracle.net.ns.NetInputStream->read
                                58.12% oracle.net.ns.NetInputStream->read
                                  58.12% oracle.net.ns.NetInputStream->getNextPacket
                                    58.12% oracle.net.ns.DataPacket->receive
                                      58.12% oracle.net.ns.Packet->receive
                                        58.12% java.net.SocketInputStream->read
                                          58.12% java.net.SocketInputStream->socketRead0
                                            58.12% Line #0
                              41.88% oracle.jdbc.driver.T4CStatement->execute_for_rows
```

View Totals By Percent

- View Totals By Percent
- View Totals By Abs Numbers
- View Lock By Percent
- View Lock By Abs Numbers
- View CPU By Percent
- View CPU By Abs Numbers
- View DB By Percent
- View DB By Abs Numbers
- View IO By Percent
- View IO By Abs Numbers
- View Net By Percent
- View Net By Abs Numbers

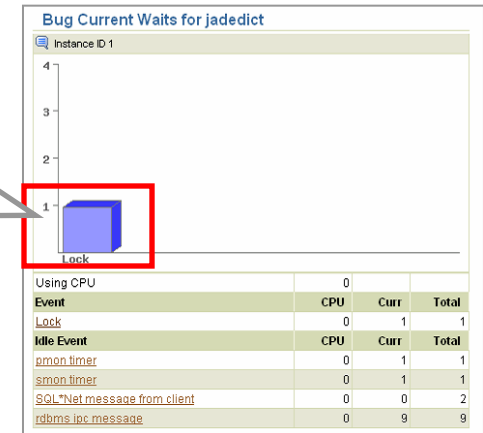
# Oracle Application Diagnostics for Java -Java/Oracle間で横断的に分析-

- Java スレッドからDB セッションまで追跡
  - DB処理待ちのJavaスレッドを特定
  - 発行SQLまでドリルダウン



# Oracle Application Diagnostics for Java -Java/Oracle間で横断的に分析-

- DB セッションからJava スレッドまで追跡
  - 待機状態やロック状態のDB セッションを表示
  - DB セッションを保持しているJava スレッドを特定



DBでロックが発生

このDBセッションと結びつくJavaスレッドを表示

Results

Inst	Category	SID	Serial#	Audsid	Logon	SQL Hash	Status	Waiting	DB User	DB PID
DB OSPID	OS User	Machine	Terminal	Process	Program	Thread	User	Module	Action	
1	Other	537	14	2560	05:13	2887320181	ACTIVE	enq: TM - contention	JADEUSER	16
