

# Oracle DBA & Developer Days 2011

日本オラクル、今年最大の技術トレーニングイベント

2011年11月9日(水)～11月11日(金) シェラトン都ホテル東京



## ORACLE®

どこまでチューニングできるのか？  
最新Oracle Database 高速化手法

日本オラクル株式会社 製品事業統括 テクノロジー製品事業統括本部  
プリンシパルエンジニア 柴田 長

以下の事項は、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とJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

# Agenda

- はじめに
- OLTP向けチューニング手法
  1. Oracle Load Testingを使用したデータベースの負荷テスト
  2. Oracle Enterprise Managerで索引チューニング
  3. Database Smart Flash Cacheの効果
- DWH向けチューニング手法
  1. Real-Time SQL Monitoring(SQL監視)で現状分析
  2. Partition Advisorで表のパーティション化
  3. Parallel Queryの適用
  4. Oracle Advanced Compressionで表を圧縮
  5. In-Memory Parallel Executionの効果

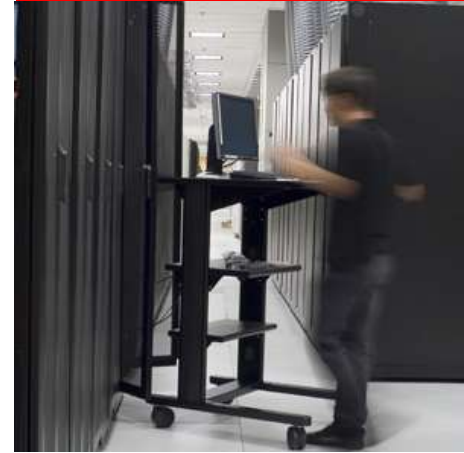


はじめに

# はじめに

## データベースのチューニングの考え方

- チューニングとは、「ボトルネックとなっている箇所を取り除き、H/W性能を限界まで引き出す」作業を意味しています。
  - まずは、ボトルネックの発生箇所(原因)を特定
  - 次に、原因を改善する策として、以下の作業を実施
    - 無駄な処理を削り取ることで、消費コストを減少
    - ボトルネック箇所のH/Wを追加し、全H/Wリソースの消費バランスを整える
- 1つ目のボトルネックを改善すれば、別の箇所がボトルネックとなり得る
  - ボトルネックは移動するものであり、一般的にはユーザーの求めるサービス・レベルに達するか、CPUリソースがボトルネックになるまで繰り返し実施
- 本セッションでは、Oracle Database 11gが提供する機能を活用し、如何にシンプルで効果的なチューニングを実施できるのかを、ご紹介します。



## OLTP向けチューニング手法

1. 容易な負荷テスト手法
2. SQLの索引チューニング
3. キャッシュの有効活用

# OLTP向けチューニング手法

## 設定シナリオと到達目標

- 設定シナリオ

- 貴方は、Webショッピングサイトのデータベース管理者です。
  - 世間が空前の〇×ブームとなり、ユーザーのオンライン・アクセスが急増
  - そんなある日、ユーザーから画面操作に時間がかかるとクレームの入電
  - 早急に原因を特定し、チューニングするように上司命令を受けました。。
    - ※ データベース環境および、負荷生成ツール「Oracle Load Testing」の準備は完了しており、直ぐにでも負荷テストを開始できる状況とします。
    - ※ データベース・サーバーには、物理メモリを限界まで搭載済みと仮定します。

- 到達目標

- アプリケーション側に手を加えることなく、データベースのスループット (Transaction Per Sec) を約100倍向上させる！



# 1. Oracle Load Testingを使用した データベースの負荷テスト



# システム品質の課題

## Oracle Application Testing Suiteで解決

課題	原因	Oracleのソリューション
応答時間が遅くユーザからのクレームが多い	性能に関する非機能要件での取り決めが甘く(あるいは実施していない)、多数のアクセスにシステムが耐えられるかテストしていない	<b>Oracle Load Testing</b> WebアプリケーションやWebサービス/SOAベース・アプリケーション、 <b>Database</b> の性能と拡張性を、性格かつ容易にテストできる負荷テストツールです。テスト方法論などのセミナーも実施しています
キャンペーンの告知をしたらサイトがダウンした		
高負荷時に他人のデータが表示されてしまった	HTTP通信レベルのテストのみでコンテンツまで確認していない	
性能検証を行いたい方法が分からない	性能に関するテストの経験がない	
入力値が違うだけのテストが多く時間がかかる	入力値やプラットフォームが違うだけのテストを手で実施しては効率が悪く、テスト漏れも誘発	<b>Oracle Functional Testing</b> 機能およびリグレッション・テストを、迅速かつ正確に実行する自動テストツールです
パッケージアプリなので環境ごとのテストが必要		
機能変更に対してテスト漏れはないだろうか	要件に対するテストの適用範囲や他の機能に対する影響度合いが可視化されていないため、単一機能についてしかテストされていない	<b>Oracle Test Manager</b> テスト工程全体を構築・体系化する、柔軟で操作が容易なテスト工程管理ソリューションです。品質に関する情報を一元管理することでテスト資産を活用し、品質状況の可視化と組織間の効果的な情報共有を実現します ※C/S, 組み込み等、Webアプリケーション以外でも適用可能です
不具合を修正したがデグレードしてしまった		
毎週の進捗会議の資料を作るのが大変	Excelなどオフラインでのテスト管理となっているため、データの集計に時間を要してしまい、正確性にかけてしまう時もある	
不具合の修正状況が分からない	メールだけでは対応が遅れ、オフショア開発では時差の影響も受ける	
開発拠点が分散していて連絡が大変		

# Oracle Application Testing Suite 9.2

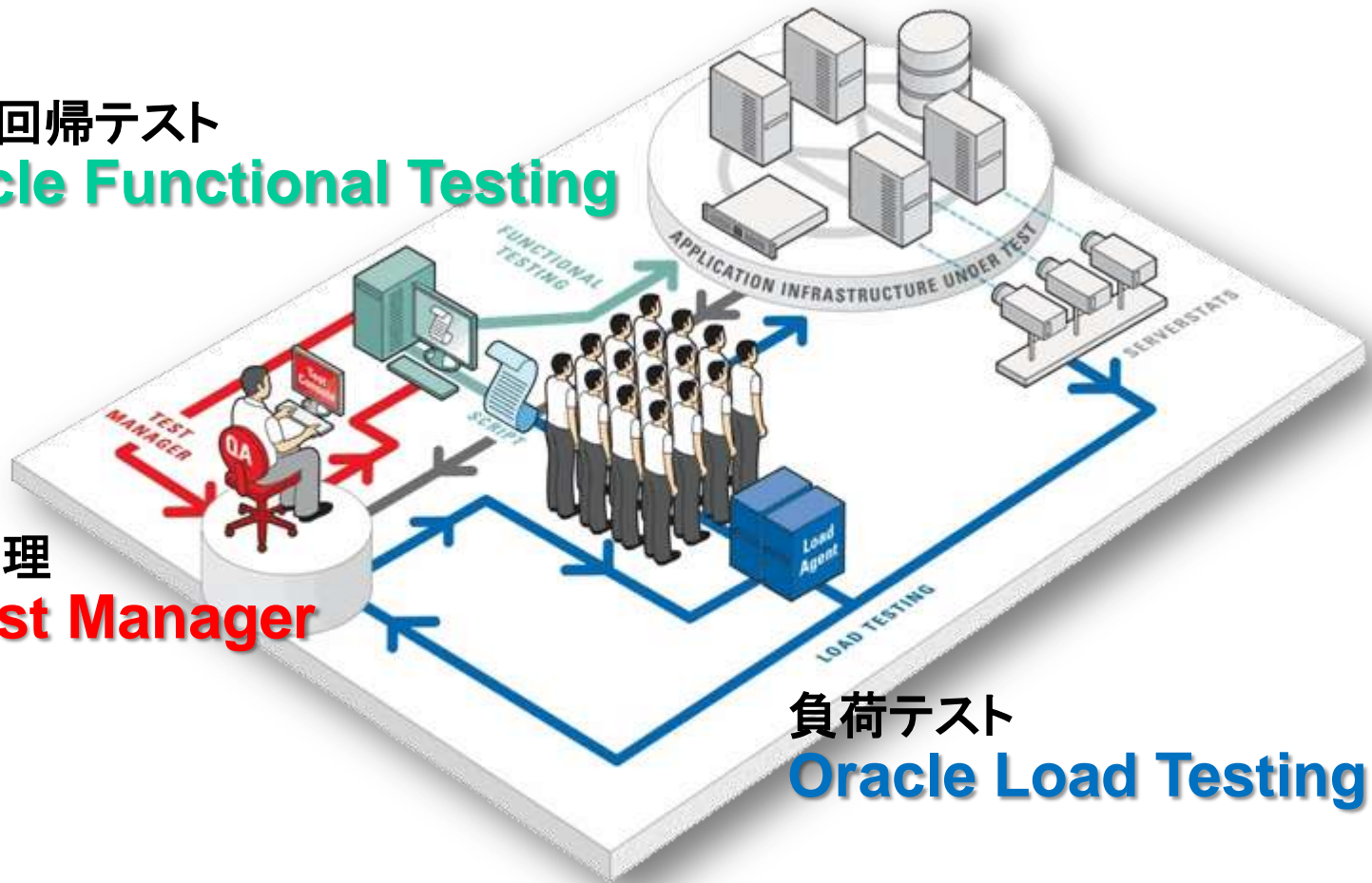
ユーザー視点のテストを簡単かつ迅速に実現する製品群

機能/回帰テスト

Oracle Functional Testing

テスト工程管理

Oracle Test Manager

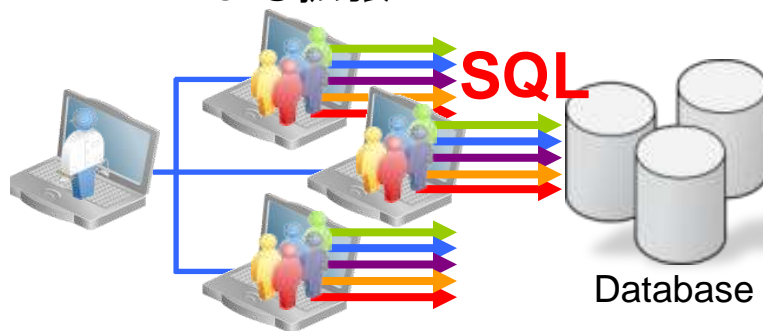


負荷テスト

Oracle Load Testing

# Oracle Load Testing Accelerator for Database

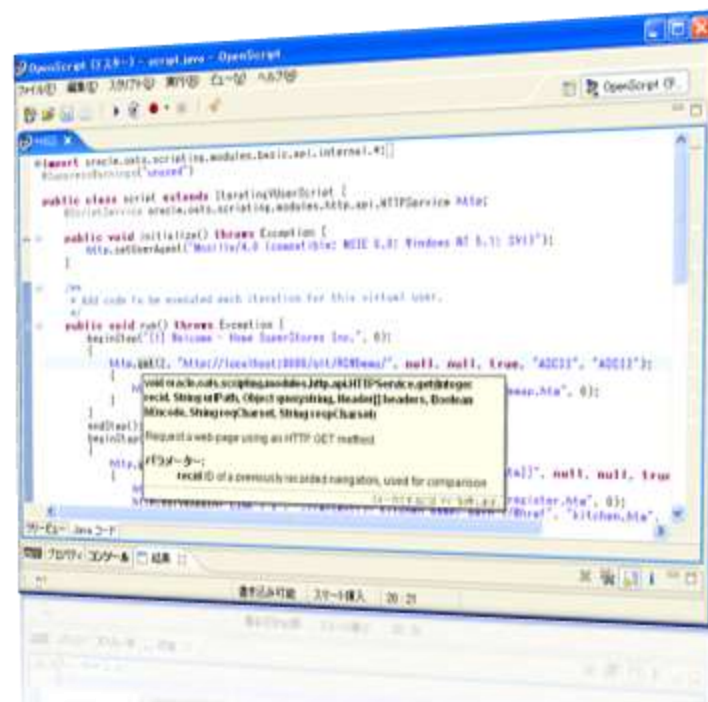
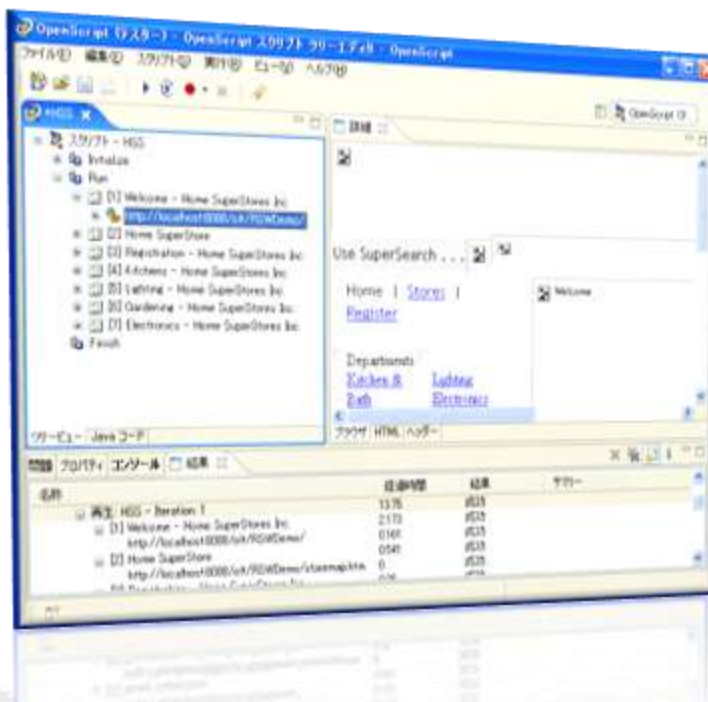
- データベースに対する直接の負荷テストをサポート
  - データベースへの接続方式
    - Oracle Thin (oracle.jdbc.driver.OracleDriver)
    - ODBC (sun.jdbc.odbc.JdbcOdbcDriver)
  - 生成可能な負荷
    - Query、DML、DDLの実行
    - PL/SQLの実行
    - SQL行数カウントテスト
    - Java APIによる拡張



# OpenScript

## テスト・スクリプトの迅速・容易な作成をサポート

- Eclipse IDEをベースとした柔軟なスクリプト作成を可能とする統合開発環境
  - グラフィカルなツリー・ビュー・インタフェース
  - プログラミングを行うJavaコード・ビュー・インタフェース

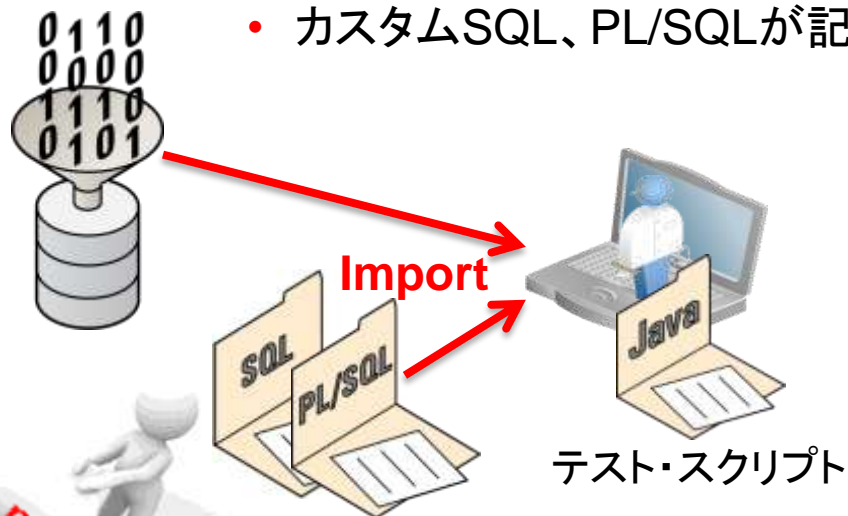




とは言っても、実際にSQLを用意したり作成するのは・・・

## そんな時は、**データベース ファイル インポート**

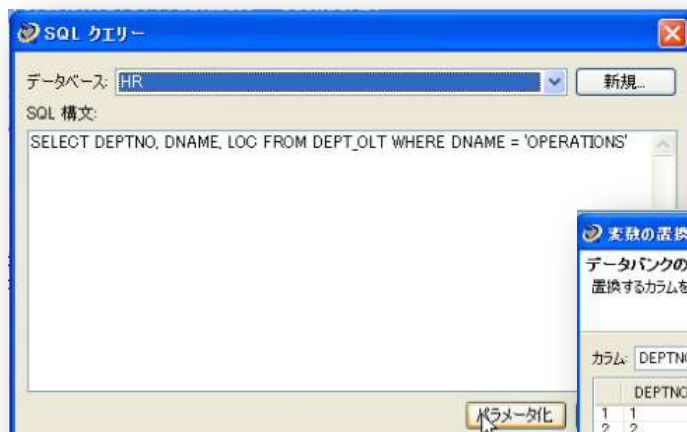
- Open Scriptでは、SQLをテスト・スクリプトへインポートすることが可能
  - データベース・リプレイ キャプチャ ファイル
    - Oracle Real Application Testingのデータベース・リプレイでキャプチャされたトランザクションSQL
    - 例えば、本番システムで実行されているSQLをスクリプト化する場合
  - SQL および PL/SQL 構文スクリプト
    - カスタムSQL、PL/SQLが記載されたテキストファイル



SQLのWhere句の条件をユーザー毎に変更したい

もちろん、**データバンク(CSVデータ)**を使用可

- データベースの負荷テストでは、扱うデータの範囲(Where句の条件の種類)によって性能が大きく異なる
  - 例えば、1種類に固定(where EMPNO=1000)した場合
    - キャッシュ・ヒット率が異常に高い傾向にあり、本番想定とは言えない
    - データバンクを使用することで解決

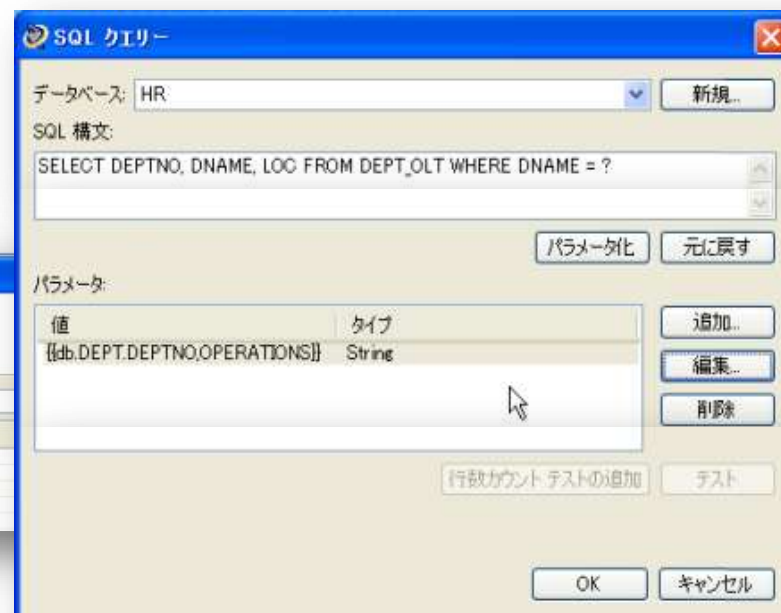


表の置換

データバンクの追加  
置換するカラムを選択してください

カラム: DEPTNO

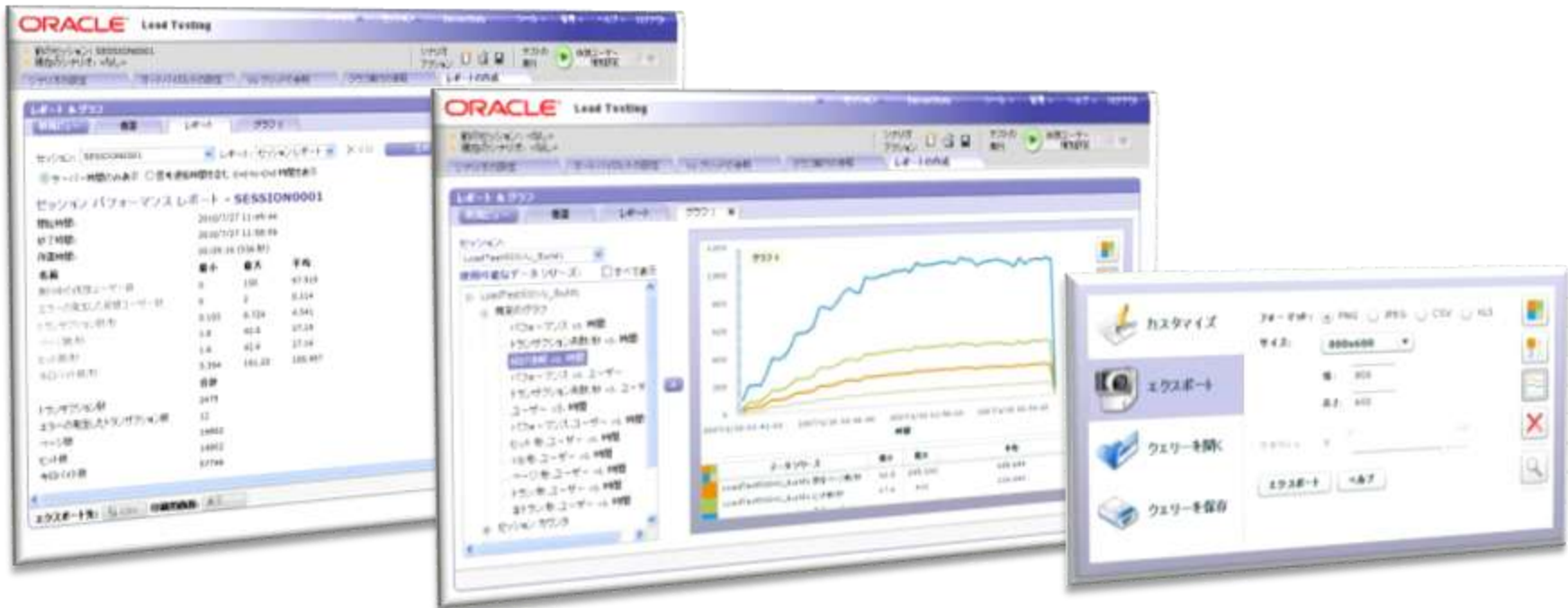
	DEPTNO	DNAME	LOC
1	1	OPERATIONS	BOSTON
2	2	ACCOUNTING	NEW YORK
3	3	RESEARCH	DALLAS
4	4	SALES	CHICAGO