	3861	null	demolnx01	null	1234	null	HTTPThreadGroup-5	N-A	null	null

Javaスレッドの詳細情報へリンク

JVM Active Threads

Thread	User	Request	IP Addr	OS PID	Current Call	File	Line	State	
HTTPThreadGroup-6				3750-3862	jadetest.Cart->confirmQty	Cart.java	52	Lock	Trace Thread
HTTPThreadGroup-5				3750-3822	jadetest.Inventory->updateAvailQty	Inventory.java	72	DB Wait	Trace Thread
HTTPThreadGroup-4				3750-3821	jadetest.Cart->confirmQty	Cart.java	52	Lock	Trace Thread
Thread-21				3750-3795	oracle.ons.InputBuffer->readMoreData	InputBuffer.java	275	Network Wait	Trace Thread
Thread-20				3750-3793	jamagent.jamagent->accept	Native Method		Network Wait	Trace Thread



# Oracle Application Diagnostics for Java

## -JVMヒープ分析-

- 最低限のオーバーヘッドでJVM ヒープ状況のスナップショットを取得
- 2つの異なるタイミングで取得されたJVM ヒープのスナップショットを比較し、メモリ・リーク分析

Show Heap Usage by Roots [View Summary](#)

nhomma-jp:5555 21-DEC-07 14:30 79/159 MB

nhomma-jp:5555 21-DEC-07 14:21 10/159 MB

Root	Heap 1			Heap 2			Delta	
	Objects	KB	Adj	Objects	KB	Adj	KB	Adj
<a href="#">Dictionary</a>	3324403	77,172	77,170	336685	8,770	8,768	68,403	68,403
<a href="#">Universe</a>	25	4	4	25	4	4	0	0
<a href="#">ProtectionDomains</a>	1407	232	0	1407	232	0	0	0
<a href="#">ThreadOther</a>	271	21	2	271	21	2	0	0
<a href="#">ObjectMonitor</a>	2	0	0	2	0	0	0	0
<a href="#">System Classes</a>	3324405	77,173	4	336687	8,770	4	68,403	0
<a href="#">JNI</a>	3324792	77,187	15	337074	8,785	15	68,403	0
<a href="#">System Other</a>	218	18	0	218	18	0	0	0

Heap Summary [View Roots](#)

nhomma-jp:5555 21-DEC-07 14:30 79/159 MB

nhomma-jp:5555 21-DEC-07 14:21 10/159 MB

Type	Signature	Heap 1			Heap 2			Delta			
		Objects	KB	Adj	KB	Objects	KB	Adj	KB	Adj	KB
Klass	project1/LeakExample	1	0	3313030	76,136	1	0	325312	7,734	2987718	68,403
Instance	java/util/HashSet	3	0	2759603	62,091	3	0	270005	6,298	2489598	55,793
Instance	java/util/HashMap	20	1	2759676	62,094	20	1	270078	6,301	2489598	55,793
Array	[Ljava/util/HashMap\$Entry;	22	8,194	2759727	62,096	22	1,026	270129	6,303	2489598	55,793
Instance	java/util/HashMap\$Entry	1379814	32,339	2759640	53,900	135015	3,164	270042	5,275	2489598	48,625
Instance	java/lang/Integer	1379798	21,559	1379798	21,559	134999	2,109	134999	2,109	1244799	19,450
Array	[Ljava/lang/Object;	207	1,288	553870	14,069	207	168	55750	1,459	498120	12,610
Instance	java/util/Vector	12	0	553421	14,052	12	0	55301	1,443	498120	12,610
Instance	java/lang/String	277913	6,514	555798	12,915	28853	676	57678	1,425	498120	11,490
Array	[C	277901	6,434	277901	6,434	28841	782	28841	782	249060	5,652
Klass	sun/reflect/DelegatingClassLoader	1	0	8	1	1	0	8	1	0	0
Symbol	sun/reflect/DelegatingClassLoader	1	0	1	0	1	0	1	0	0	0
Klass	sun/reflect/Reflection	1	0	23	2	1	0	23	2	0	0

# Oracle Application Diagnostics for Java

## -Javaプロファイラの実装方式-

実装方式	詳細
JVMPI/JVMTI	Java SDKで提供されるJavaVMの挙動を監視し、情報収集するためのインターフェース。詳細な情報を取れる反面オーバーヘッドがかなり大きい。
(dynamic)ByteCode Instrumentation(BCI) 又はByteCode modification	バイトコードを操作して、プロファイリングのためのコードを埋め込む方式。埋め込む量が多くなればオーバーヘッドが大きく、少なければ得られる情報が限られてしまう。
サンプリング	情報を収集したい時点のみのスナップショット(JVM、メモリヒープ)を取り分析する方式。得られる情報は限られるが、オーバーヘッドはきわめて小さい。 <b>[AD4J]</b>

- AD4Jはサンプリング方式を実装することにより  