# Oracle Load Testing

## 充実したレポート機能

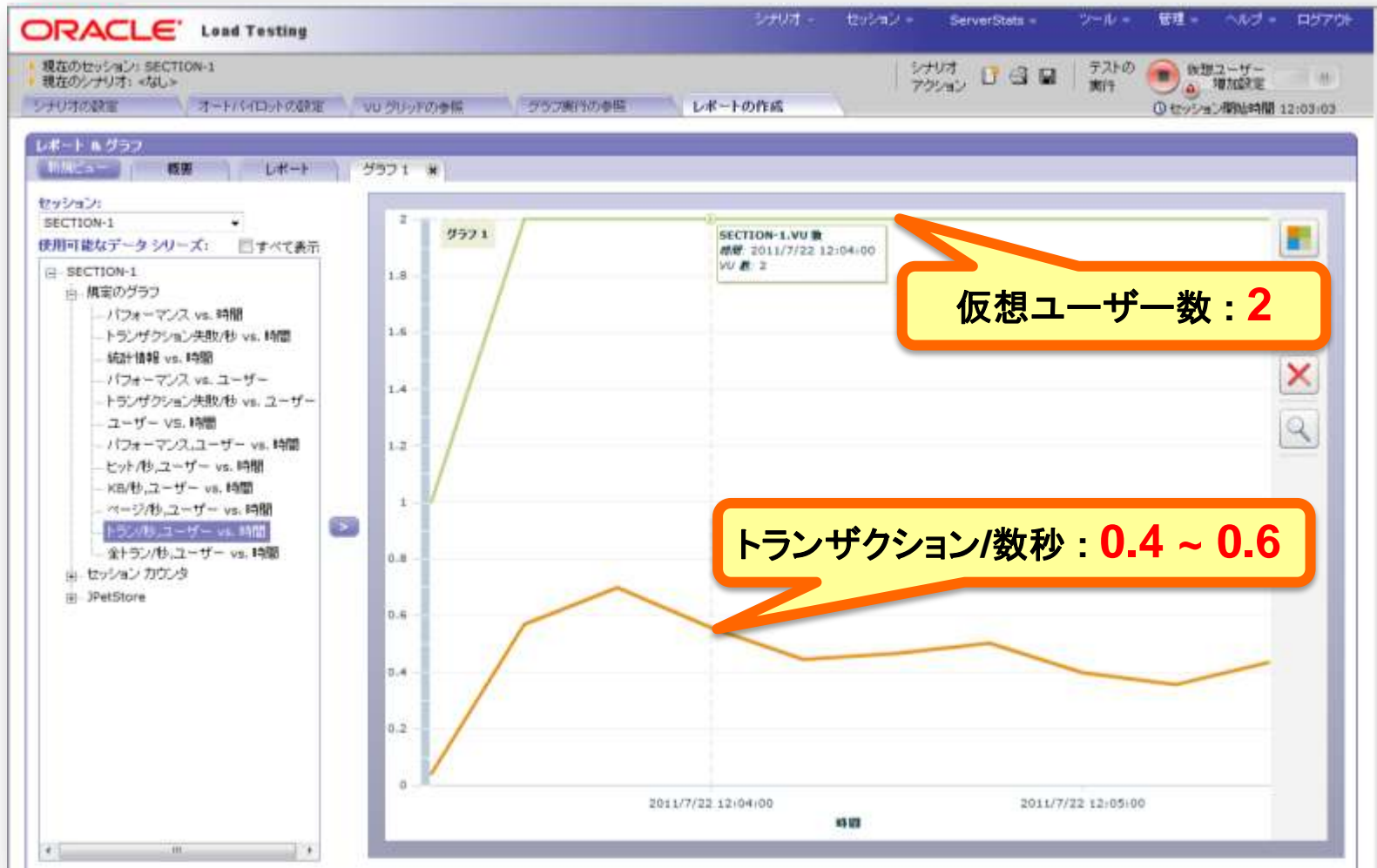
- 必要な結果データのグラフを瞬時に作成することが可能
  - 負荷テスト中にも参照可能で、オンデマンドでのチューニングが可能
  - 複数の条件の異なるテスト結果を1つのグラフとして表現することも可能
  - グラフは画像ファイルやCSVファイルへの出力が可能



ORACLE

# 本シナリオにおける問題発生時のスループット

## Oracle Load Testing - 負荷テスト結果1







## 2. Oracle Enterprise Managerで SQLの索引チューニング

# Oracle Enterprise Manager

## データベース・インスタンスの負荷状況を確認

- 「パフォーマンス」タブを選択し、「平均アクティブ・セッション」のグラフを確認
  - 「User I/O」が大部分を占めていることから、I/Oボトルネックであることが判明



CPU使用率は10%程度



# 【参考】 User I/O関連の主な待機イベント

- **db file sequential read**
  - バッファ・キャッシュを経由したシングル・ブロック単位での読み込み時に発生する待機イベント。
  - 主に、索引を使用して表データへのアクセスする際に発生。キャッシュ・ヒット率が悪い場合に、多発する傾向。
- **db file scattered read**
  - バッファ・キャッシュを経由したマルチ・ブロック単位での読み込み時に発生する待機イベント。
  - 主に、Table Full Scan が行われる際に発生する為、適切に索引が作成されていない/使用されていない場合の判断として使用する。
  - また、Pre-Warming機能が有効な場合や、索引のリーフブロックを全て読み込む処理 (Index Full ScanやIndex Fast Full Scan) 時にも発生する。
- **direct path read**
  - バッファ・キャッシュを経由しない、マルチ・ブロック単位での読み込み時に発生する待機イベント。主に、パラレルでTable Full Scanを実行時に発生する。
  - ただし、Oracle 11g 以降のバージョンでは、シリアルでTable Full Scanを行った場合にも、バッファ・キャッシュのサイズに対し表のサイズが大きい際に使用されることがあります。

# 待機イベントの原因を確認

## 問題となっているSQLの実行計画の参照

- PRODUCT表に対する「TABLE ACCESS FULL(全表検索)」を確認  
→ 索引が「存在しない」or「存在するが使用されていない」と予想

ORACLE Enterprise Manager 11g Database Control

データベース・インスタンス: orcl > トップ・アクティビティ > SQLの詳細: duwz811xc2jgv

SQL IDに切替え  **実行** データの表示

実行時間: 手動リフレッシュ **リフレッシュ** **SQLワークシート** **SQLチューニング・アドバイザのスケジュール** **SQL修復アドバイザー**

テキスト

```
select /*+ SQL2-0 */ productid, name, descr, category
from product where(lower(name) like :1)
```

詳細

すべて開く | すべて閉じる

操作	オブジェクト	順序	行	バイト	コスト	CPU (%)	問合せブロック名/オブジェクトの別名	フィルタ	予測
SELECT STATEMENT		1			3,444	100			
TABLE ACCESS FULL	PRODUCT	1	25,625	4,203M	3,444	0.0:0.42	SEL\$1 / PRODUCT@SEL\$1	LOWER("NAME") LIKE :1	"PRODUCTID" [VARCHAR2,40], "CAT...

解決策が解らなくても心配なし！

# SQLチューニング・アドバイザの実行

- 「SQLチューニング・アドバイザのスケジュール」ボタンを押下し、このSQLに対する推奨アドバイスを要求



# SQLチューニング・アドバイザー 推奨の確認

- ファンクション索引(LOWER(NAME)の生成が推奨
- 次に、「実行計画の比較」で、推奨の適用効果を確認

ORACLE Enterprise Manager 11g Database Control

設定 プリファレンス ヘルプ ログアウト  
データベース

データベース・インスタンス: orcl > アドバイザ・セントラル > SQLチューニング・タスク:SQL\_TUNING\_1311304135988 > SYSとしてログイン

SQL IDの推奨:duwz811xc2jgv

戻る

推奨される実装が1つ見つかっています。

SQLテキスト

```
select /*+ SQL2=0 */ productid, name, descr, category from product where(lower(name) like :1)
```

推奨の選択

元の実行計画(注釈付き)

実装

選択	タイプ	結果	推奨	論理	ベネフィット(%)	その他統計	新規実行計画	実行計画の比較
<input checked="" type="radio"/>	索引	索引を1つ以上作成すると、この文の実行計画を改善できます。	物理スキーマ設計を改善するAccess Advisorの実行か、推奨される索引の作成を検討してください。 JPETSTORE.PRODUCT(LOWER("NAME"))	推奨される索引を作成すると、この文の実行計画が大きく改善されます。ただし、単一の文ではなく代理SQLワークロードを使用した“ Access Advisor”の実行が適切な場合もあります。この処理により、索引メンテナンス・オーバーヘッドおよび追加領域消費が考慮された包括的な索引推奨事項を取得できます。	99.68		oo	oo

物理スキーマ設計を改善するAccess Advisorの実行か、推奨される索引の作成を検討してください

JPETSTORE.PRODUCT(LOWER("NAME"))

# SQLチューニング・アドバイザー

## 推奨適用前後の実行計画を比較

- ファンクション索引を実装する前に、その効果（SQLの実行コストが大幅に削減されること）を確認可能

ORACLE Enterprise Manager 11g  
Database Control

設定 プリファレンス ヘルプ ログアウト

データベース

データベース・インスタンス: orcl > アドバイザ・セントラル > SQLチューニング・タスク: SQL\_TUNING\_1311304135988 > SQL IDの推奨: duwz811xc2jgv >

SYSとしてログイン

### 実行計画の比較

#### 元の実行計画(注釈付き)

計画ハッシュ値 427209646

すべて開く | すべて閉じる

操作	ラインID	オブジェクト	オブジェクト・タイプ	順序	行	バイト	コスト	時間	CPUコスト	I/Oコスト
SELECT STATEMENT	0			2		4,304,199	3,444	42	290,509,312	3,429
TABLE ACCESS FULL	1	JPETSTORE.PRODUCT	TABLE	1		4,304,199	3,444	42	290,509,312	3,429

# Before

#### 索引のある新しい実行計画

計画ハッシュ値 2463378923

すべて開く | すべて閉じる

操作	ラインID	オブジェクト	オブジェクト・タイプ	順序	行	バイト	コスト	時間	CPUコスト	I/Oコスト
SELECT STATEMENT	0			3		3,023	11	1	85,716	11
TABLE ACCESS BY INDEX ROWID	1	JPETSTORE.PRODUCT	TABLE	2		3,023	11	1	85,716	11
INDEX RANGE SCAN	2	IDX\$\$\$_00210001	INDEX	1			3	1	24,964	3

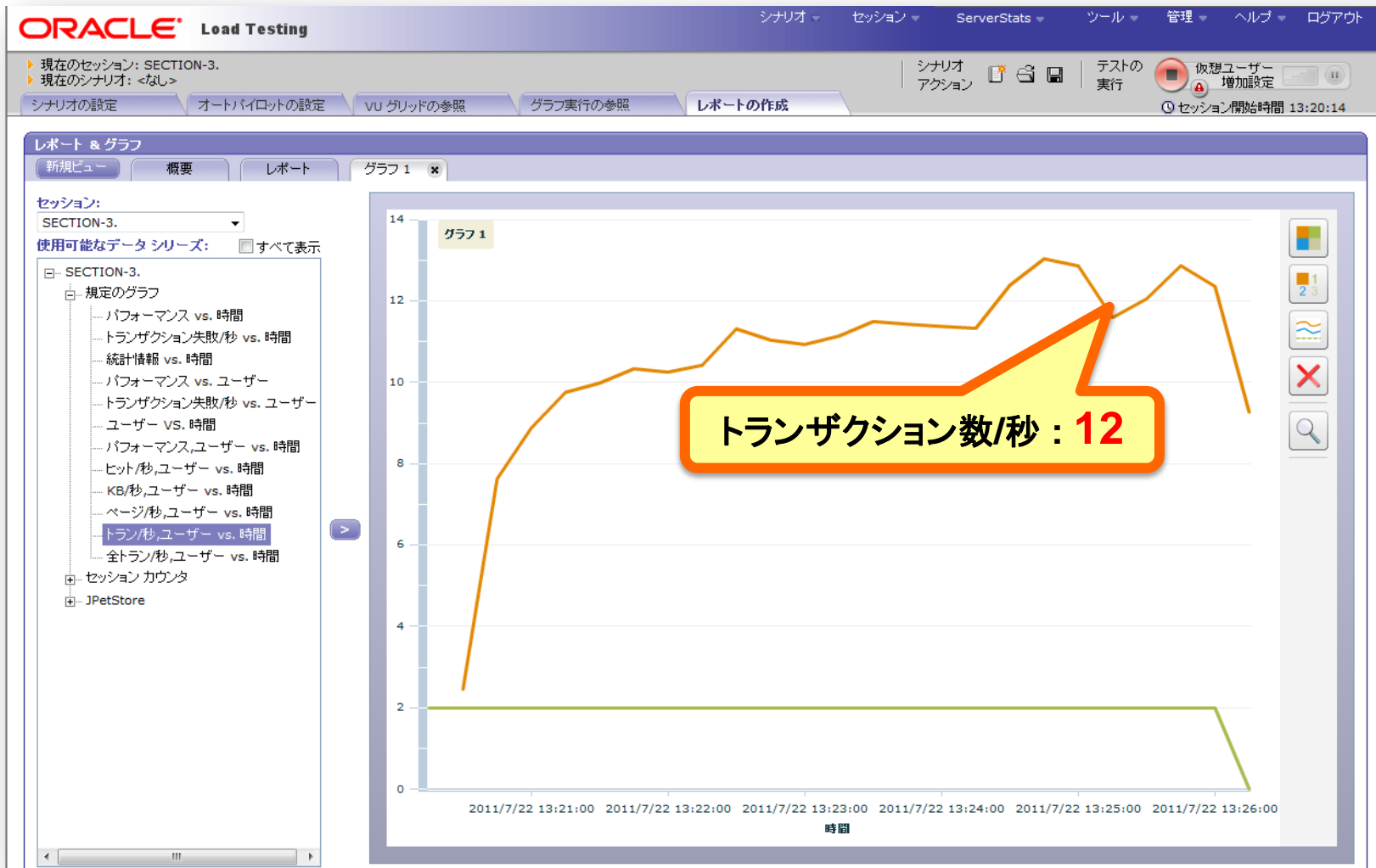
# After

ORACLE



# 索引チューニングの効果を確認

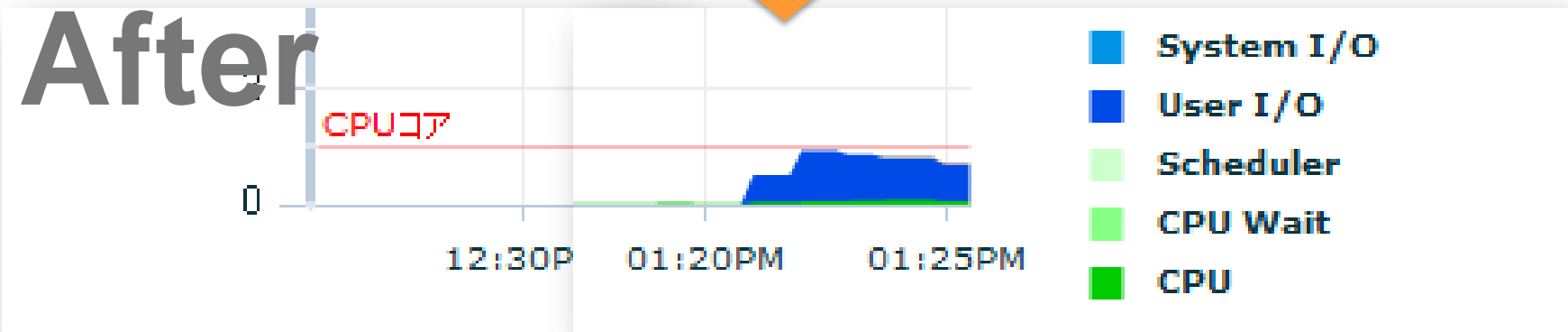
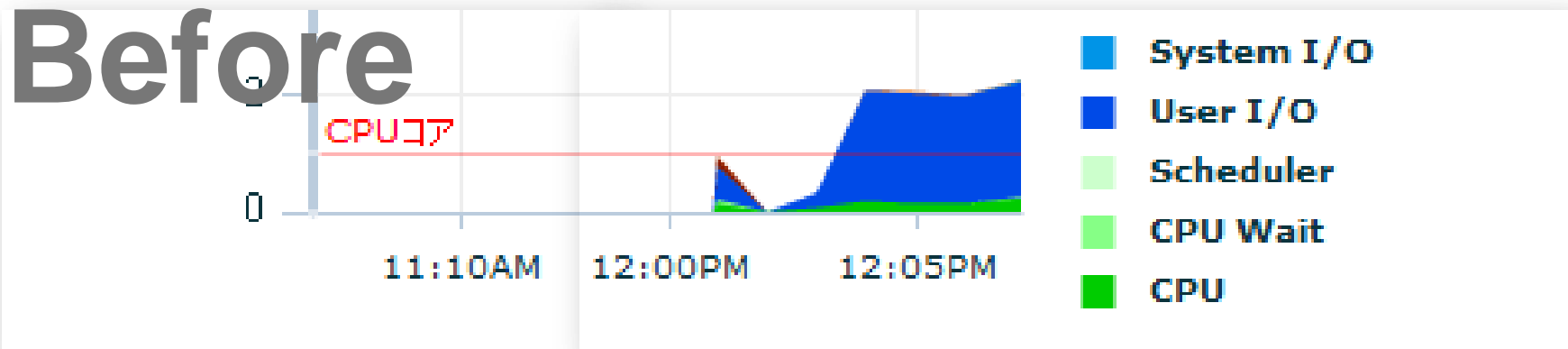
## Oracle Load Testing - 負荷テスト結果2



# Oracle Enterprise Manager

## データベース・インスタンスの負荷状況を確認

- 索引チューニング前に高い割合であった「User I/O」が大幅に減少



# Oracle Load Testing

## 実行中の負荷テストの仮想ユーザ数を増加

- 実行中の負荷テストを停止することなく、仮想ユーザー数を「15」に増加
  - 負荷生成中のOracle Load Testingの画面において、「シナリオの設定」タブを選択し、VU数に「13」を設定し、「オートパイロットへ追加」ボタンを押下

ORACLE Load Testing

シナリオ ▼ セッション ▼ ServerStats ▼ ツール ▼ 管理 ▼ ヘルプ ▼ ログアウト

前のセッション: <なし>  
現在のシナリオ: <なし>

シナリオアクション | テストの実行 | 仮想ユーザー増加設定

シナリオの設定 | オートパイロットの設定 | VUグリッドの参照 | グラフ実行の参照 | レポートの作成

スクリプトの選択

パス: /Default/JPetStore/JPetStore.jwg  
スクリプトのみ

Default  
JPetStore

シナリオに追加

シナリオのパラメータを設定

スクリプト: JPetStore  
システム: OLT Server

VU数

VU数を「13」に設定

オートパイロットへ追加を押下

オートパイロットへ追加

# VU数増加前後のスループットを確認

## Oracle Load Testing - 負荷テスト結果3

- VU数が15へ増加し、スループットが上昇することを確認  
→ 索引チューニングで、より多くのユーザー・リクエストの処理が可能



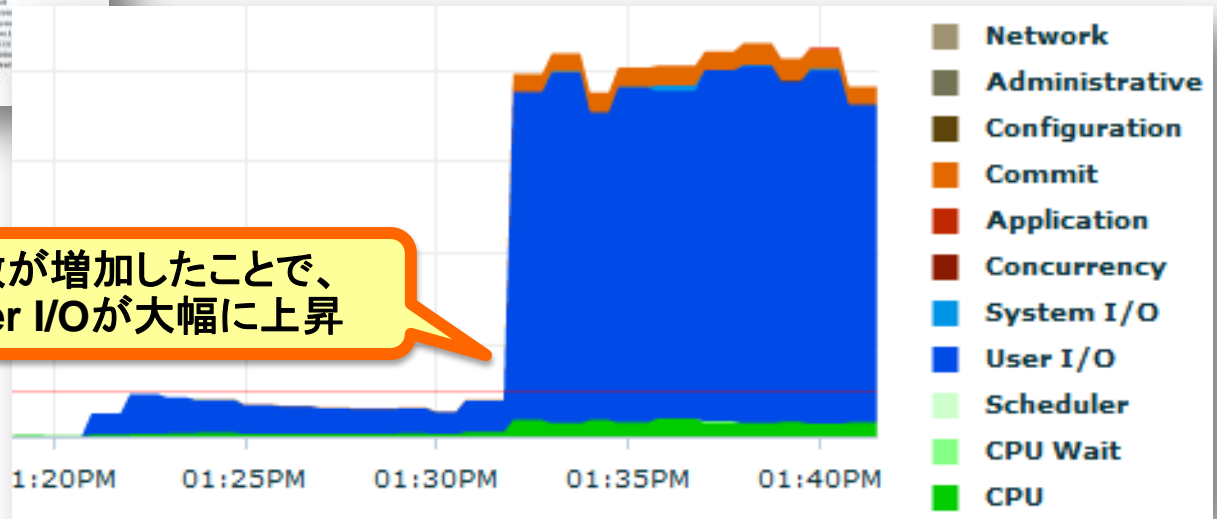
# Oracle Enterprise Manager

## VU追加後のデータベース・インスタンスの負荷状況を確認

- 再び、「User I/O」の割合が大幅に上昇していることを確認



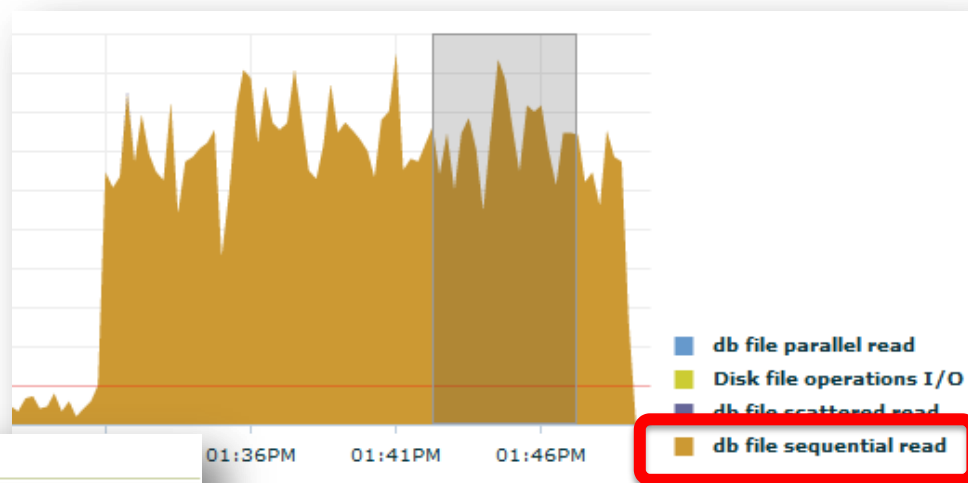
VU数が増加したことで、  
User I/Oが大幅に上昇



# I/Oが頻発しているSQLの特定

## SQL毎の待機イベントを比較

- SQL ID「3z2ujuthu61u」において、大部分の待機イベント「db file sequential read」が発生していることが確認



上位SQL: User I/O

アクション: SQLチューニング・アドバイザのスケジュール

すべて選択 | 選択解除

選択	アクティビティ (%)	SQL ID	SQLタイプ
<input type="checkbox"/>	89.58	3z2ujuthu61u	SELECT
<input type="checkbox"/>	10.10	duwz811xc2jgv	SELECT
<input type="checkbox"/>	14	2xkgh8av20s04	INSERT