**低オーバーヘッドを実現している**

# Oracle Application Diagnostics for Java

-優位性-

## BCIベースのツール

1. インストール、デプロイ、使用方法が難しい
  - コード・インスツルメンテーションとサーバー再起動
  - インスツルメントと診断にはエキスパートが必要
2. 高オーバーヘッド (>20%)
  - 本番環境には向かない
3. JavaからDBに至るまでの可視化はできない

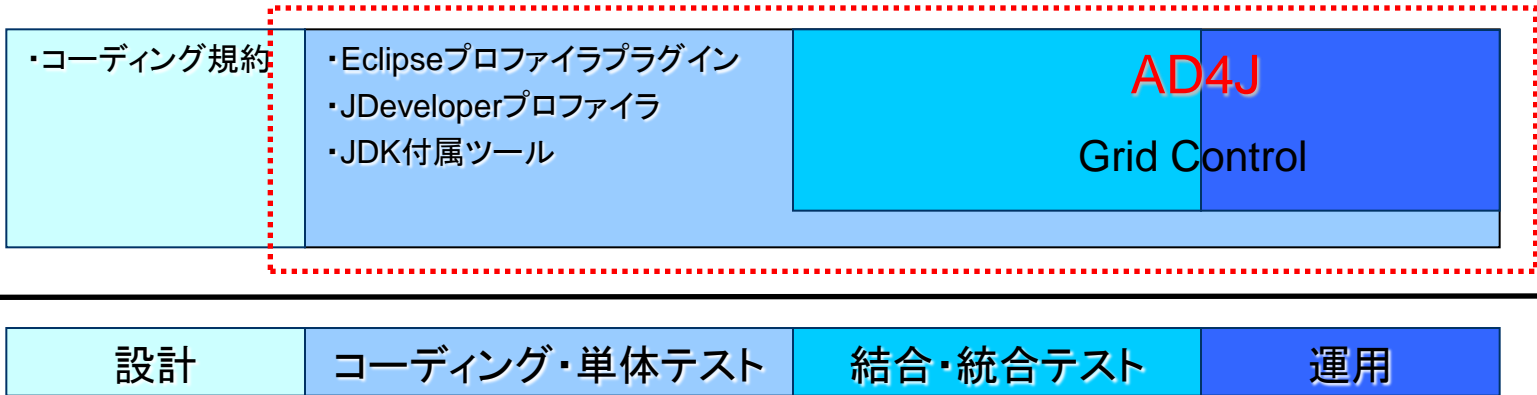
## ORACLE®

1. インストール、デプロイ、使用方法が簡単
  - コードの変更やサーバーの再起動は不要
  - アプリケーションの高度な知識が無くても、すぐに可視化可能。
2. 低オーバーヘッド (<1%)
  - 本番環境のサーバーで動かせる
3. トランザクション可視化
  - JavaからOracle DB層に渡るトランザクションを追跡

ORACLE®

# Oracle Application Diagnostics for Java

## -他のプロファイラとの役割-



- 結合・統合テスト・負荷テストフェーズ以降での利用が効果的
- Oracle Enterprise Manager 10g Grid Control (統合運用管理ツール) と組み合わせることにより、システムの一元的・包括的な運用管理が可能

# Agenda

- Webアプリケーション負荷テストの基礎知識
- テストデータの作成
- 効果的な負荷テストの方法
- ボトルネックの切り分けと分析
- まとめ
- Appendix

## 無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
  - MySQLからの移行相談
  - PostgreSQLからの移行相談
  - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
- Oracle Developer/2000 Webアップグレード相談
  - パフォーマンス・クリニック
  - Oracle Database 構成相談
- Oracle Database 高可用性診断
  - システム連携アセスメント
  - システムセキュリティ診断
  - 簡易業務診断

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

# まとめ

- 質の高いテストデータの使用
  - 負荷テストでは、本番データと同等のデータを使用
  - データのマスキングによる情報漏洩リスクの低減
- 性能テストでアプリの問題点を把握
  - シンクタイムなどを考慮し実ユーザの負荷を忠実にエミュレート
  - 負荷をかけて始めて顕在化する問題へ対処
- 性能の『見える化！』
  - 応答時間/スループット/サーバリソースをリアルタイム確認
  - ボトルネックの特定と原因の分析
  - Javaアプリの挙動やSQLの負荷などをツールにより改善

# Agenda

- Webアプリケーション負荷テストの基礎知識
- テストデータの作成
- 効果的な負荷テストの方法
- ボトルネックの切り分けと分析
- まとめ
- Appendix

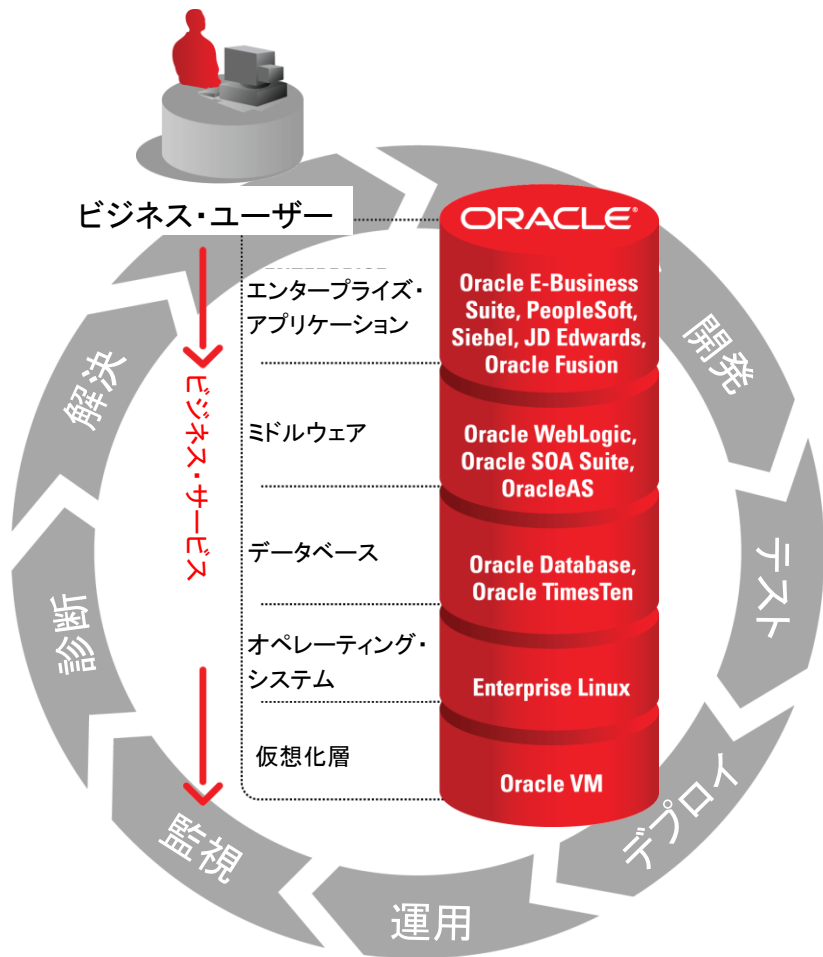
## 無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
  - MySQLからの移行相談
  - PostgreSQLからの移行相談
  - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
- Oracle Developer/2000 Webアップグレード相談
  - パフォーマンス・クリニック
  - Oracle Database 構成相談
- Oracle Database 高可用性診断
  - システム連携アセスメント
  - システムセキュリティ診断
  - 簡易業務診断

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

# オラクルの運用管理ソリューション Oracle Enterprise Manager



- プロアクティブな監視
- 問題解決のアドバイスとオペレーションの自動化
- 構成情報管理
- 効率的なテストと品質管理





# Oracle Enterprise Manager ご採用企業例



SOA インフラのデプロイが **92% 高速に**



データベースの管理にかかる時間と工数が **80% 削減**



オンラインショップでの収益損失を **25% 回避**



IT の生産性を **25% 改善**



資産の使用効率を **70% まで向上**



構成管理の工数を **90% 削減**



Oracle Enterprise Manager で **約2億円** のコスト削減



Oracle Enterprise Manager で年間 **1700 万円** のコスト削減



手作業を自動化することにより、**50% の時間** を削減



データベースのテストにかかる時間を **90% 削減**



パッチ適用にかかる工数を **75% 削減**



アプリケーションのテストにかかる時間を **1週間** 分削減



アプリケーションのテストにかかる時間を **数週間** から **数時間** に削減

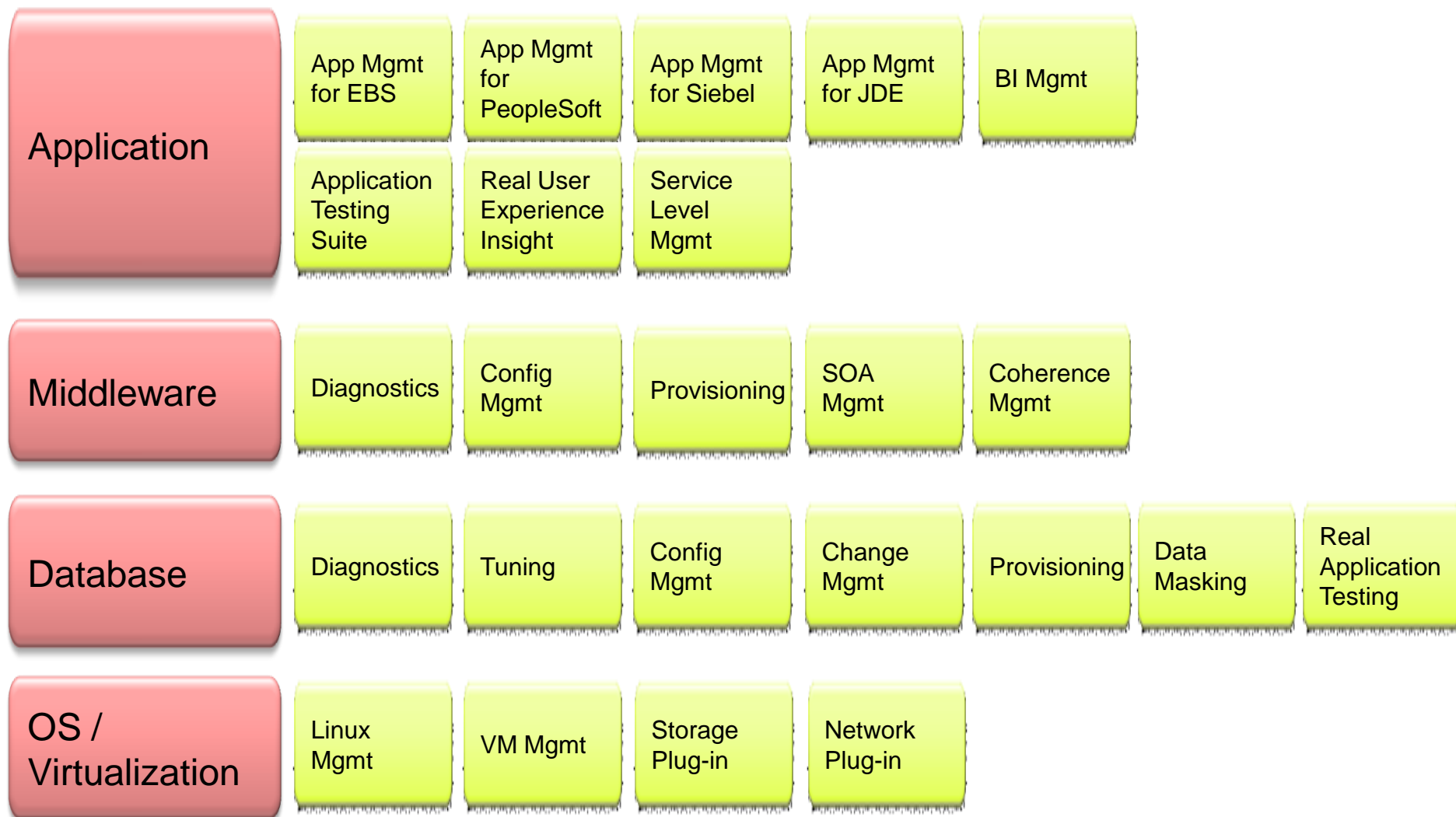


クリティカル・パッチの適用時間を **80% 削減**



Oracle Enterprise Manager により **24/7 の運用** が可能に

# Oracle Enterprise Manager 製品群



# 関連サイト

## オラクルの品質管理ソリューション ポータルサイト

<http://www.oracle.co.jp/campaign/sam/>

- ✓ 製品情報
- ✓ Webアプリケーションテスト基礎講座
  - ✓ 負荷テストを行う上での基礎知識
  - ✓ WebアプリケーションのパフォーマンスボトルネックTop25+
  - ✓ 効果的な回帰テスト
  - ✓ 効果的なテストプロセスのすすめ
- ✓ 導入事例



オラクルの品質管理は、  
ユーザー満足度の向上につながります。

ORACLE

## Oracle Wiki で技術情報を公開しています

<http://wiki.oracle.com/page/Oracle+Application+Testing+Suite>

ORACLE

Home | Discussions | Photos | Videos | Updates | To-Dos | Droplets | Members | Invite > My Profile

Search this site

EasyEdit Edit tags Email page Add a To-Do More tools (what's this?)