### 3. Database Smart Flash Cacheの効果

Oracle Enterprise ManagerのSQLチューニング・アドバイザの推奨が無いことから、SQLをチューニングすることでの性能改善ができないことが確認できました。しかし、目標のスループットには到達していない状況です。

I/O性能不足が問題であることから、メモリー・アドバイザの確認とOracle Database 11g Release 2の新機能を活用したチューニングを実施します。

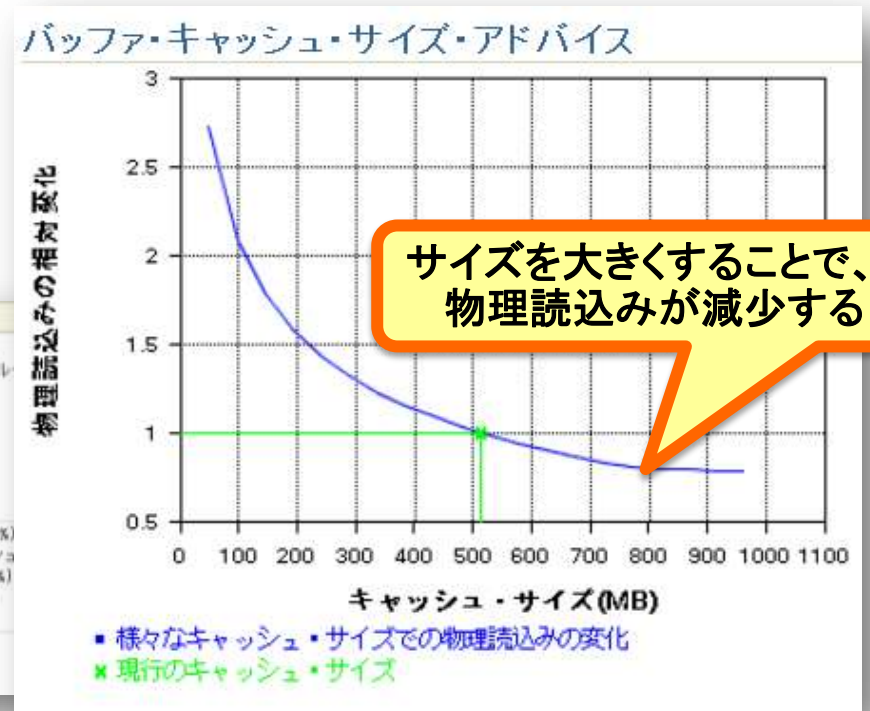


# Oracle Enterprise Manager

## バッファ・キャッシュ・サイズ・アドバイス

- 「サーバ」→「メモリー・アドバイザ」→「バッファ・キャッシュ・アドバイス」
  - ※ RAC環境の場合は、インスタンスを選択済みの場合にリンクが表示される  
選択していない場合は、セントラル・アドバイザ内にリンクが表示される

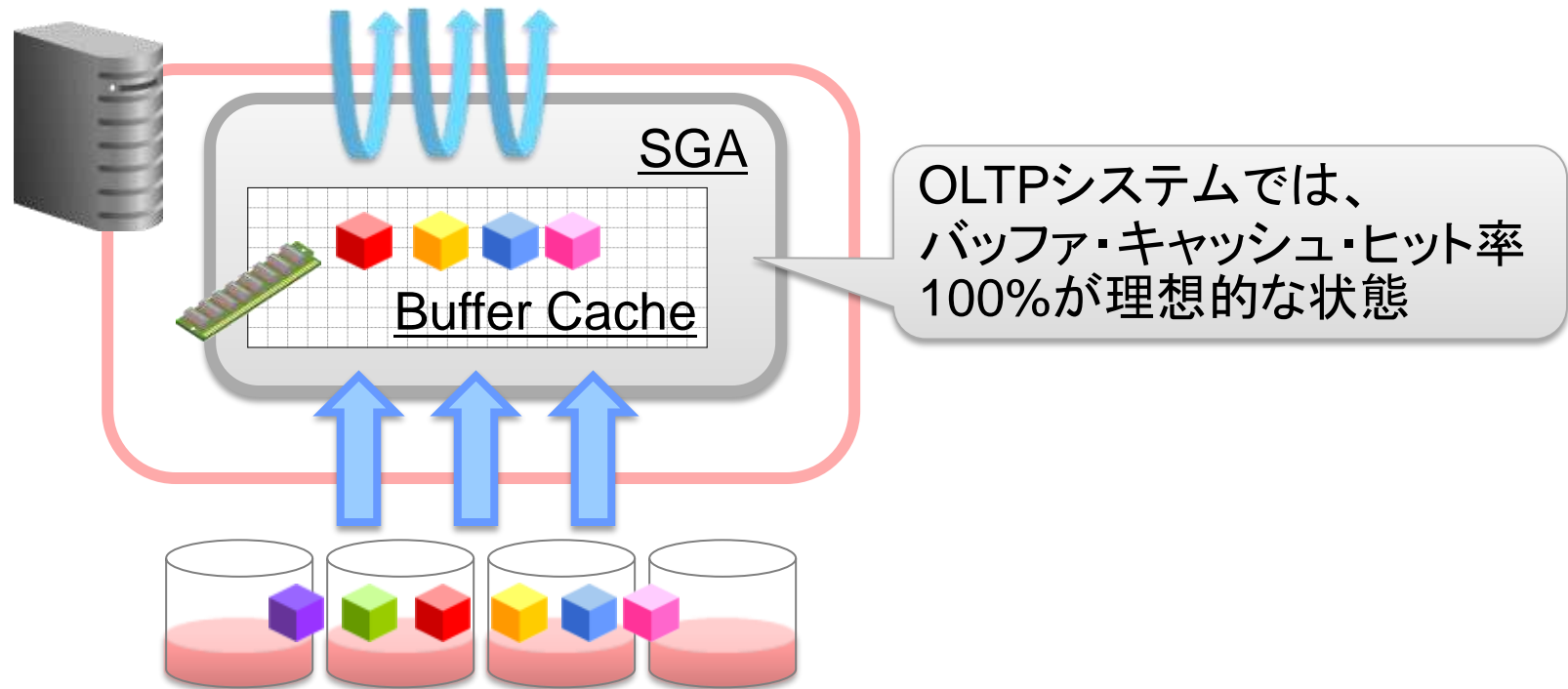
The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The navigation menu includes 'ホーム', 'パフォーマンス', '可用性', 'サーバ', 'スキーマ', and 'データ移動'. The 'サーバ' menu is expanded, showing 'メモリー・アドバイザ' (Memory Advisor) highlighted with a red box. Below this, the 'バッファ・キャッシュ・サイズ・アドバイス' (Buffer Cache Size Advisor) link is also highlighted with a red box. The main content area shows the '自動共有メモリー管理' (Automatic Shared Memory Management) section, with '無効' (Disabled) selected. A table lists memory pools: '共有プール' (512 MB), 'バッファ・キャッシュ' (512 MB), 'ラージ・プール' (0 MB), and 'Javaプール' (0 MB). A pie chart shows the memory usage: '共有プール (49.4%)', 'バッファ・キャッシュ (49%)', 'ラージ・プール (0%)', 'Javaプール (0%)', and 'その他 (1.1%)'. The 'SGA合計 (MB)' is 1035. A '計算' (Calculate) button is visible at the bottom.



# データベースのOLTP処理の基本動作

## データのキャッシング

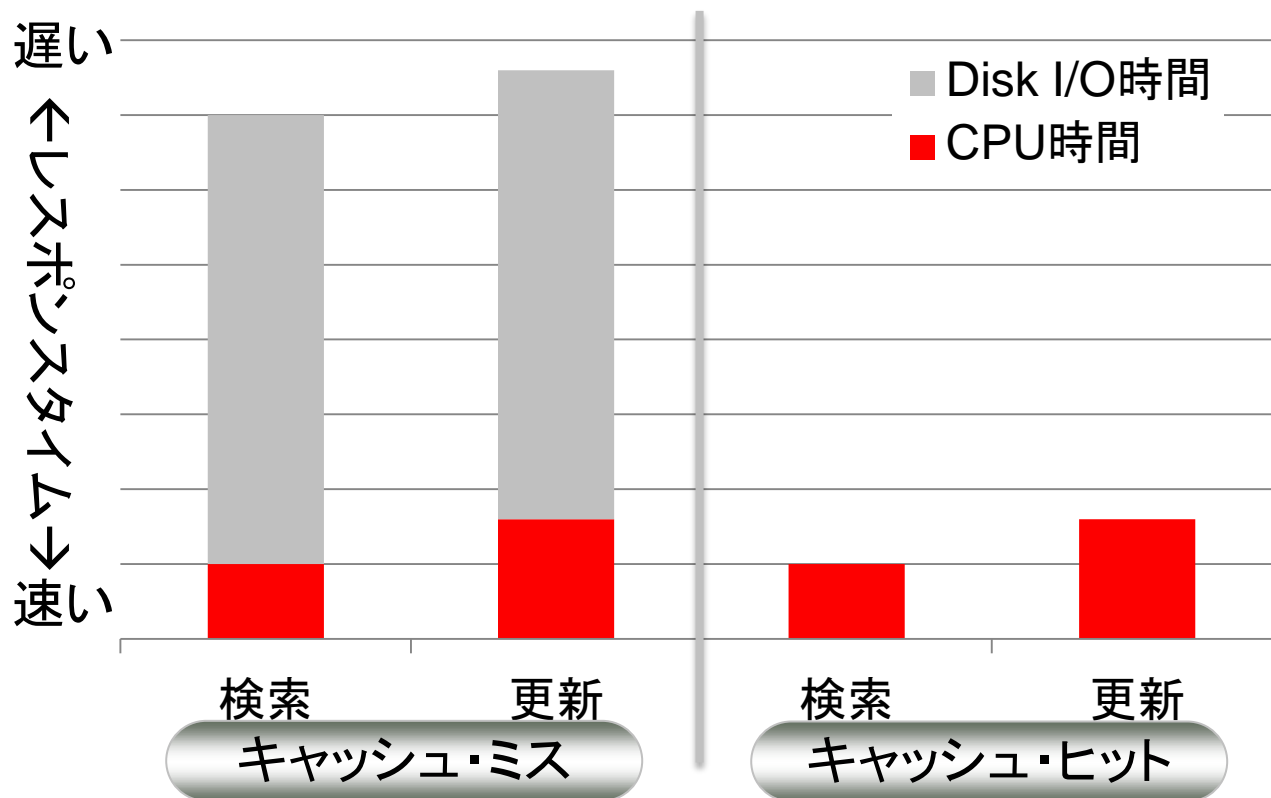
- HDD上のデータを物理メモリ(DRAM)上にキャッシュし、SQL処理を高速化
- OLTPでは、全ての処理を極力物理メモリ上で行えるようH/W構成を決定



# データベースのOLTP処理の基本動作

## SQLの処理時間の内訳イメージ

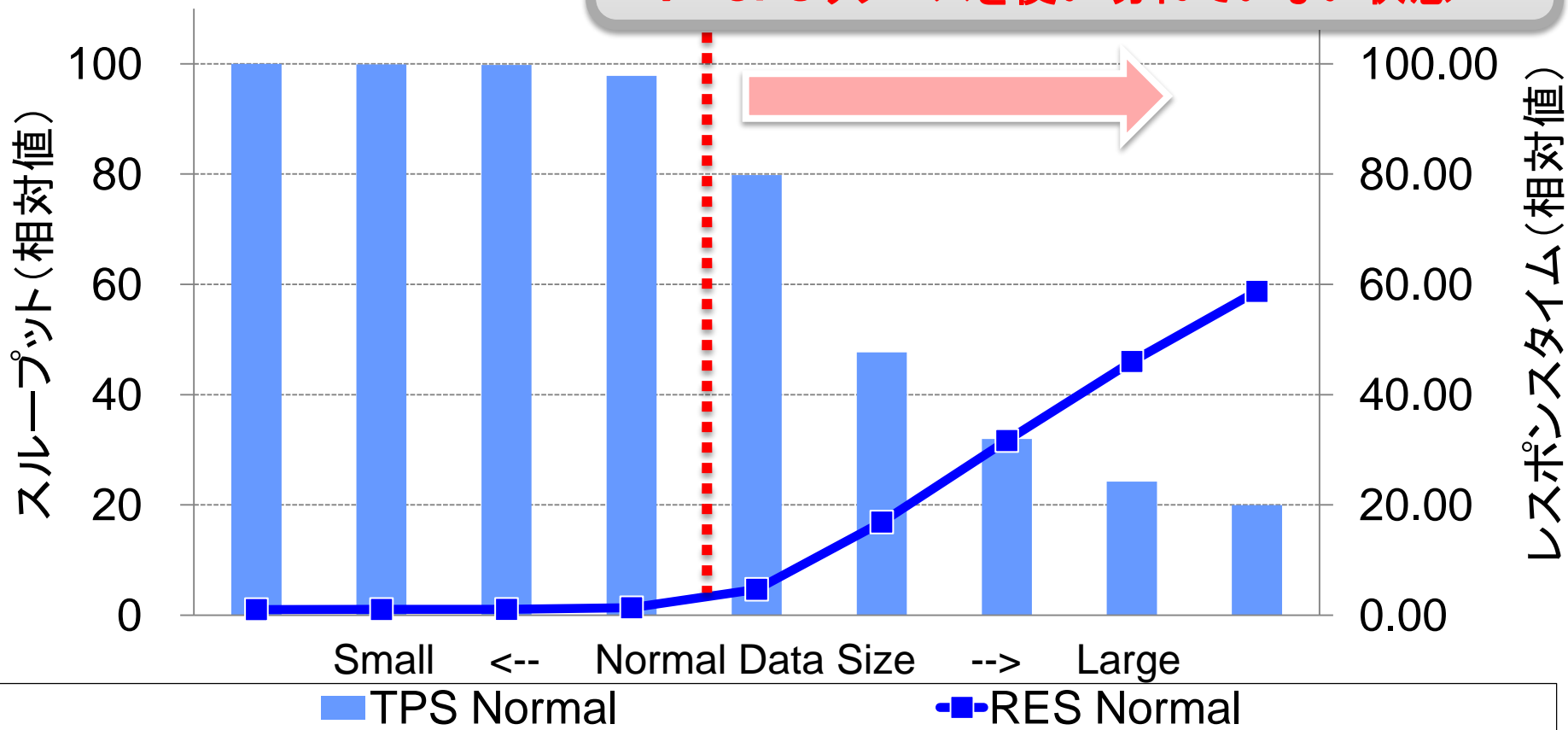
- 一般的に、SQL処理時間の大部分はHDDからのデータ読み込み待ち
- DRAM(メモリ)にデータをキャッシュすることで高速なレスポンスを実現



# OLTPシステムの課題①

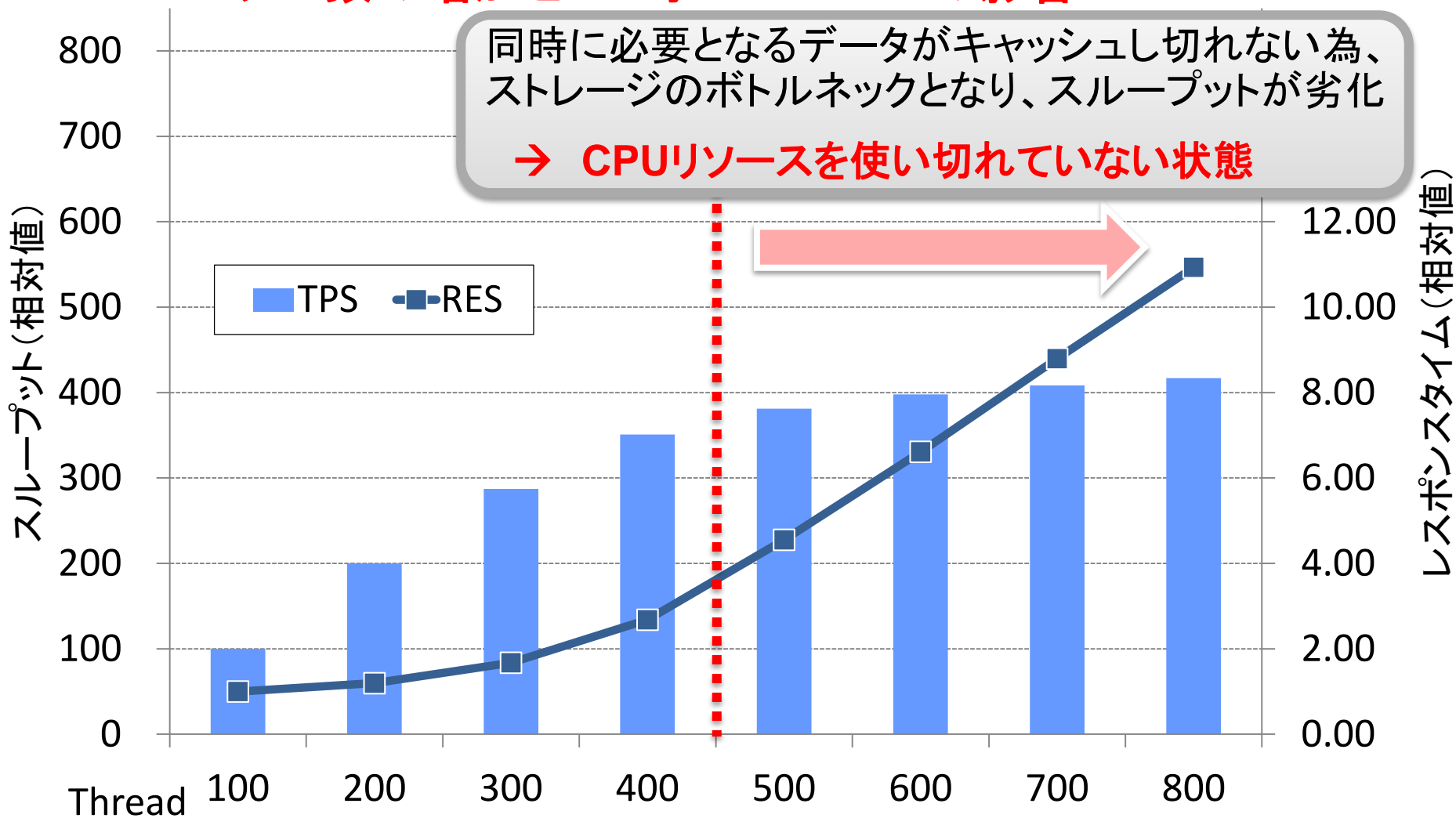
## データ量の増加とパフォーマンスへの影響

キャッシュ・ヒット率が低下し、スループットが劣化  
→ CPUリソースを使い切れていない状態



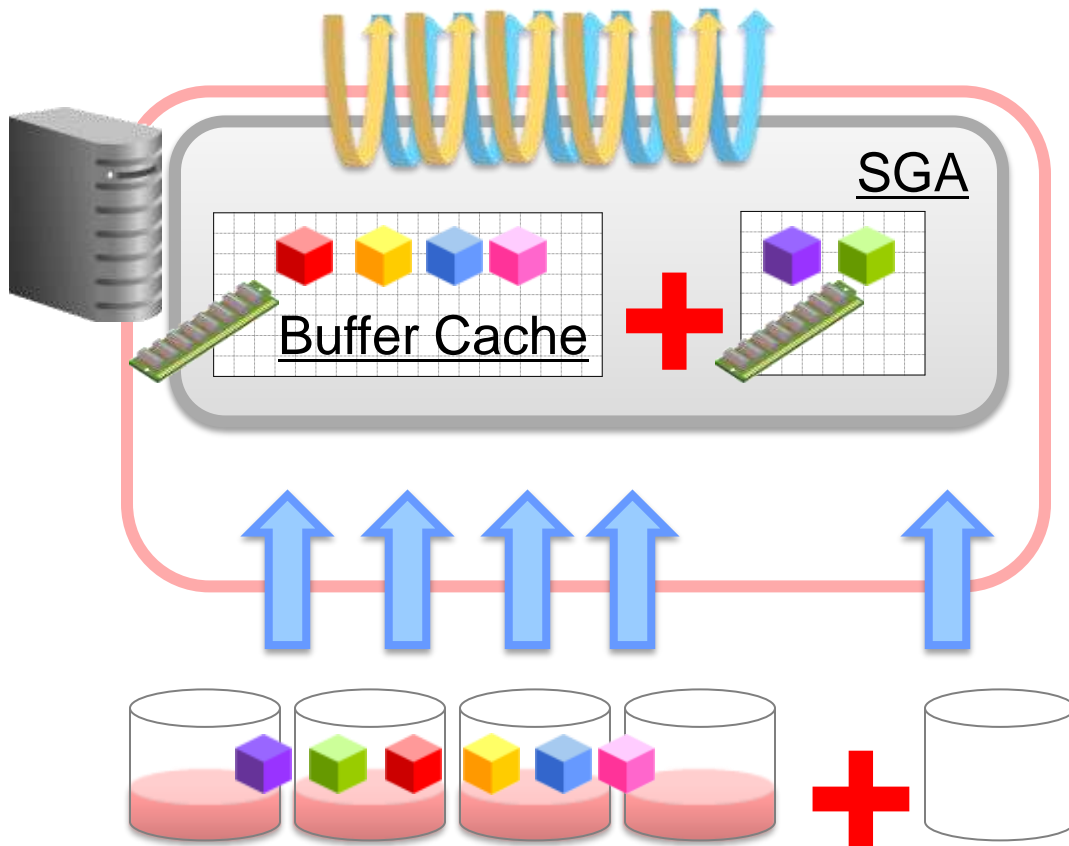
# OLTPシステムの課題②

## ユーザー数の増加とパフォーマンスへの影響



# OLTPシステムの課題に対する従来の解決策

## 高額なシステム投資



### SQLチューニング

効率的な索引の作成等  
→ 工数増大、限界有り

### 物理メモリの追加

Buffer Cacheを拡張し、  
ヒット率を高める  
→ 高コスト、増設に上限有り

### HDDの追加

データを多数のHDDに分散し、  
IOPsを高める  
→ 未使用領域増大

# Solid State Drive/Device (SSD) の登場

## HDDの高速な代替デバイス



- 性能と価格比: HDD < SSD < DRAM
- HDDが苦手とする「**Small Random Read**」が得意(10~30倍)
  - SSD: 記憶媒体としてフラッシュ・メモリを活用
  - HDDにおいてデータアクセス時に必要であったシークタイム(ヘッドをディスク上で移動させる時間)やサーチタイム(目的のデータがヘッド位置まで回転してくるまでの待ち時間)



- データベースをSSD上に構成すると、HDDより遥かに高速なI/O性能が期待
  - 特に、数件の検索処理が大量に発生するOLTP系処理で効果大
  - その他、データベースにおいて「Small Random Read」が発生する処理
    - 一時表領域からの読み込み
    - リカバリ処理

# Database Smart Flash Cache(11gR2~)

## SSDをバッファ・キャッシュの拡張領域として活用

- 現状のSSDはHDDの代替として使用するには容量あたりの単価が高い  
→ 全てのデータをSSD上に配置することは難しい傾向



## • Database Smart Flash Cache

- SSDをHDDの代替デバイスとしてではなく、キャッシュとして活用
- Oracle Database 11g Release 2 Enterprise Edition の新機能

### ※ 対応OS

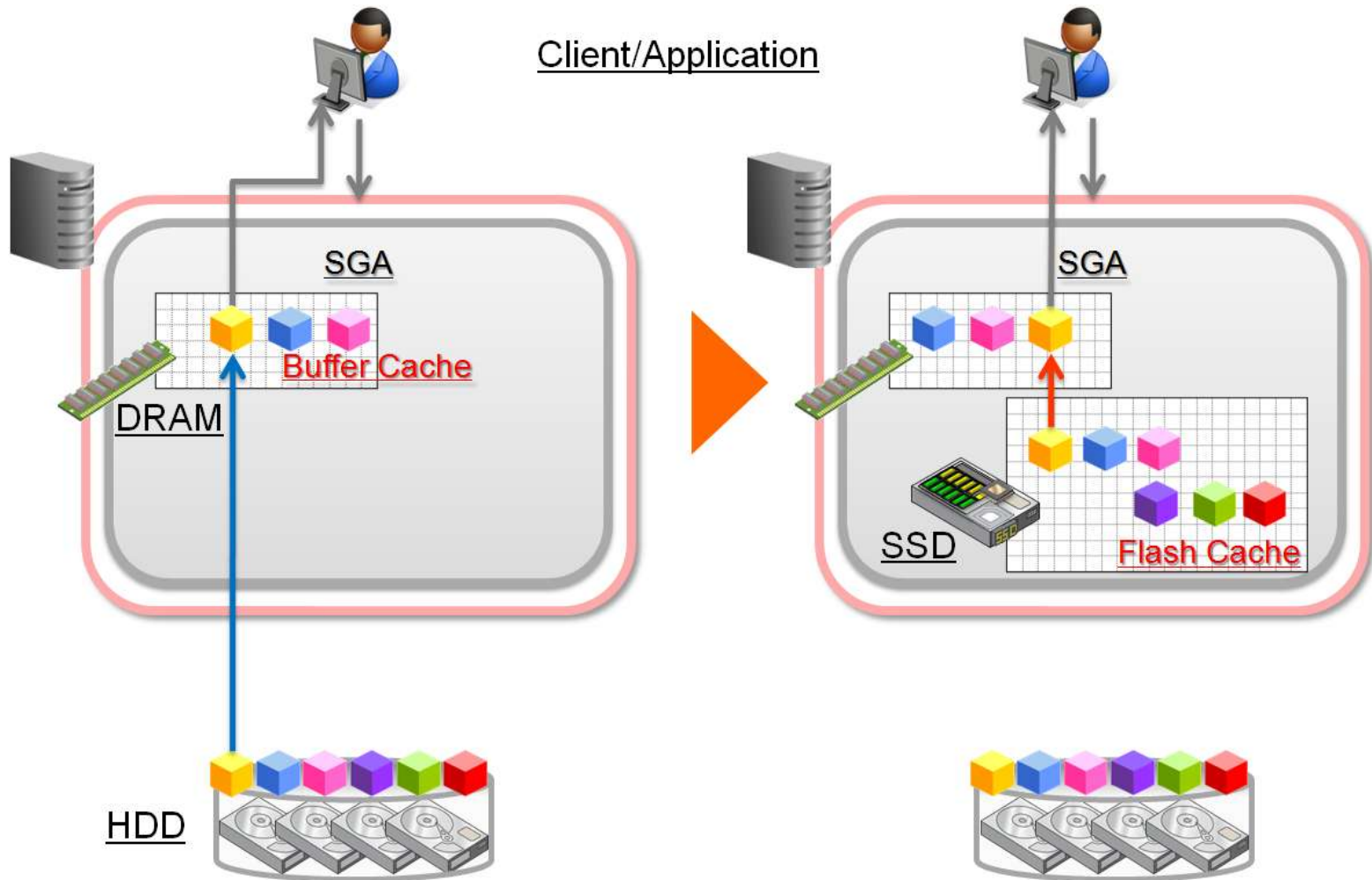
- Oracle Linux
- Oracle Sun Solaris SPARC (64-bit)
- Oracle Sun x86-64 Solaris





# Database Smart Flash Cache

データ・ブロックの読み込みを高速化

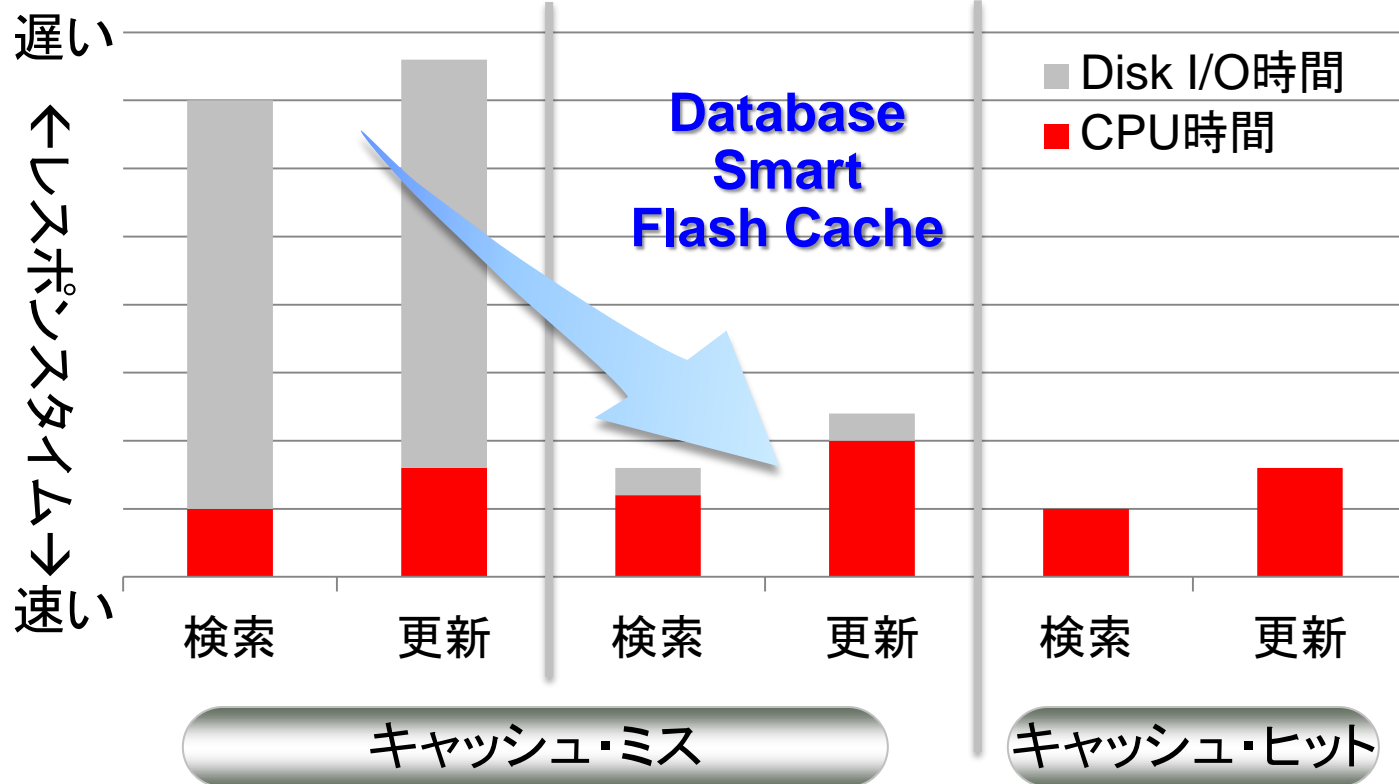


ORACLE

# Database Smart Flash Cacheの効果

## SQLの処理時間の内訳イメージ

- Buffer Cacheでキャッシュ・ミスした場合でも、I/O待ち時間を大幅に削減  
→ **キャッシュ・ヒットした場合と同等のレスポンスタイムを実現**



# Database Smart Flash Cache

## 適用ケース

- Database Smart Flash Cacheはバッファ・キャッシュの拡張領域である為、以下の条件を満たす場合に効果が期待できる
  - 待機イベント「db file sequential read」が多発している
  - Buffer Pool Advisory(AWR / STATSPACK)がBuffer Cacheの増加が有効であると示している
  - ストレージI/O性能がボトルネックであり、CPUリソースが余っている

※ 大量データの検索処理を平行実行した際のDirect Path Readのようにバッファ・キャッシュを経由しないデータ・ブロックの読みが行われる場合、Database Smart Flash Cache領域にもデータがキャッシュされない為、この効果を得ることはできません。

# Database Smart Flash Cache

## 設定パラメータ

- SSDのパスを設定
  - `db_flash_cache_file = '<filename>'`
- Database Smart Flash Cacheの領域に割り当てるサイズを設定
  - `db_flash_cache_size = <size>`
- サイズの見積もり方法
  - Database Smart Flash Cacheのサイズは、Database Buffer Cacheの**2倍から10倍**の範囲を推奨しています。
    - Database Smart Flash Cacheを有効化した場合、Database Smart Flash Cache領域に格納されているデータ・ブロックの管理領域がバッファ・キャッシュ上に割り当てられます。その管理領域のサイズは、以下の見積もり式で計算することが可能です。

$[db\_flash\_cache\_size] / [db\_block\_size] \times 100$  (RACの場合は200)

# Database Smart Flash Cache

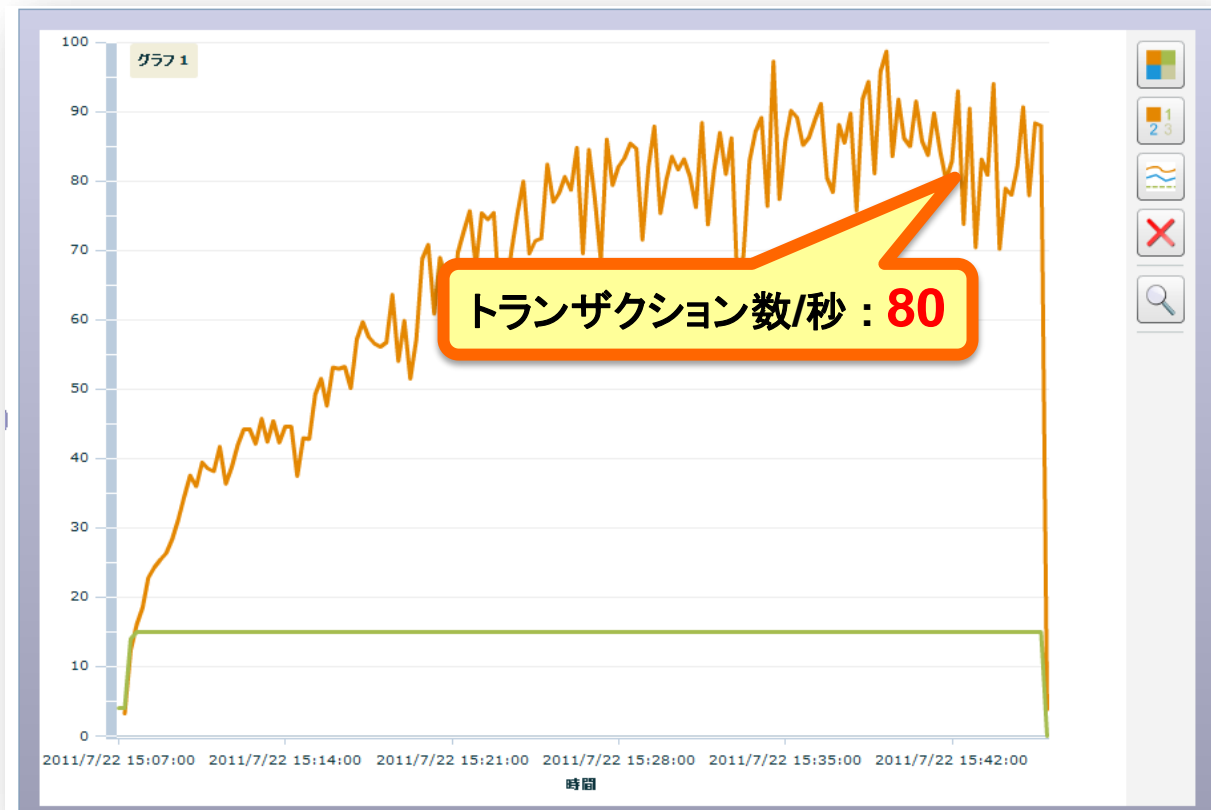
## 特徴

- 高いコストパフォーマンス
  - キャッシュとして活用することで、**アクセス頻度の高いデータのみ**SSD上へ
- サーバー上のFlash PCI Cardに対応
  - 既存システムへの導入は簡単かつ低コスト
- より賢いLRUアルゴリズムを採用
  - RACノード間でFlash Cache上のデータの一貫性を保持
  - FLASH\_CACHE { KEEP | NONE | DEFAULT } 属性により、表やパーティション単位でDatabase Smart Flash Cache領域の使用の調整が可能  
(KEEP:優先的にキャッシュする、NONE:キャッシュしない、DEFAULT:標準動作)
- バッファ・キャッシュの拡張領域
  - バッファ・キャッシュ同様、インスタンス再起動後にウォームアップが必要
  - Database Smart Flash Cache上のデータを意識せず、バックアップ可能

# Database Smart Flash Cacheの効果

## Oracle Load Testing - 負荷テスト結果4

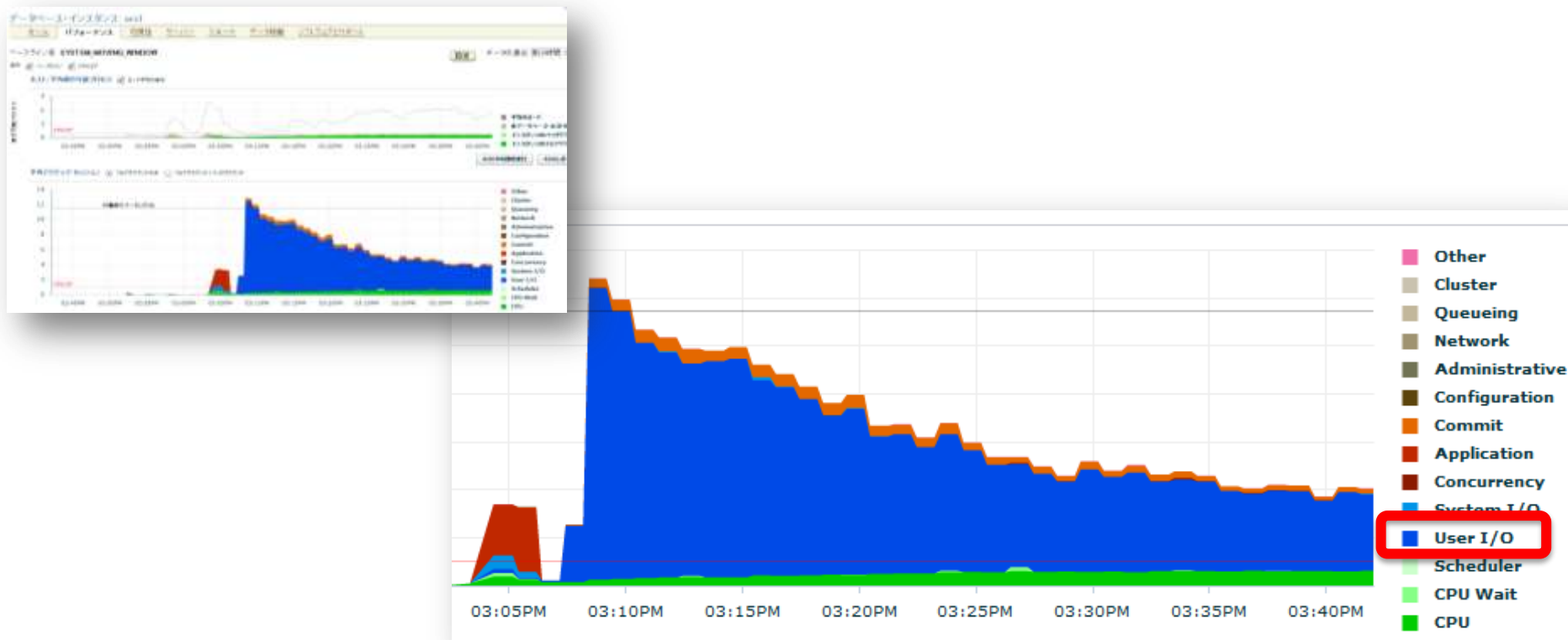
- Database Smart Flash Cache上にデータ・ブロックがキャッシュされるにつれて、スループットが上昇していくことを確認



# Oracle Enterprise Manager

## データベース・インスタンスの負荷状況を確認

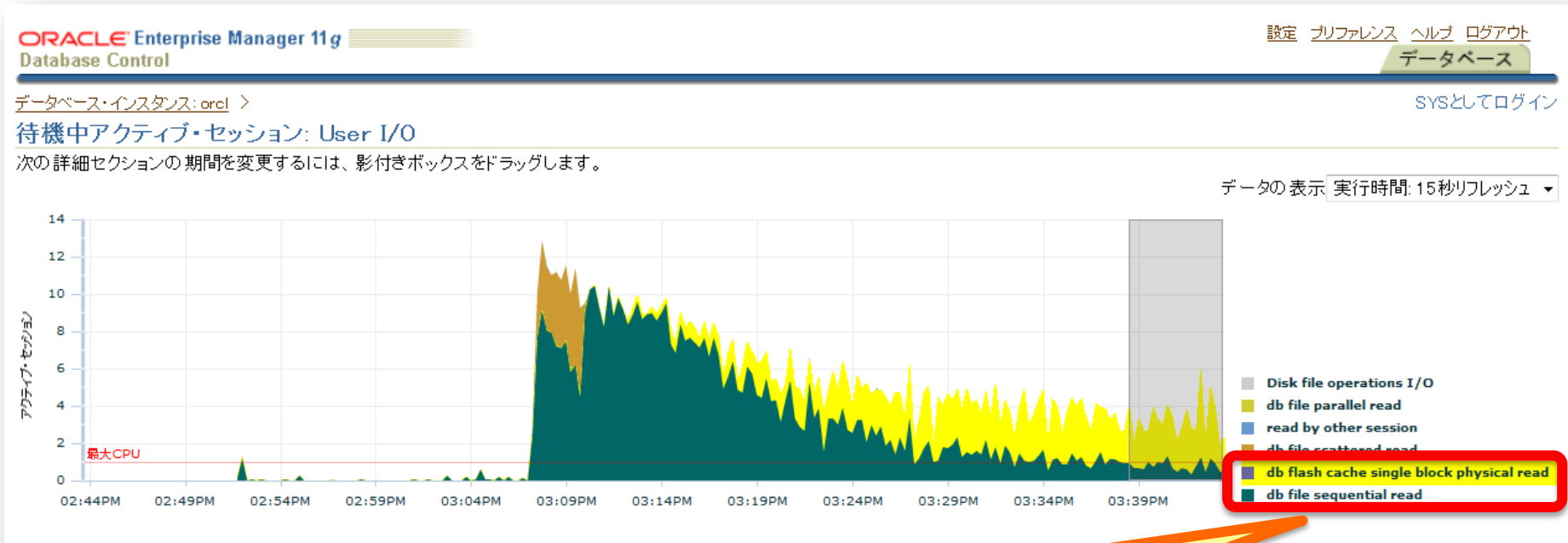
- Database Smart Flash Cacheを適用したことで、「User I/O」の比率が少しずつ低下していることを確認



# Oracle Enterprise Manager

## User I/Oの待機イベントの内訳を確認

- HDDへのI/O (db file sequential read) が徐々にSSDへのI/O (db flash cache single block physical read) に移行し、待機中のセッション数が減少