### Oracle Application Testing Suite

Oracle Application Testing Suiteは、ミッションクリティカルなアプリケーションをテストするための包括的なソリューションのセットを提供します。

**機能テスト: Oracle Functional Testing for Web Applications** [Datasheet]  
WebアプリケーションやWebサービスの品質を確保する最短の方法として、テストプロセスの自動化を実現する、使いやすい機能: 回帰テストツールです。

**負荷テスト: Oracle Load Testing for Web Applications** [Datasheet]  
多数のユーザからのアクセスを擬似しWebアプリケーションの性能検証を実現。運用を開始する前に、アプリケーションの性能問題をあぶりだし、Quality of Experienceの向上を支援します。

**テスト工程管理: Oracle Test Manager for Web Applications** [Datasheet]  
ソフトウェアやハードウェアの品質に関わる情報を一元管理することにより、テスト資産の活用、組織間の効果的な情報共有を実現します。  
(組み込み系のアプリケーションやハードウェアのテスト工程管理にもご利用いただけます)

**関連サイト**

- 製品情報
- 試用版のダウンロード

**イベント・セミナー情報**

- Oracle Direct Seminar (ダイゼミ) 5月21日(木) 小子数システムテスト操縦法
- 体験セミナー 5月15日(金) 負荷テスト

※詳細: 開催履歴はこちら

**【導入】**

- 動作環境
- インストールガイド

**【チュートリアル】**

- Oracle Functional Testing for Web Applications
- Oracle Load Testing for Web Applications
- Oracle Test Manager for Web Applications

**【FAQ】**

- Oracle Application Testing Suite (製品全体)
- Oracle Functional Testing for Web Applications
- Oracle Load Testing for Web Applications
- Oracle Test Manager for Web Applications

Wetpaint Wiki accounts and submissions are covered by Wetpaint.com terms of use and privacy policy. In addition, any submissions by or on behalf of Oracle are subject to the Oracle.com [Terms of Use](#). Oracle will treat any information it collects from this site in accordance with Oracle's [Your Privacy Rights](#).

- ✓ 評価版ダウンロード
- ✓ 製品情報
- ✓ 動作環境
- ✓ インストールガイド
- ✓ ライブデモ
- ✓ チュートリアル
- ✓ FAQ
- ✓ イベント・セミナー情報
- ✓ RSS

## テスト直前になって困っていませんか？ 安全でリアルなテストデータの作成術



「そろそろ負荷テストのフェーズだけど、そういえばテストデータってどうなってたっけ？」テスト直前になってからテストデータが無いことに気づくということはありませんか？

安定した品質のシステムを作るためには、本番環境相当のデータを使って本番と同じシナリオや負荷でテストをすることが欠かせません。しかし機密情報・個人情報をそのまま使用することには、通常とても大きな制約、リスクがあります。そこで無関係なデータをツールで生成したり、「あああ」に変換したりしていないでしょうか？

本セミナーでは、データのリアリティを保ちながらもセンシティブな情報をマスクするOracle Enterprise Managerのテクノロジーをご紹介します。

### Agenda

- ・Webアプリケーション負荷テストの基礎知識
- ・テストデータの作成
- ・効果的な負荷テストの方法
- ・ボトルネックの切り分けと分析

### 開催概要

- 日程 2010年2月4日(木) 13:30～14:30 (接続開始 13:45)
- 受講方法 インターネット・ライブ・セミナー
- 参加費 無料 (事前登録制)

### 詳細・お申し込み

[http://www.oracle.com/goto/jpi100204\\_1330/](http://www.oracle.com/goto/jpi100204_1330/)

# OTN×ダイセミ でスキルアップ!!



- ・技術的な内容について疑問点を解消したい！
- ・一般的なその解決方法などを知りたい！
- ・ 세미나資料など技術コンテンツがほしい！

Oracle Technology Network(OTN)を御活用下さい。

<http://otn.oracle.co.jp/forum/index.jspa?categoryID=2>

技術的な疑問点は、OTN揭示版の  
「Oracle Enterprise Manager」へ

※OTN揭示版は、基本的にOracleユーザー有志からの回答となるため100%回答があるとは限りません。  
ただ、過去の履歴を見ると、質問の大多数に関してなんらかの回答が書き込まれております。

<http://www.oracle.com/technology/global/jp/ondemand/otn-seminar/index.html>

過去のセミナー資料、動画コンテンツはOTNの  
「OTNセミナー オンデマンドコンテンツ」へ

※ダイセミ事務局にダイセミ資料を請求頂いても、お受けできない可能性がございますので予めご了承ください。  
ダイセミ資料はOTNコンテンツ オン デマンドか、セミナー実施時間内にダウンロード頂くようお願い致します。

ORACLE

# OTNセミナー オンデマンド コンテンツ

期間限定にて、ダイセミの人気セミナーを動画配信中!!

ダイセミのライブ感はそのままに、お好きな時間で受講頂けます。

最新のコンテンツ

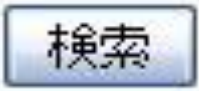
エンジニアのためのITIL実践術 再生時間: 60分	ここからはじめよう Oracle PL/SQL入門 再生時間: 60分	実践!!高可用システム構築 -RAC基本 再生時間: 60分	お悩み解決! Oracle のサイジング 再生時間: 60分

Database

今さら聞けない!!バックアップ-リカバリ入 再生時間: 60分	意外と簡単!?! Oracle Database 11g -セ 再生時間: 60分	実践!!バックアップ-リカバリ 再生時間: 60分	意外と簡単!?! Oracle Database 11g -デ 再生時間: 60分