この部分を押下すると、  
待機時間のヒストグラムを参照可能



# Oracle Enterprise Manager

## 待機時間のヒストグラムを比較

- 「db flash cache single block physical read」と「db file sequential read」のI/O時間ヒストグラムを比較
  - Flash Cacheの方が、1回当たりに要する時間が圧倒的に高速
  - 待機回数の比較ではなく、1回当たりの待機に要する時間を比較が重要



db flash cache single block physical read



db file sequential read

# Oracle Load Testingのレポート作成機能

## 報告書に掲載するグラフの作成

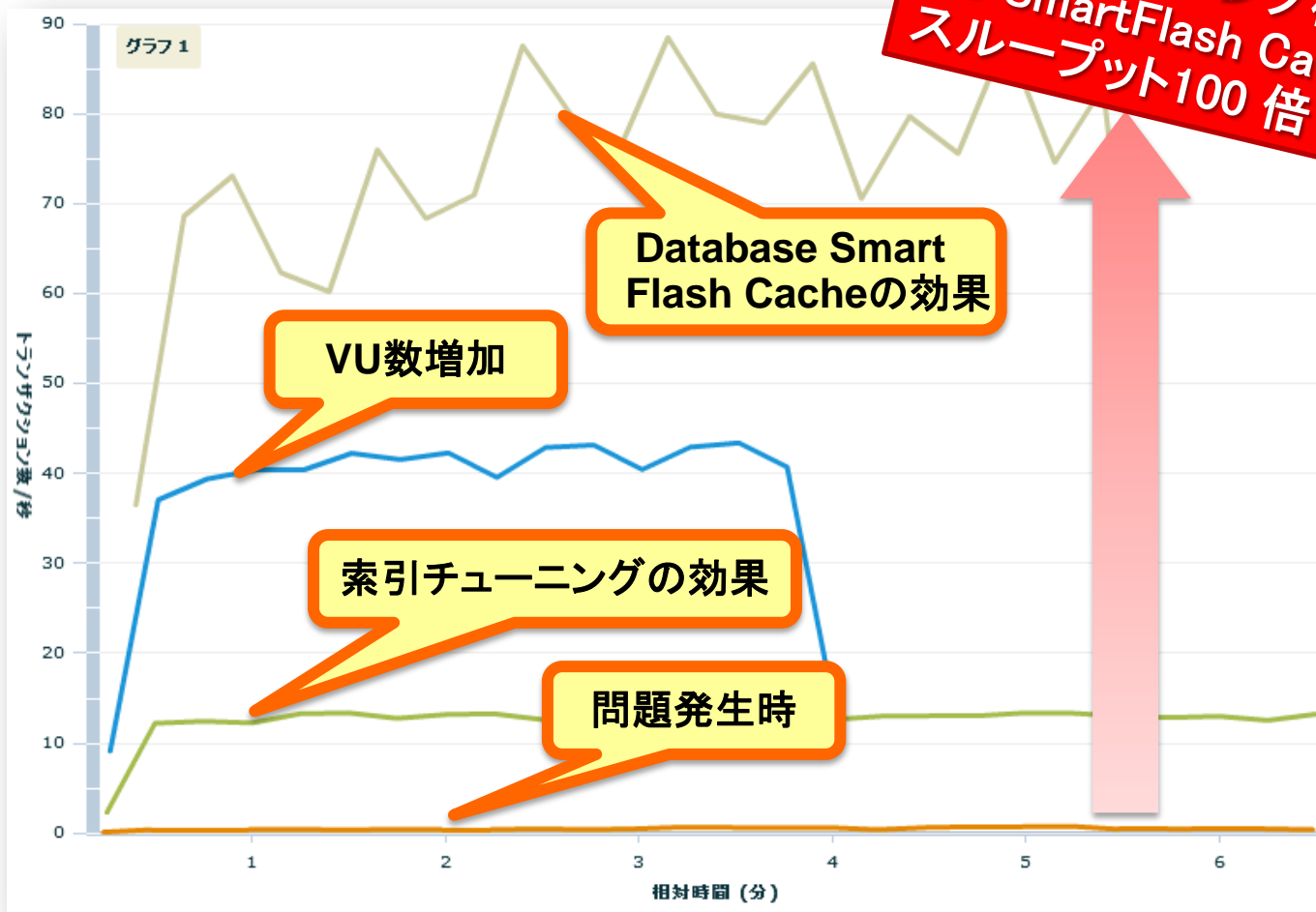
- Oracle Load Testingの「レポートの作成」タブにおいて、複数の負荷テストの結果を1つのグラフ上に表現することが可能
  - 複数の負荷テストの結果を選択後、X軸のスケールを「相対時間」に変更



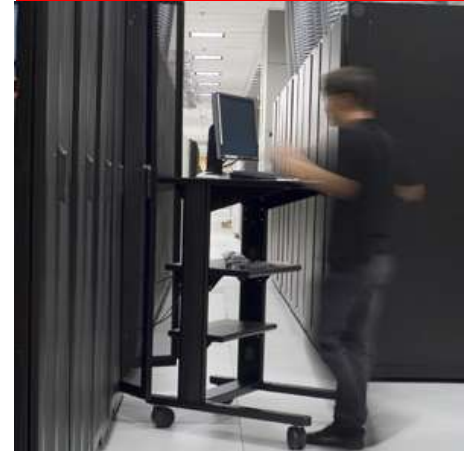
- 作成したグラフは、画像ファイルやCSVファイルとしてエクスポート（抽出）することが可能
  - 画像ファイルであれば、そのまま報告書に貼り付け
  - CSVファイルであれば、表計算ソフトで自らグラフの作成が可能

# Oracle Load Testingのレポート機能

## 報告書に掲載するグラフの作成



自動チューニング機能 & DB SmartFlash Cacheでスループット100倍!!!!



## DWH向けチューニング手法

1. 効率的なSQL実行のモニタリング
2. パーティションによるアクセス範囲の限定
3. マルチコアを活かすSQL文の並列実行
4. データ圧縮によるI/O削減
5. キャッシュ・データでの並列実行

# DWH向けチューニング手法

## 設定シナリオと到達目標

- 設定シナリオ
  - 貴方は、DWHのデータベース管理者です。
  - この1年間、会社のビジネスが絶好調なある日、ユーザーの部門長から次のようなクレームが入りました。
    - 1年前までは、DWHシステムの分析結果の表示までに要する時間が10秒程度だったのだが、ここ最近、1分～2分程度待たされるようになった。
    - 待ち時間が長い為、従業員の帰宅時間が遅くなり、残業代の支払いが・・・
  - 早急に原因を特定し、元々のレスポンスタイムを実現するよう、チューニングするように、またまた上司命令を受けました。。。
  - また、Enterprise Editionの機能を有効活用できていない状態を前提とします。
- 到達目標
  - アプリケーション側に手を加えることなく、Enterprise Editionの機能を使用して、DWHシステムの分析クエリー（SELECT文）を約10秒まで向上させる！

# DWH向けチューニング手法

## 対象SQL

```
WITH
DUMMY_SALES AS
( select *
  from (select 0 from CHANNELS ) D1, sales D2),
SACOMMON1340 AS
( select sum(T220.AMOUNT_SOLD) as c1,
       sum(T220.QUANTITY_SOLD) as c2,
       T147.CHANNEL_CLASS as c3,
       T228.CALENDAR_QUARTER_DESC as c4,
       T228.CALENDAR_YEAR as c5,
       T185.PROD_CATEGORY as c6
  from CHANNELS T147,
       PRODUCTS T185,
       DUMMY_SALES T220,
       TIMES T228
  where ( T220.TIME_ID < to_date('1999/01/01','YYYY/MM/DD')
        and T228.TIME_ID = T220.TIME_ID
        and T147.CHANNEL_ID = T220.CHANNEL_ID
        and T185.PROD_ID = T220.PROD_ID)
  group by T147.CHANNEL_CLASS,
         T185.PROD_CATEGORY,
         T228.CALENDAR_QUARTER_DESC,
         T228.CALENDAR_YEAR),
SAWITH0 AS
( select distinct 0 as c1, D1.c3 as c2, D1.c4 as c3, D1.c5 as c4,
       D1.c6 as c5, D1.c2 as c6, D1.c1 as c7,
       cast(NULL as DOUBLE PRECISION ) as c8
  from SACOMMON1340 D1),
SAWITH1 AS
( 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, D1.c7 as c7, D1.c8 as c8,
       sum(D1.c7) as c9
  from SAWITH0 D1
  group by D1.c1, D1.c2, D1.c3, D1.c4, D1.c5, D1.c6, D1.c7, D1.c8),
SAWITH2 AS
( select distinct 1 as c1, D1.c3 as c2, D1.c4 as c3, D1.c5 as c4,
       D1.c6 as c5, D1.c2 as c6, D1.c1 as c7
  from SACOMMON1340 D1),
SAWITH3 AS
( 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, D1.c7 as c7, sum(D1.c6) as c8,
       sum(D1.c7) as c9
  from SAWITH2 D1
  group by D1.c1, D1.c2, D1.c3, D1.c4, D1.c5, D1.c6, D1.c7),
SAWITH4 AS
(
  ( 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, D1.c7 as c7, D1.c8 as c8,
       sum(D1.c9) over (partition by D1.c3, D1.c4, D1.c5) as c9
    from SAWITH1 D1
    union all
    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, D1.c7 as c7,
       sum(D1.c8) over (partition by D1.c3, D1.c4, D1.c5) as c8,
       sum(D1.c9) over (partition by D1.c3, D1.c4, D1.c5) as c9
    from SAWITH3 D1 )
)
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, D1.c7 as c7, D1.c8 as c8, D1.c9 as c9
from SAWITH4 D1 order by c1, c3, c5, c4;
```

CPUバウンド

```
WITH
SACOMMON1340 AS
( select sum(T220.AMOUNT_SOLD) as c1,
       sum(T220.QUANTITY_SOLD) as c2,
       T147.CHANNEL_CLASS as c3,
       T228.CALENDAR_QUARTER_DESC as c4,
       T228.CALENDAR_YEAR as c5,
       T185.PROD_CATEGORY as c6
  from CHANNELS T147,
       PRODUCTS T185,
       SALES T220,
       TIMES T228
  where ( T220.TIME_ID < to_date('1999/01/01','YYYY/MM/DD')
        and T228.TIME_ID = T220.TIME_ID
        and T147.CHANNEL_ID = T220.CHANNEL_ID
        and T185.PROD_ID = T220.PROD_ID)
  group by T147.CHANNEL_CLASS,
         T185.PROD_CATEGORY,
         T228.CALENDAR_QUARTER_DESC,
         T228.CALENDAR_YEAR),
SAWITH0 AS
( select distinct 0 as c1, D1.c3 as c2, D1.c4 as c3, D1.c5 as c4,
       D1.c6 as c5, D1.c2 as c6, D1.c1 as c7,
       cast(NULL as DOUBLE PRECISION ) as c8
  from SACOMMON1340 D1),
SAWITH1 AS
( 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, D1.c7 as c7, D1.c8 as c8, sum(D1.c7) as c9
  from SAWITH0 D1
  group by D1.c1, D1.c2, D1.c3, D1.c4, D1.c5, D1.c6, D1.c7, D1.c8),
SAWITH2 AS
( select distinct 1 as c1, D1.c3 as c2, D1.c4 as c3,
       D1.c5 as c4, D1.c6 as c5, D1.c2 as c6, D1.c1 as c7
  from SACOMMON1340 D1),
SAWITH3 AS
( 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, D1.c7 as c7, sum(D1.c6) as c8, sum(D1.c7) as c9
  from SAWITH2 D1
  group by D1.c1, D1.c2, D1.c3, D1.c4, D1.c5, D1.c6, D1.c7),
SAWITH4 AS
(
  ( 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, D1.c7 as c7, D1.c8 as c8,
       sum(D1.c9) over (partition by D1.c3, D1.c4, D1.c5) as c9
    from SAWITH1 D1
    union all
    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, D1.c7 as c7,
       sum(D1.c8) over (partition by D1.c3, D1.c4, D1.c5) as c8,
       sum(D1.c9) over (partition by D1.c3, D1.c4, D1.c5) as c9
    from SAWITH3 D1 )
)
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, D1.c7 as c7, D1.c8 as c8, D1.c9 as c9
from SAWITH4 D1
order by c1, c3, c5, c4;
```

I/Oバウンド



# 1. Real-Time SQL Monitoring (SQL監視)で現状分析

まず、問題となっているクエリーのレスポンス・タイムを測定します。Enterprise Editionの機能を未使用な状態とします。ただし、SQLの分析は、Enterprise ManagerのリアルタイムSQL監視を使用します。

# Real-Time SQL Monitoring (SQL監視)

## 現状のパフォーマンス分析

- Enterprise Managerの「パフォーマンス」タブを選択し、「その他の管理リンク」セクション内の「SQL監視」を押下

The screenshot shows the Oracle Enterprise Manager Performance page. The 'パフォーマンス' (Performance) tab is highlighted with a red box and labeled '① パフォーマンスタブを押下'. In the 'その他の監視リンク' (Other Monitoring Links) section, the 'SQL監視' (SQL Monitoring) link is highlighted with a red box and labeled '② [SQL監視] を押下'. The page content includes a navigation bar with tabs for Home, Performance, Availability, Servers, Schemas, Data Movement, and Software and Support. Below the navigation bar, there are sections for 'その他の監視リンク' and '関連リンク' (Related Links).

その他の監視リンク

ASHからの上位セッションおよび上位SQLのデータは、「トップ・アクティビティ」ページで検索できます。

- トップ・アクティビティ
- 上位コンシューマ
- 重複したSQL
- ブロックしているセッション
- ハンガ分析
- インスタンス・ロック
- インスタンス・アクティビティ
- セッションの検索
- SQLの検索
- スナップショット
- AWRベースライン
- SQLチューニング・セット
- SQLパフォーマンス・アナライザ
- SQL監視

ホーム パフォーマンス 可用性 サーバー スキーマ データ移動 ソフトウェアとサポート

関連リンク

Exadataセル・ターゲットの追加  
アドバイザ・セントラル  
ジョブ  
ターゲット・プロパティ  
ポリシー・グループ  
メモリー・アクセス・モードで監視

SQL  
アラ  
スケジューラ・セントラル  
ブラックアウト  
メトリックとポリシー設定  
ユーザー定義メトリック

① パフォーマンスタブを押下

② [SQL監視] を押下



# Real-Time SQL Monitoring (SQL監視)

## 現状のパフォーマンス分析

- 「SQL監視」の画面では実行済のSQLだけではなく、実行中のSQLの詳細状況が確認可能

ORACLE Enterprise Manager 11g Database Control

データベース・インスタンス: orcl > SYSとしてログイン

監視されたSQL実行

現在までのアクティブ期間 2時間

リフレッシュ 10秒 リフレッシュ停止

ステータス	期間	SQL ID	ユーザー	パラレル	データベース時間	I/Oリクエスト	開始	終了	SQLテキスト
✓	2.5m	33yv4svtpc8z	SH		2.5m	8,917	11:38:12	11:40:42	WITH DUMMY_SALES AS ( select * from (select 0 fr
✓	18.0s	6gvch1xu9ca3g	SYS		17.7s	959	11:26:12	11:26:30	DECLARE job_BINARY_INTEGER := :job; next_date DA

② ステータス列のアイコンを押下

① [SQLテキスト]から確認したいSQL文を特定

# CPUバウンドなSQL

## SQL監視を用いたパフォーマンスの確認

- 各SQLの詳細画面では、SQLの実行計画、I/O量・バッファ読み取り量、データベース時間等の内訳を確認可能



実行計画のステップ

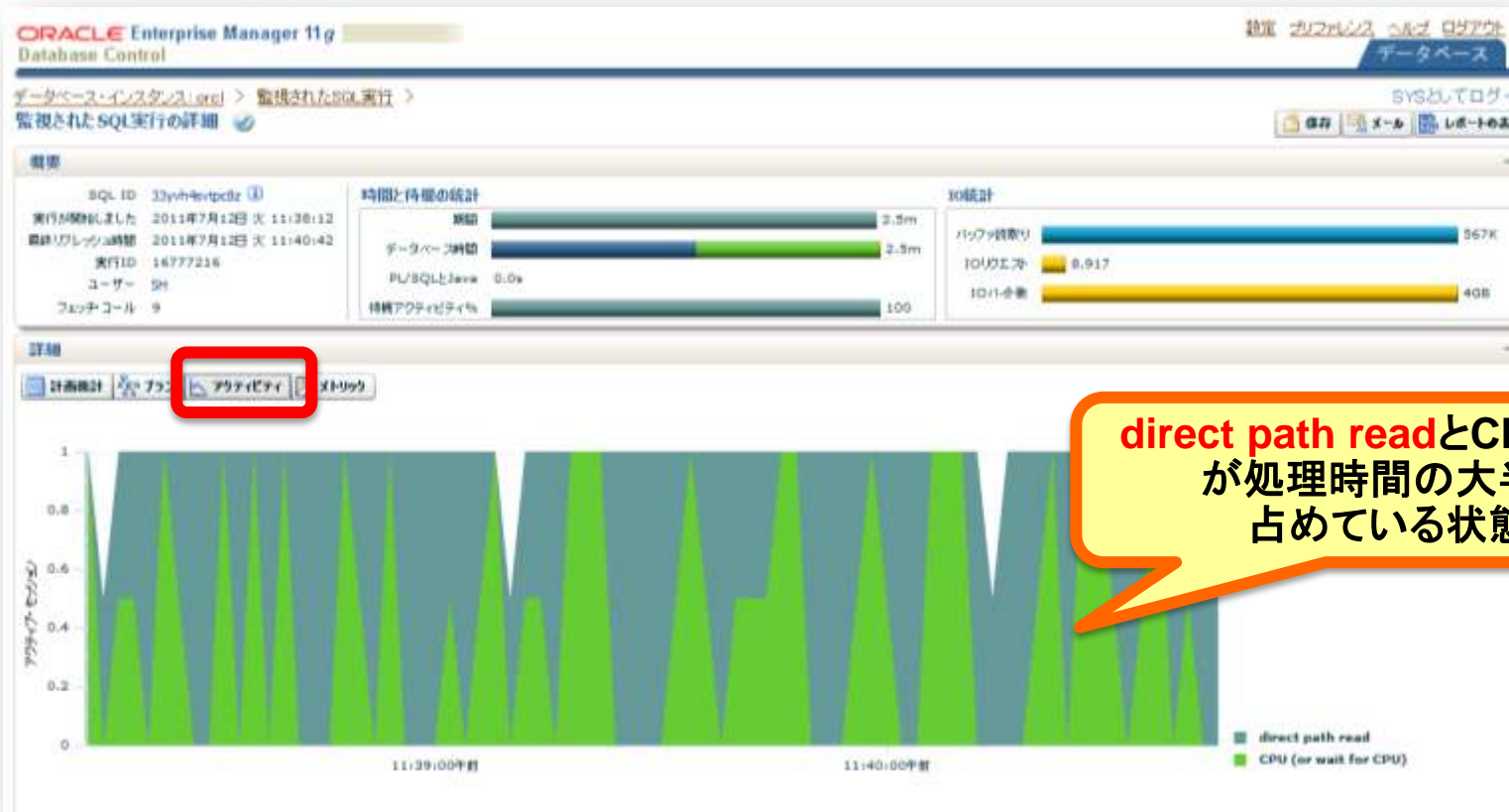
CPUとI/Oの比率が半々

4GBのデータの読み込みが発生

# CPUバウンドなSQL

## SQL監視を用いたパフォーマンスの確認

- 「アクティビティ」ページで、SQLを実行している(した)サーバー・プロセスで発生した待機イベントを時系列で参照可能



# I/OバウンドなSQL

## SQL監視を用いたパフォーマンスの確認

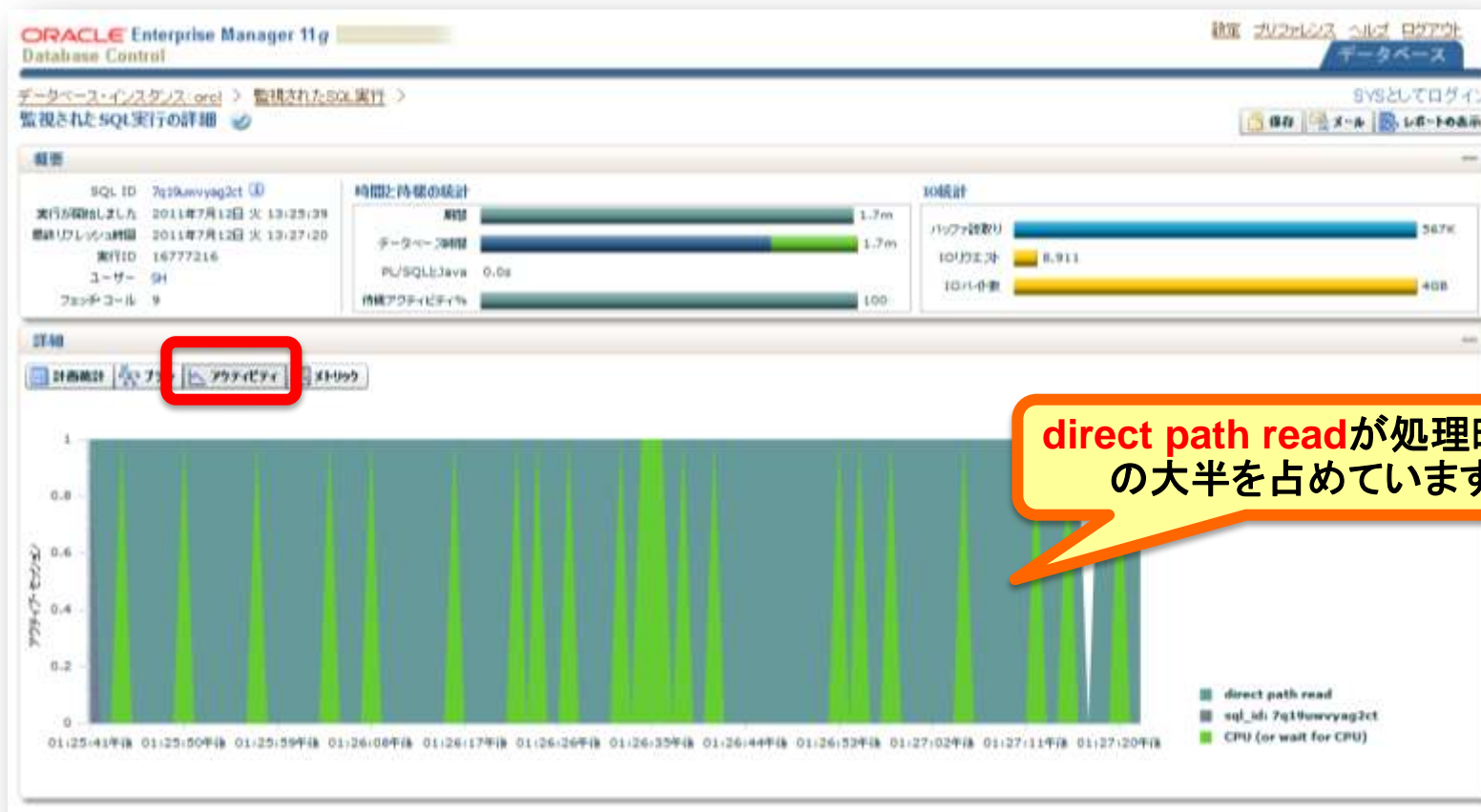
- データベース時間の内訳より、ユーザーI/Oの割合が高いことを確認

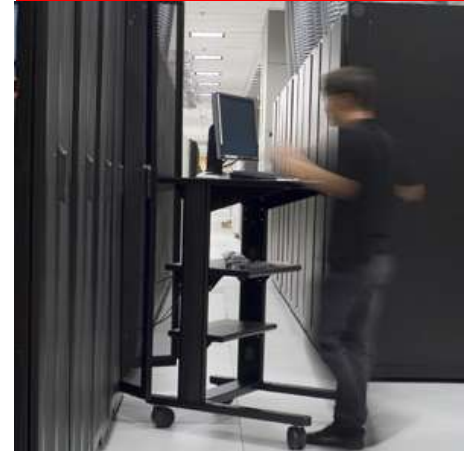


# I/OバウンドなSQL

## SQL監視を用いたパフォーマンスの確認

- 待機イベントが主に「direct path read」であることを確認





## 2. Partition Advisorで表のパーティション化

どこから手を付けるべきか判断できない状態であっても、まずは、Enterprise ManagerのSQLアクセス・アドバイザを実行します。

# Oracle Enterprise Manager

## SQLアクセス・アドバイザの使用

- Enterprise Managerの「パフォーマンス」→「アドバイザ・セントラル」→「SQLアドバイザ」

The screenshot shows the Oracle Enterprise Manager interface. At the top, the navigation menu includes 'ホーム', 'パフォーマンス', '可用性', 'サーバー', 'スキーマ', 'データ移動', and 'ソフトウェアとサポート'. The 'パフォーマンス' (Performance) tab is highlighted with a red box. Below the menu, the '関連リンク' (Related Links) section contains a grid of links. The link 'SQLアクセス・アドバイザの追加' (Add SQL Access Advisor) is highlighted with a red box. Below this, the 'アドバイザ・セントラル' (Advisor Central) page is shown, with the 'アドバイザ' (Advisor) tab selected. In the 'アドバイザ' section, the 'SQLアドバイザ' (SQL Advisor) link is highlighted with a red box.

ホーム **パフォーマンス** 可用性 サーバー スキーマ データ移動 ソフトウェアとサポート

関連リンク

<a href="#">EMのSQL履歴</a>	<a href="#">SQLアクセス・アドバイザの追加</a>	<a href="#">SQLワークシート</a>
<a href="#">アクセス</a>	<a href="#">アドバイザ・セントラル</a>	<a href="#">アラート・ログの内容</a>
<a href="#">アラート履歴</a>	<a href="#">ジョブ</a>	<a href="#">スケジューラ・セントラル</a>
<a href="#">すべてのメトリック</a>	<a href="#">ターゲット・プロパティ</a>	<a href="#">ブラックアウト</a>
<a href="#">ベースライン・メトリックしきい値</a>	<a href="#">ポリシー・グループ</a>	<a href="#">メトリックとポリシー設定</a>
<a href="#">メトリック収集エラー</a>	<a href="#">メモリー・アクセス・モードで監視</a>	<a href="#">ユーザー定義メトリック</a>
<a href="#">監視相</a>		

データベース・インスタンス: orcl >

アドバイザ・セントラル

アドバイザ チェック

アドバイザ

<a href="#">ADDM</a>	<a href="#">MTTRアドバイザ</a>	<a href="#">SQLアドバイザ</a>
<a href="#">SQLパフォーマンス・アナライザ</a>	<a href="#">Streamsパフォーマンス・アドバイザ</a>	<a href="#">セグメントアドバイザー</a>
<a href="#">データ・リカバリ・アドバイザ</a>	<a href="#">メモリー・アドバイザ</a>	<a href="#">自動UNDO管理</a>

# Oracle Enterprise Manager

## SQLアクセス・アドバイザの使用

ORACLE Enterprise Manager 11g  
Database Control

設定 プリファレンス ヘルプ ログアウト  
データベース

データベース・インスタンス: orcl > アドバイザ・セントラル >

SYSとしてログイン

### SQLアドバイザ

SQLアドバイザはSQLに関連するいくつかの重要なユースケースに対処します。それはSQLワークロードを最適化する物理構造の識別、大容量の実行計画による個々の文のチューニング、結果セットの相違点の識別および修正、エラーが発生したSQLに対するテストケースの構築などです。

#### SQLアクセス・アドバイザ

SQLアクセス・アドバイザ

SQLのワークロード全体を評価し、SQLワークロードの総合的なパフォーマンスを向上させる索引、パーティショニングおよびマテリアライズド・ビューを推奨します。

#### SQLチューニング・アドバイザ

SQLチューニング・アドバイザ  
自動SQLチューニングの結果

個々のSQL文を分析し、SQLのパフォーマンスに対してSQLプロファイル、統計、索引および再構築されたSQLを推奨します。  
監視されている高負荷のSQLIにおける、SQLチューニング・アドバイザの自動実行の結果を表示します。

#### SQL 修復アドバイザ

ORACLE Enterprise Manager 11g  
Database Control

設定 プリファレンス ヘルプ ログアウト  
データベース

アドバイザ・セントラル >

SYSとしてログイン

### SQLアクセス・アドバイザ: 初期オプション

初期オプションのセットを選択します。

取消 続行

索引、パーティショニング、マテリアライズド・ビュー、パーティショニングなどのみの使用の検証

新規アクセス構造の推奨

保存済みのアクセス構造またはテンプレートからの継承オプション

#### 概要

SQLアクセス・アドバイザは、ワークロード・ソースのSQL文を評価し、ワークロード全体のパフォーマンスを向上させる索引、パーティション、マテリアライズド・ビューおよびマテリアライズド・ビュー・ログを提案します。

ヒント ウィザードの開始地点を選択しています。すべてのオプションは、ウィザード内から変更可能です。

取消 続行

データベース | 設定 | プリファレンス | ヘルプ | ログアウト

ORACLE



# Oracle Enterprise Manager

## SQLアクセス・アドバイザの使用

ORACLE Enterprise Manager 11g Database Control

設定 プリファレンス ヘルプ ログアウト データベース

ワークロード・ソース 推奨オプション スケジュール 確認

### SQLアクセス・アドバイザ: ワークロード・ソース

データベース orcl  
ログイン時の権限 SYS

取消 ステップ 1 / **次へ(X)**

以下に使用するワークロード・ソースを選択します。基礎となる表にアクセスするすべてのSQL文を代表するワークロードが、最も適しています。

- 現在と最近のSQLアクティビティ  
SQLがキャッシュから選択されます。
- 既存のSQLチューニング・セットを使用  
SQLチューニング・セット

### SQLアクセス・アドバイザ: 推奨オプション

データベース orcl  
ログイン時の権限 SYS

取消 戻る(K) ステップ 2 / **次へ(X)**

#### 推奨するアクセス構造

- 索引
- マテリアライズドビュー
- パーティショニング

#### 有効範囲

アドバイザは、制限モードまたは包括モードのどちらかで実行できます。制限モードは、特定のしきい値より小さいコストの文を無視し、最もコストの大きい文の処理後に迅速に推奨を返します。包括モードでは、完全な分析が行われます。

- 制限  
分析は最もコストの大きい文に絞り込まれます
- 包括  
完全な分析が行われます

# Oracle Enterprise Manager

## SQLアクセス・アドバイザの結果を確認

- 推奨事項の詳細で、表のパーティション化が推奨されていることを確認

Oracle Enterprise Manager 11g Database Control

タスクの結果: SQLACCESS4183519

タスク名: SQLACCESS4183519  
 ステータス: COMPLETED  
 アドバイザ・モード: LIMITED  
 スケジュールのジョブ: ADV\_SQLACCESS4183519

このグラフと表には、最もコストが改善した順に推奨事項がリストアップされています。

コスト改善額の推奨事項

推奨事項 ID	コスト改善額
1	252,000

コスト改善

実行用の推奨の選択

実行/RETAINアクションをすべて  
 推奨事項の詳細  スケジュール実行  SQL表示

すべて選択 | 選択解除

推奨事項の詳細

SQLアクセス・アドバイザは、デフォルトのオブジェクト名を生成し、タスクの作成中に指定されたデフォルトのスキーマと表領域を使用しますが、これらは変更できます。読取り専用として表示されている名前や依存名を編集すると、適宜更新されます。「表領域」フィールドが空白の場合は、スキーマのデフォルト表領域が使用されます。「OK」をクリックすると、SQLスクリプトが変更されますが、推奨ページかSQL文ページで「スケジュール実行」を選択するまで実際には実行されません。

アクション

ステータス	推奨事項 ID/アクション	オブジェクト名	オブジェクトの種別	スキーマ	表領域	パーティション	推奨使用量 (MB)
■	1.2 PARTITION TABLE SALES			SH		("TIME_ID") PARTITION BY RANGE ("TIME_ID") INTERVAL...	0.0
✓	1.2 RETAIN INDEX	PRODUCTS_I					0.1
✓	1.2 RETAIN INDEX	PRODUCTS_I					0.1
✓	1 RETAIN INDEX	CHANNELS_F					0.1

推奨の影響を受けるSQL

文ID	文	元のコスト	新規コスト	コスト改善率 (%)	実行回数	
7055	WITH DUMMY_SALES AS ( select * from	160677	35283	125394	78.04	1

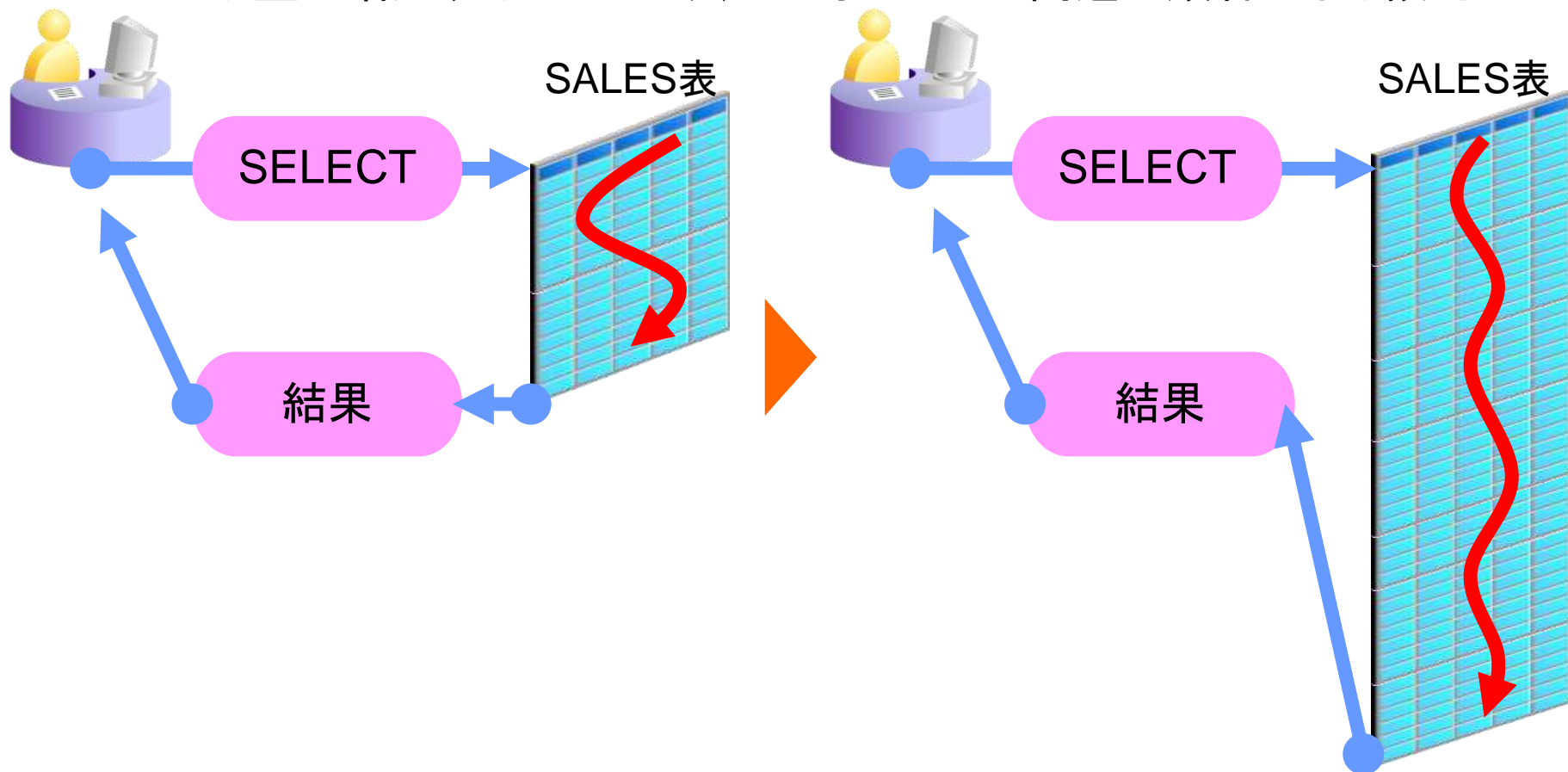
パーティション・キー ("TIME\_ID")

SQLパーティション PARTITION BY RANGE ("TIME\_ID") INTERVAL(...)

# データ量増大により発生する課題

## データ集計処理のパフォーマンスの劣化

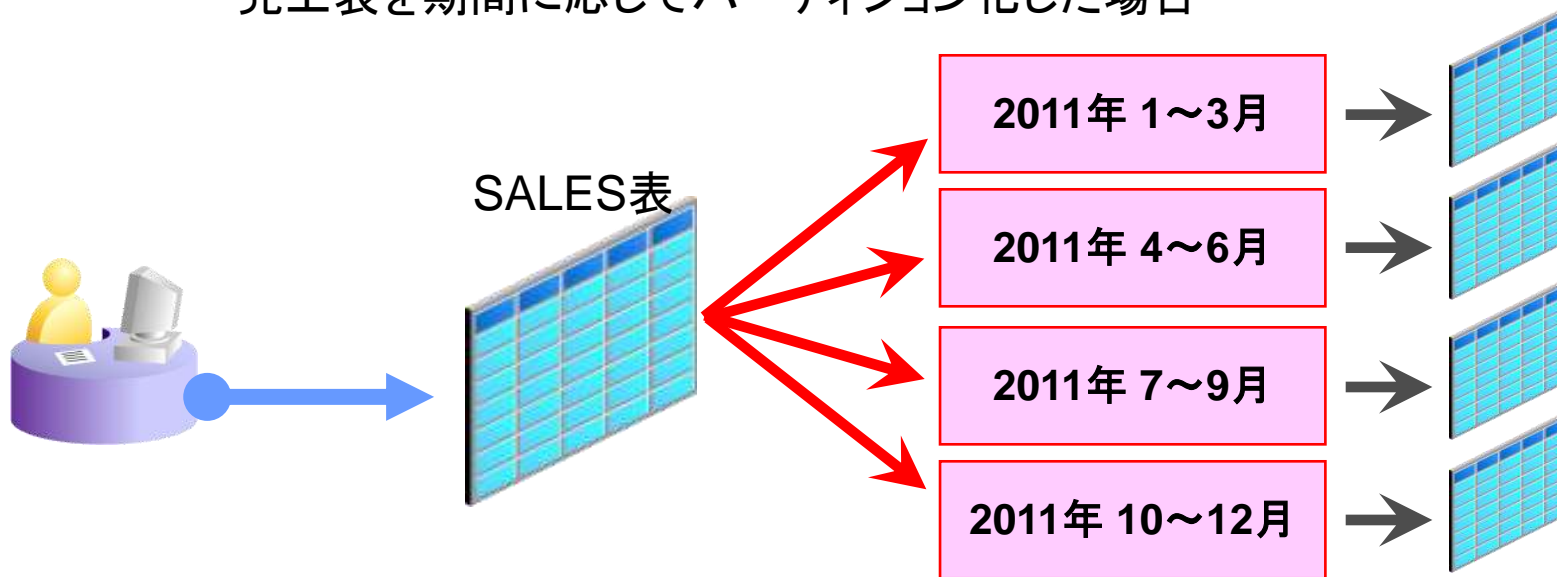
- データ量が増大することにより、パフォーマンス問題が顕著になる傾向



# Oracle Partitioning Option

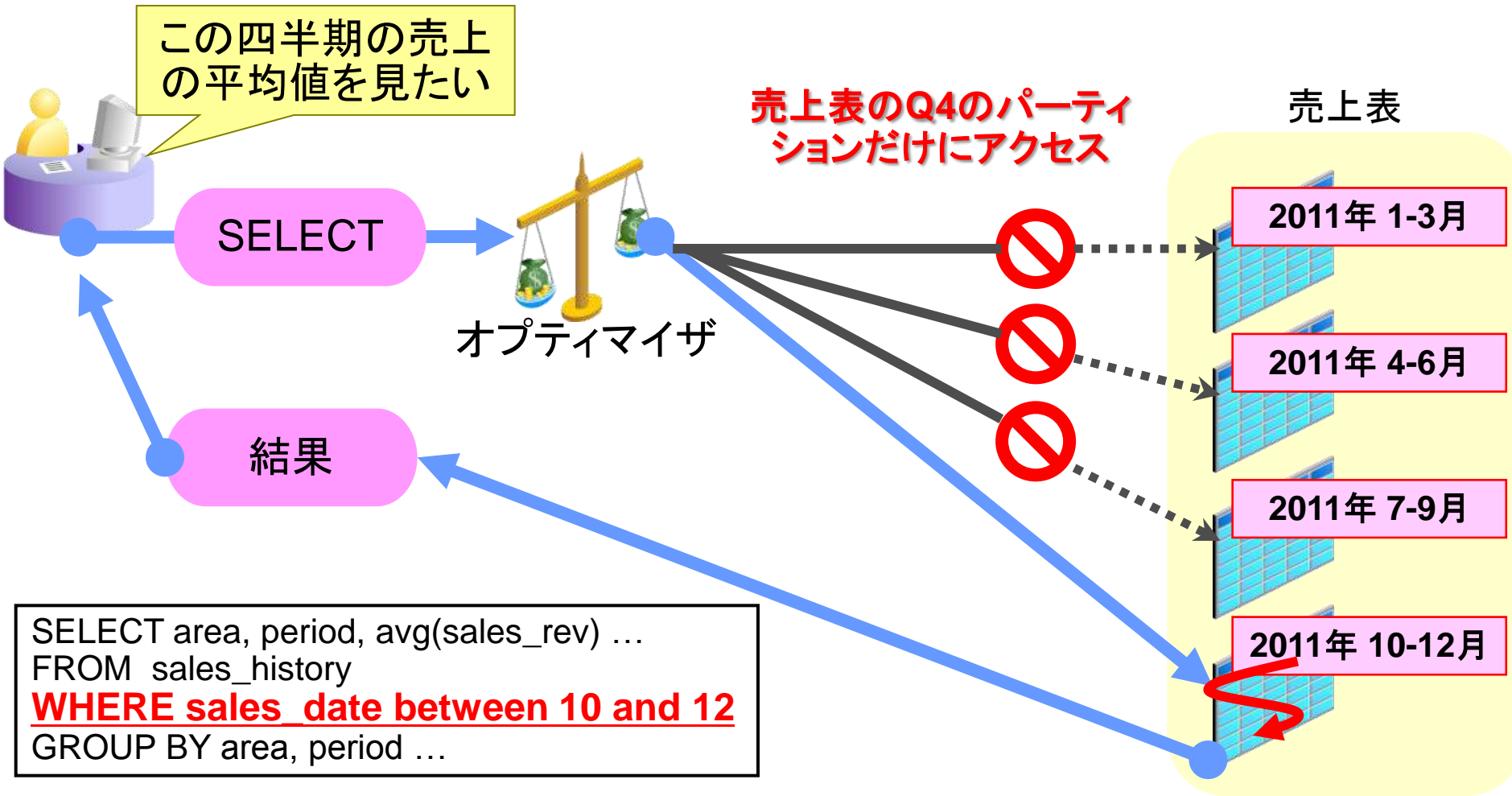
## 表を内部的に分割して管理

- 設定したルールに従って、大きな表を内部的に分割して管理
  - パフォーマンスの向上、運用管理工数の削減
  - アプリケーションから透過的（従来同様に1つの表としてアクセス可能）
- 例)
  - 売上表を期間に応じてパーティション化した場合



# Oracle Partitioning Option

パーティション・プルーニング (読み込み対象データを限定)



# Oracle Partitioning Option

## 対象SQLで期待される効果

- SALES表は、1998年～2001年の4年間分のデータを保持
- 2つの対象SQLは、1998年の1年間分のデータのみが検索対象

```
WITH
SACOMMON1340 AS
(  select sum(T220.AMOUNT_SOLD) as c1,
    .....
    from CHANNELS T147,
        SALES T220,
    .....
 where ( T220.TIME_ID < to_date('1999/01/01','YYYY/MM/DD')
        and T147.CHANNEL_ID = T220.CHANNEL_ID
        and T185.PROD_ID = T220.PROD_ID)
 group by T147.CHANNEL_CLASS,
    .....
```

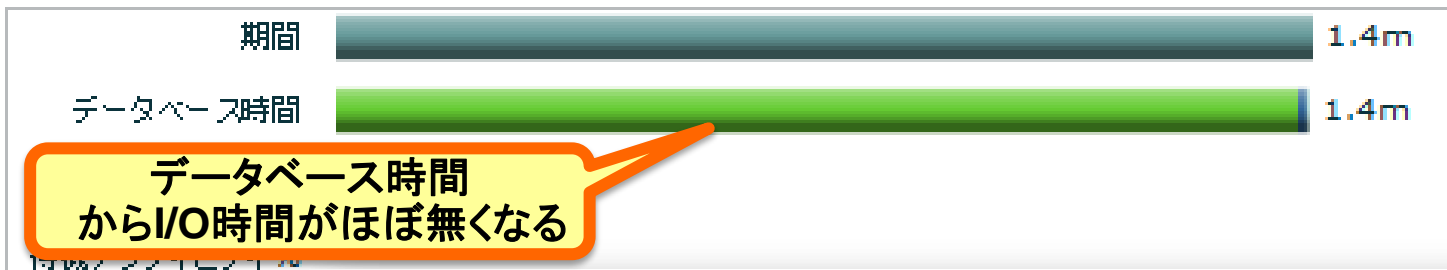
- つまり、データの読み込み量が**1/4**に削減される可能性有り

# CPUバウンドなSQL

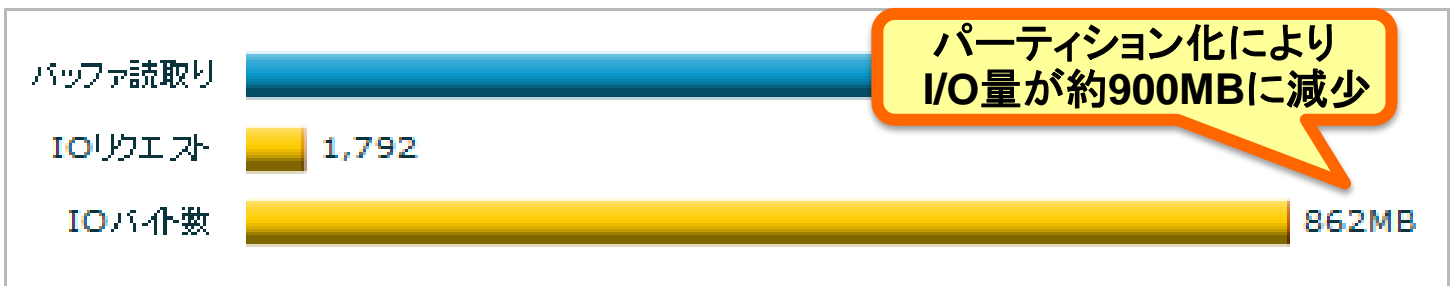
## パーティション化の効果をSQL監視で確認

- 表をパーティション化することで、「I/Oバイト量」が**4GB** から **900MB**に削減されたことを確認
- データベース時間からI/O時間が大幅に削減され、大半がCPU時間へ