>> もっと見る

OTN オンデマンド



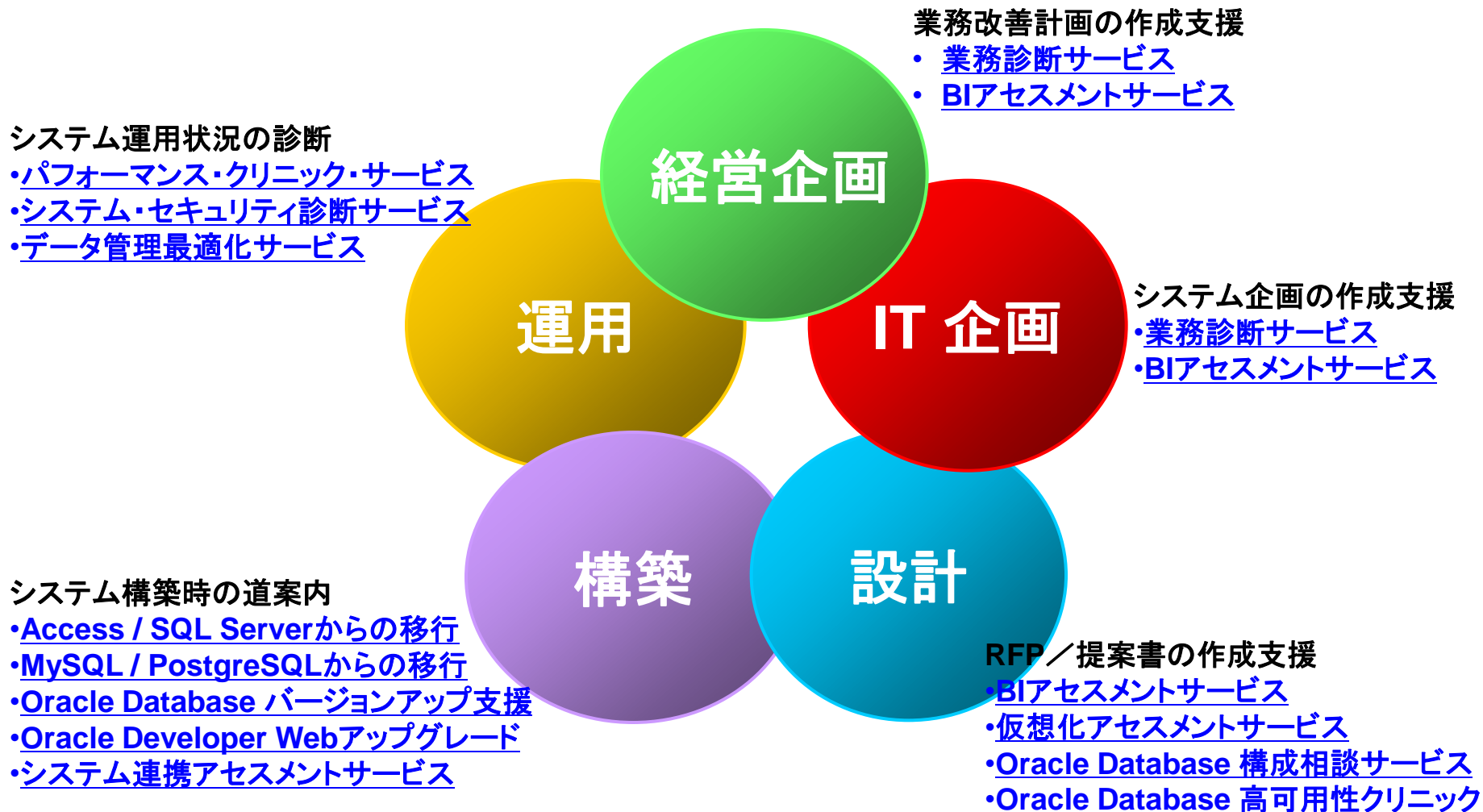
※掲載のコンテンツ内容は予告なく変更になる可能性があります。

期間限定での配信コンテンツも含まれております。お早めにダウンロード頂くことをお勧めいたします。



# ITプロジェクト全般に渡る無償支援サービス

## Oracle Direct Conciergeサービスメニュー





あなたにいちばん近いオラクル



# Oracle Direct

まずはお問合せください

システムの検討・構築から運用まで、ITプロジェクト全般の相談窓口としてご支援いたします。

システム構成やライセンス/購入方法などお気軽にお問い合わせ下さい。

## Web問い合わせフォーム

専用お問い合わせフォームにてご相談内容を承ります。

[http://www.oracle.co.jp/inq\\_pl/INQUIRY/quest?rid=28](http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28)

※フォームの入力には、Oracle Direct Seminar申込時と同じ  
ログインが必要となります。

※こちらから詳細確認のお電話を差し上げる場合がありますので、ご登録されている連絡先が最新のものになっているか、ご確認下さい。

## フリーダイヤル

**0120-155-096**

※月曜～金曜 9:00～12:00、13:00～18:00

(祝日および年末年始除く)

ORACLE®

# ORACLE®

以上の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

Oracle、PeopleSoft、JD Edwards、及びSiebell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。その他の名称はそれぞれの会社の商標の可能性がります。