### 時間と待機の統計



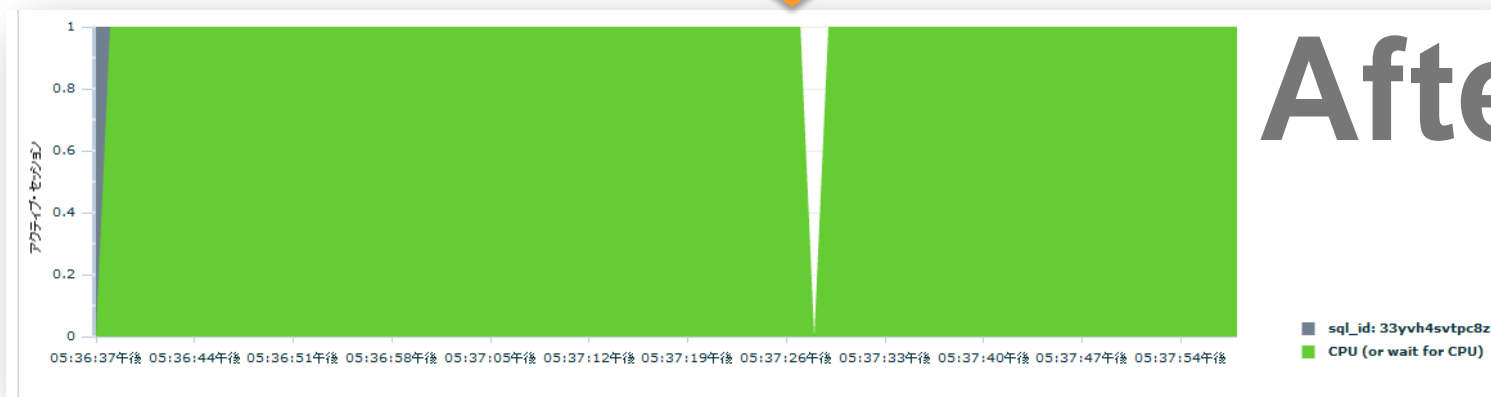
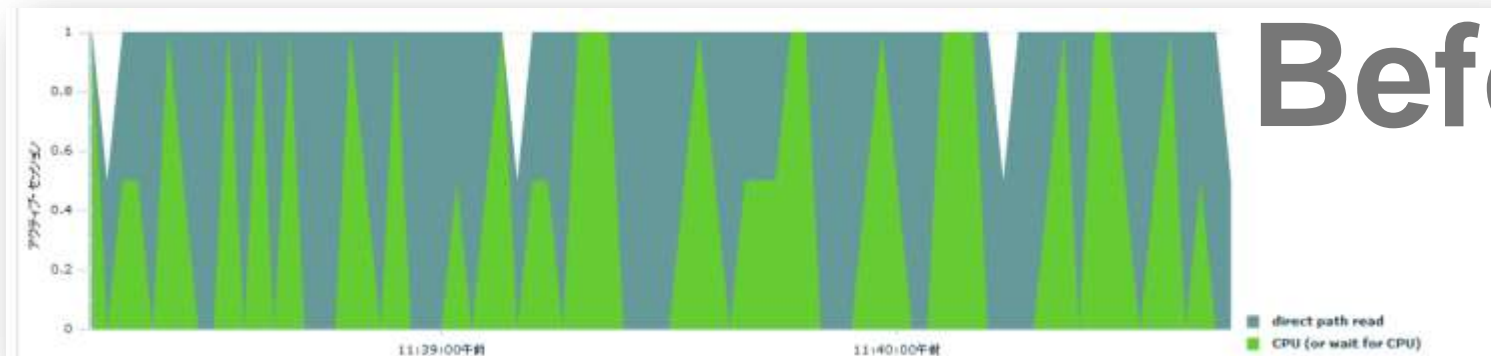
### IO統計



# CPUバウンドなSQL

## パーティション化の効果を確認

- I/Oボトルネックが改善し、CPUを最大限に使用できていることを確認  
→ つまり、CPU（正確には、1CPUコア）がボトルネックとなっている状態



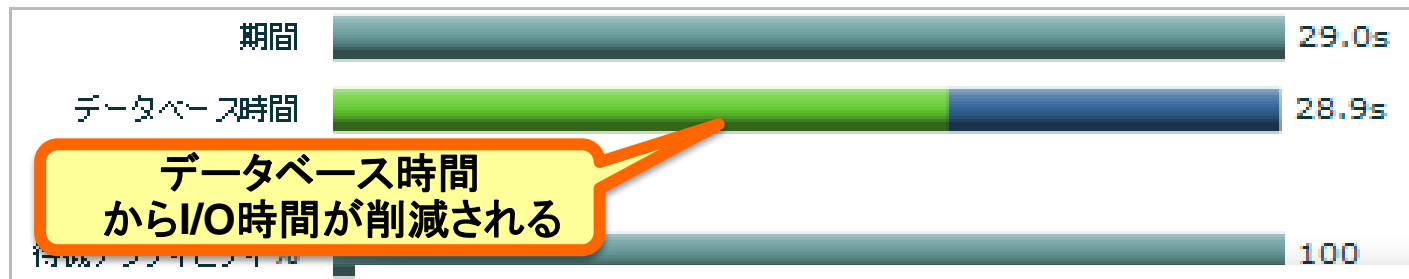


# I/OバウンドなSQL

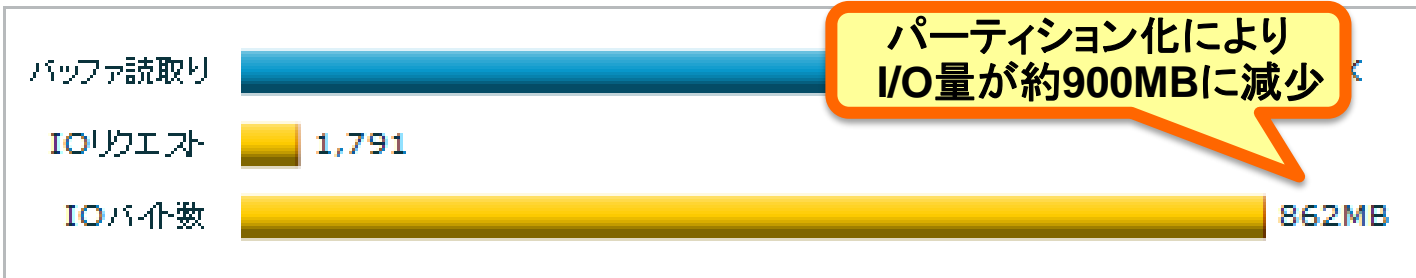
## パーティション化の効果

- 表をパーティション化することで、「I/Oバイト量」が**4GB** から **900MB**に削減されたことを確認
- データベース時間からI/O時間が削減され、CPU時間の割合が増加

### 時間と待機の統計



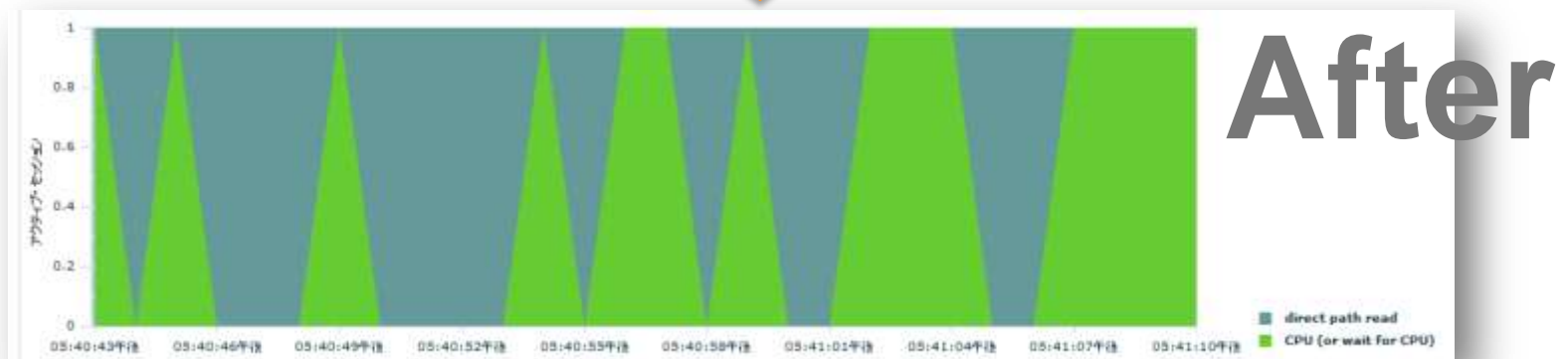
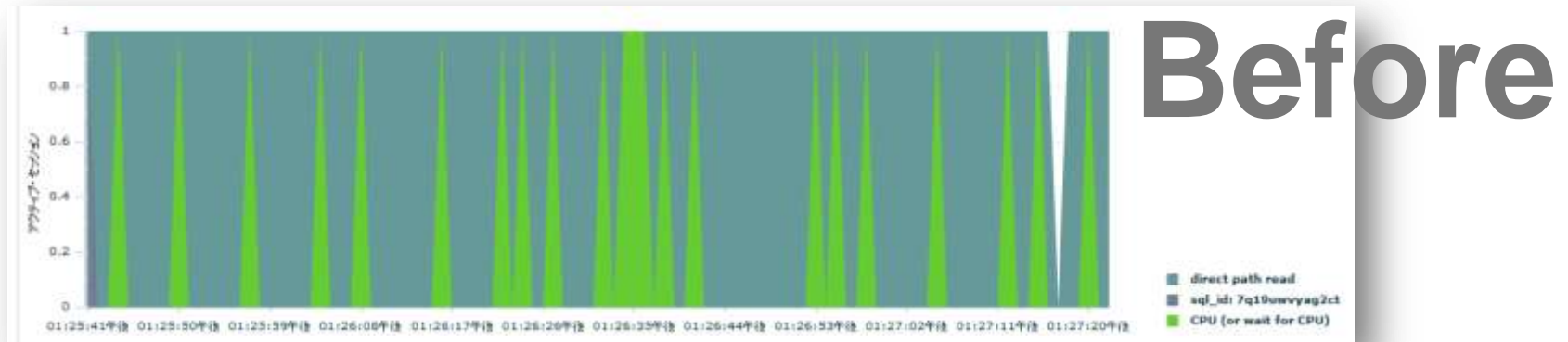
### IO統計



# I/OバウンドなSQL

## パーティション化の効果

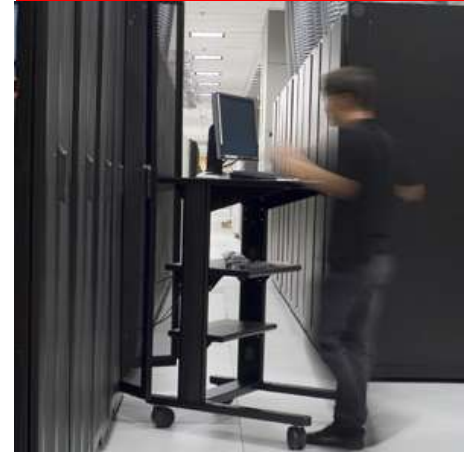
- 「direct path read」待機イベントの時間が、パーティション化を行うことで、大幅に減少していることを確認



# DWH向けチューニング手法

## Summary

適用機能				CPUバウンド	I/Oバウンド
Partitioning					
—	—	—	—	150 sec	102 sec
○	—	—	—	<b>84</b> sec	<b>29</b> sec



## 3. Parallel Queryの適用

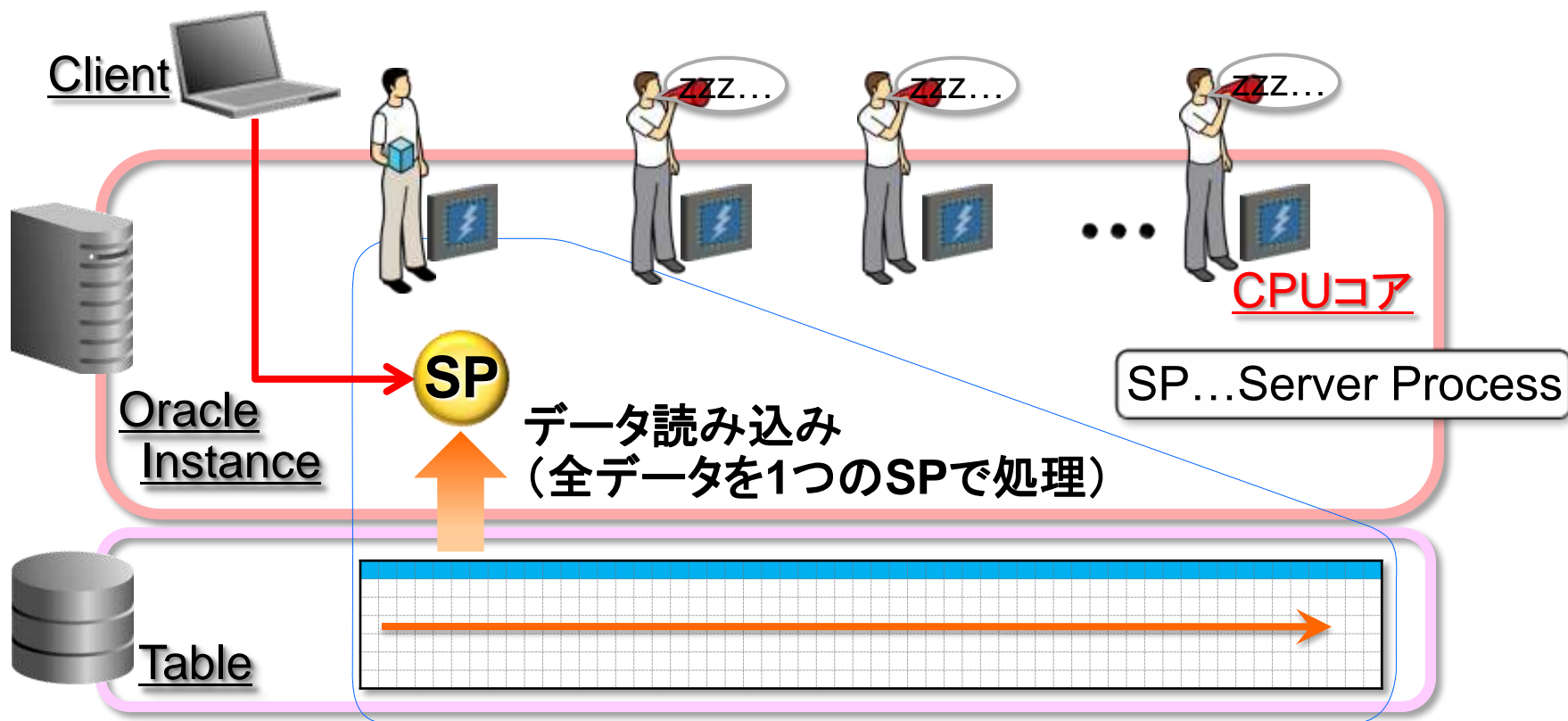
パーティション化することで、1CPUコアのボトルネックになることが確認できました。

次に、「自動パラレル度設定」機能を活用し、1つのSQLの実行時に複数CPUコアを使用するParallel Queryをご紹介します。

# DWHにおけるCPUリソースの使用

## 大量データを集計するようなSQLをシリアル実行した場合

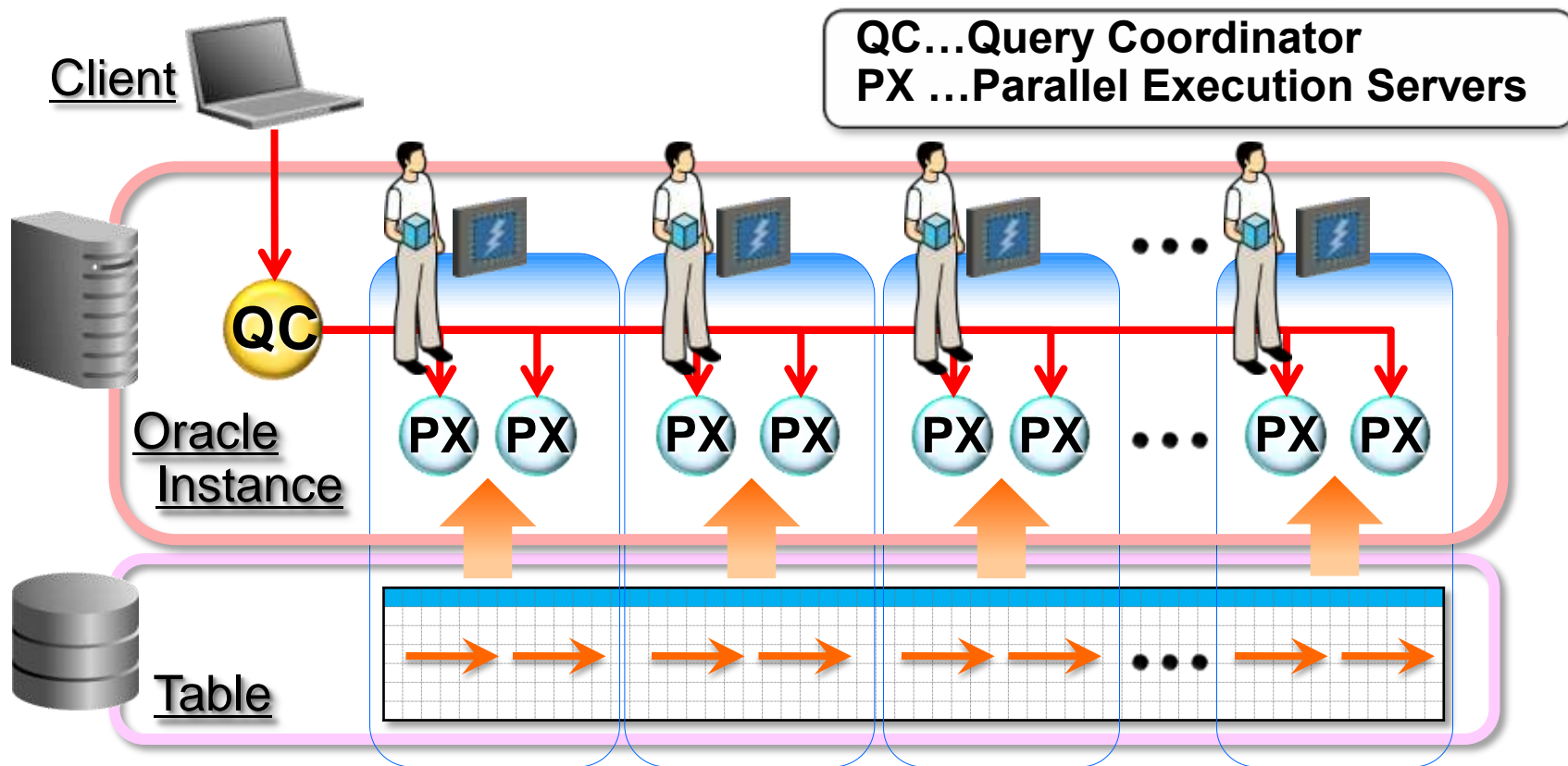
- **Standard Edition**ではSQLをシリアルで実行するため、1つのCPUコアしか使用しない。その為、CPUコアを追加しても性能向上は期待できない



# Parallel実行によるSQLの高速化

## マルチコアの有効活用

- **Enterprise Edition**のParallel実行を利用することで、複数CPUコアを活用し、処理の高速化を実現



# 自動パラレル度設定

## アプリケーションから透過的に使用可能

- Oracle Database 11g Release 2~の新たなパラレル度設定の方法
  - 各SQLの最適なパラレル度を自動的に設定
  - H/Wリソースの有効活用を実現
- パラレル度設定に関する負担を大幅に軽減
  - 初期化パラメータの設定
    - PARALLEL\_DEGREE\_POLICY
  - IO Calibrate statisticsの収集
    - DBMS\_RESOURCE\_MANAGER.CALIBRATE\_IOプロシージャの実行
  - アプリケーション側での設定は不要
    - 従来は、DBAが個別に最適なパラレル度を分析/設定する必要有り

# 自動パラレル度設定

## 設定方法

- PARALLEL\_DEGREE\_POLICY
  - 「LIMITED」もしくは「AUTO」に設定(デフォルト「MANUAL」)

	MANUAL	LIMITED	AUTO
自動パラレル度設定	×	○	○
In-Memory Parallel実行	×	×	○
Parallel Statementキューイング	×	×	○

- alter system文、alter session文での動的変更が可能

```
alter system set parallel_degree_policy=AUTO scope=both;
```

```
alter session set parallel_degree_policy=AUTO;
```



# Parallel実行の確認

## Real-Time SQL Monitoring

- 自動パラレル度設定が機能し、SQLが自動的にパラレル化
  - パラレル列に「2」と表示されていることから判断可能
- ※ アプリケーション側には何も変更していないことが重要



# CPUバウンドなSQL

## 自動パラレル実行の効果

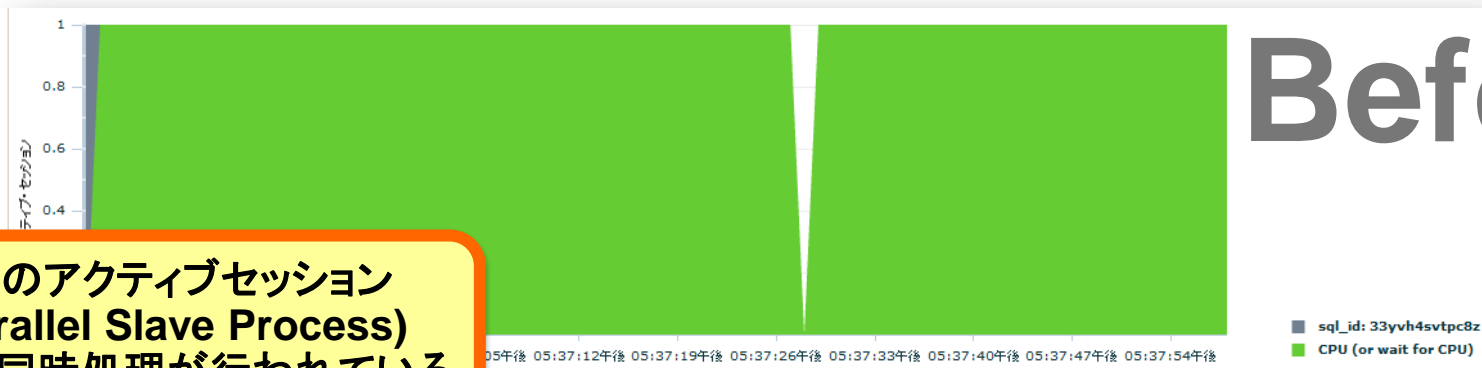
- シリアル実行時と比較し、SQLの実行時間が大幅に改善



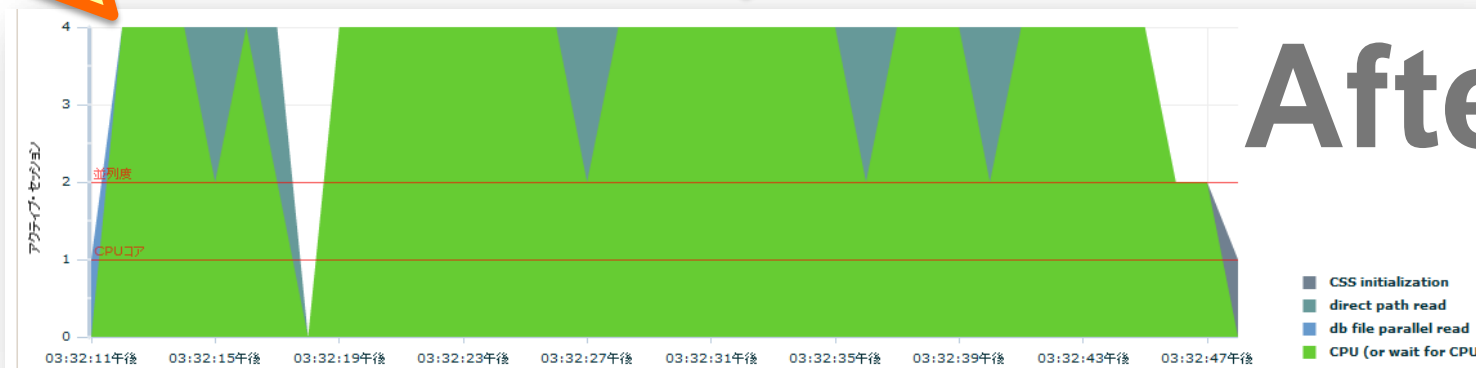
# CPUバウンドなSQL

## 自動パラレル実行の効果

- パラレル実行により、複数のアクティブ・セッション (Parallel Slave Process) が、この1つのSQLを同時に処理していることを確認



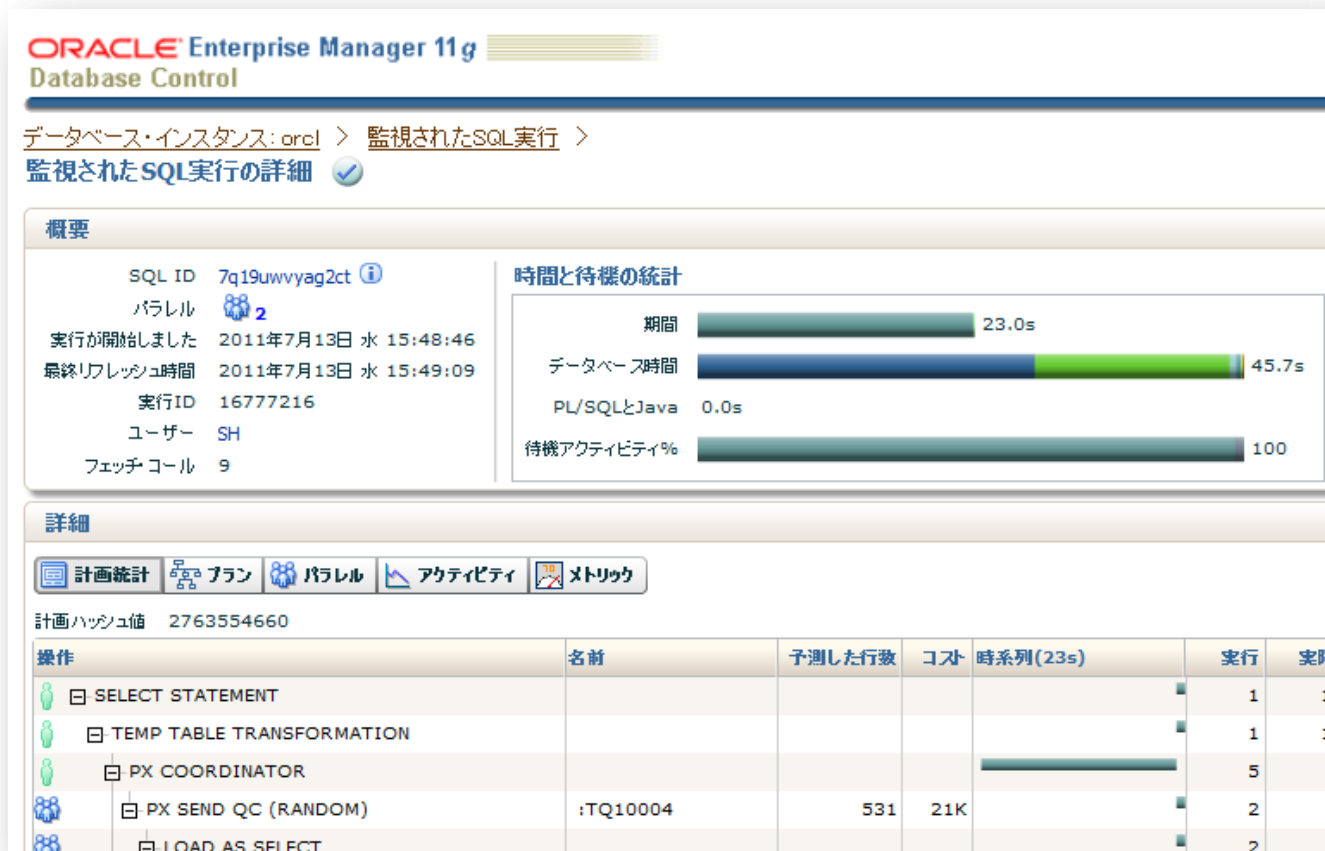
複数のアクティブセッション (Parallel Slave Process) による同時処理が行われている



# I/OバウンドなSQL

## 自動パラレル実行の効果

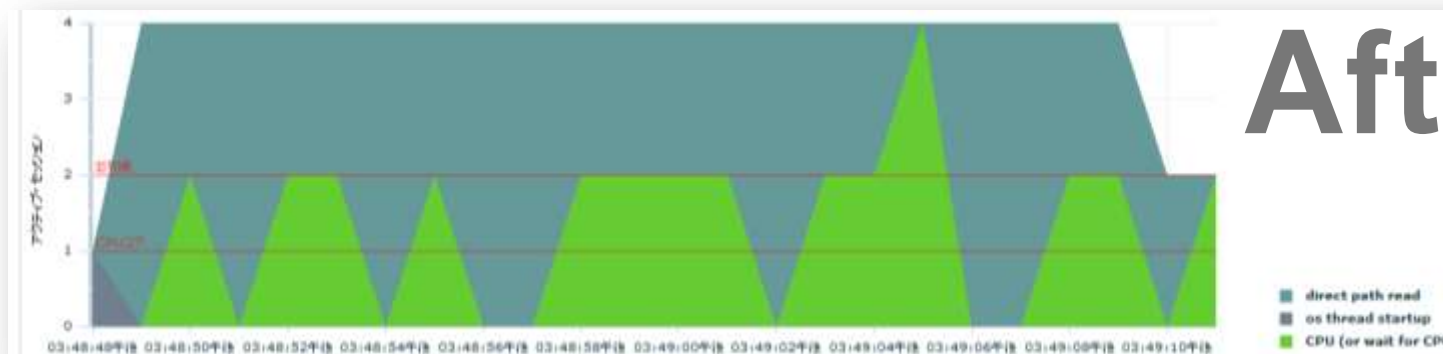
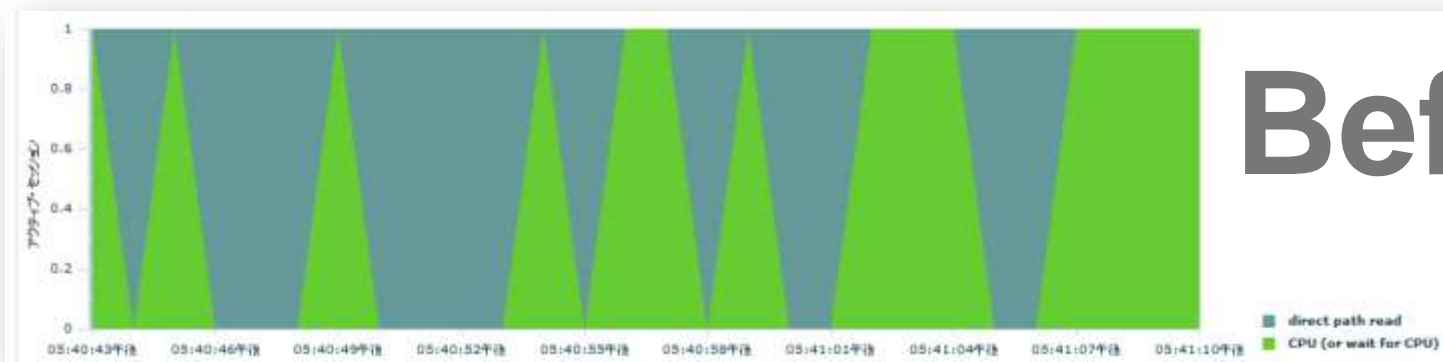
- シリアル実行時と比較し、CPU処理時間が高速化されましたが、I/O性能がボトルネックな処理のため、大幅な高速化はされていない



# I/OバウンドなSQL

## 自動パラレル実行の効果

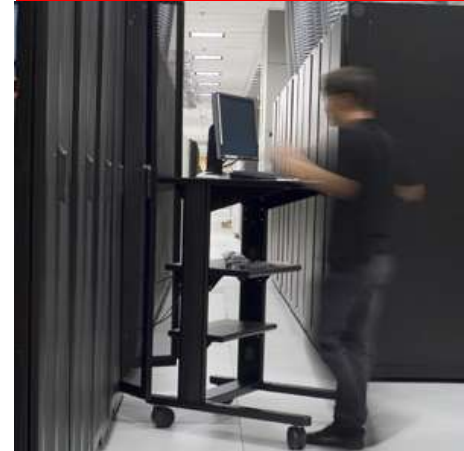
- I/O性能がボトルネックの為、CPU性能を生かし切れていない状態



# DWH向けチューニング手法

## Summary

適用機能				CPUバウンド	I/Oバウンド
Partitioning	Parallel Query				
—	—	—	—	150 sec	102 sec
○	—	—	—	<b>84</b> sec	<b>29</b> sec
○	○	—	—	<b>37</b> sec	



## 4. Oracle Advanced Compressionで 表を圧縮してI/O量を削減

I/OバウンドなSQLの場合、I/O性能がボトルネックとなり、Parallel実行ではレスポンス・タイムが改善しないことが判明しました。

次は、I/O量を削減する目的で、表データをAdvanced Compression Optionの「OLTP表圧縮」機能で圧縮し、パーティション化 + パラレル化 + 圧縮の効果を確認します。

# Advanced Compression Option

## 圧縮機能一覧

- 最大限のリソース活用とコスト削減を支援する包括的な圧縮機能 (Oracle Database 11g~)

### 1. 格納データの圧縮

- ・OLTP表の圧縮
- ・非構造化データ (SecureFiles) の圧縮・重複除外

### 2. バックアップの圧縮

- ・Data Pumpの圧縮
- ・RMANの高速圧縮



### 3. 通信データの圧縮

- ・Data GuardのREDO転送

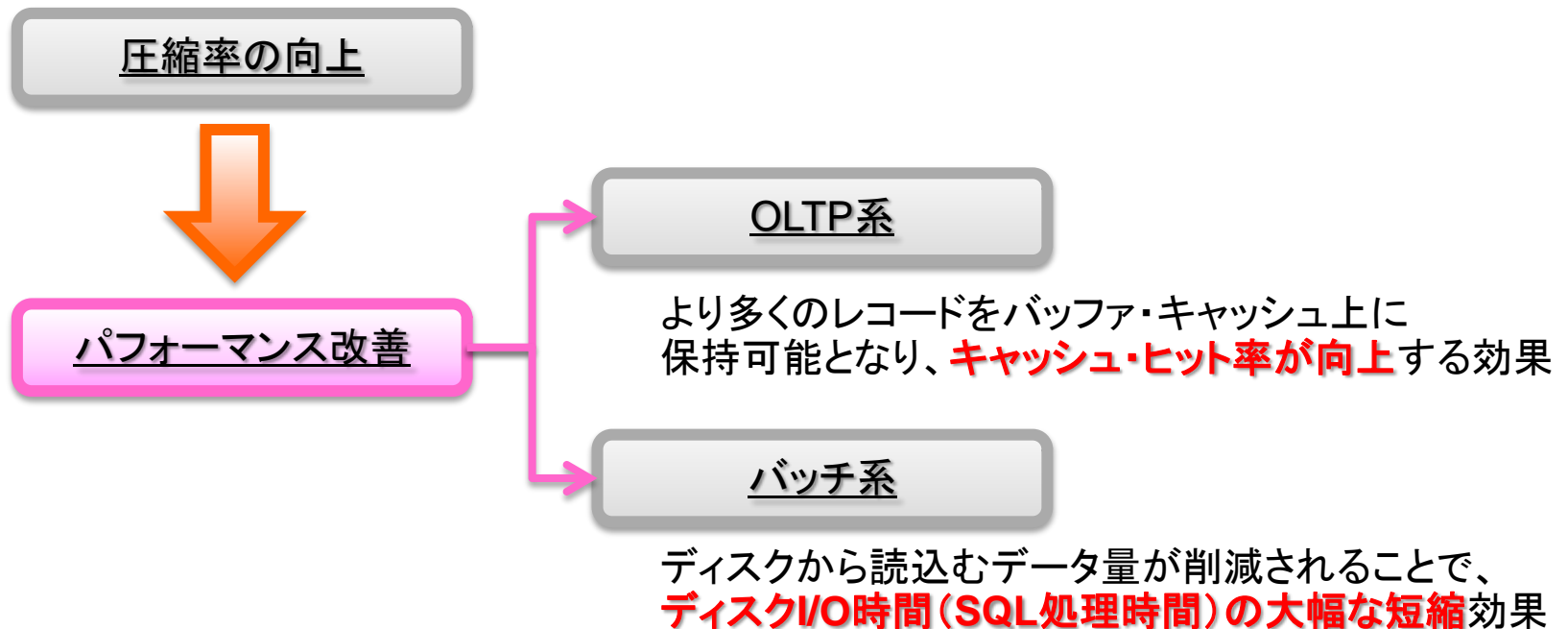




# OLTP表圧縮により期待される効果

## 圧縮率の向上によるパフォーマンス改善

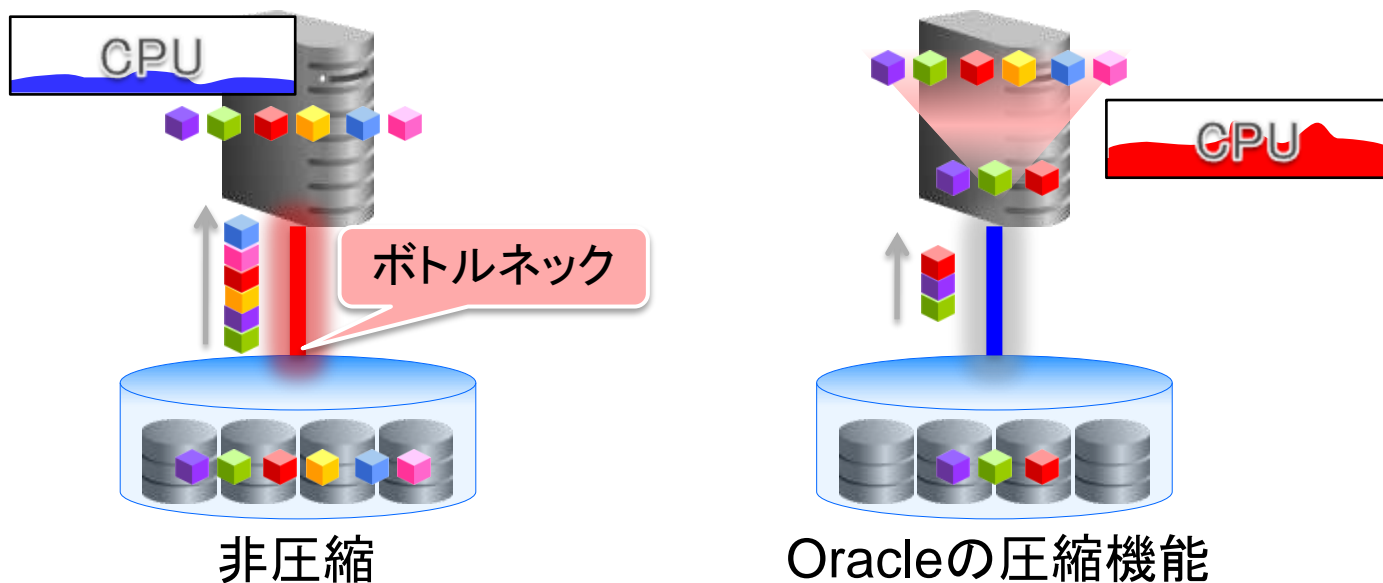
- OLTP表圧縮でデータ量を縮小し、検索系処理のパフォーマンスを改善
  - パフォーマンス改善は、ディスクI/O性能のボトルネックを解消することで実現
  - その場合、CPUリソースが余っていることが前提



# 大量データ読込みの高速化

## ディスクI/O性能のボトルネックの解消

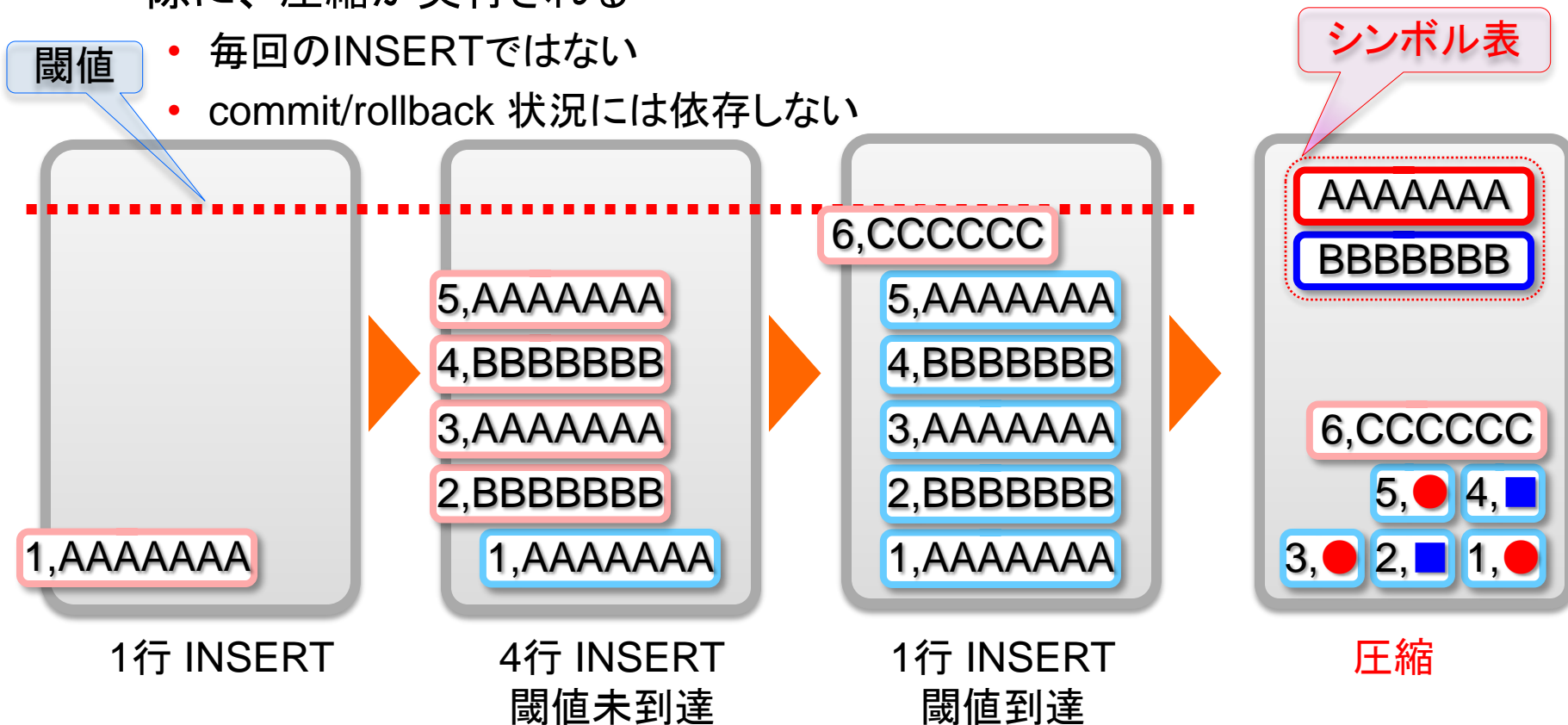
- Oracle Databaseの圧縮機能は、H/Wリソースを有効活用
  - Oracleは、サーバー側で展開する仕組みでボトルネックを解消



# OLTP表圧縮

## データ圧縮のアルゴリズム

- ブロック空き領域が内部的に定められた閾値以下になるINSERTを行った際に、圧縮が実行される
  - 毎回のINSERTではない
  - commit/rollback 状況には依存しない



# 圧縮表の作成方法

## Oracle Database 11g Release 2 以降

- 表領域レベル/表レベル/パーティションレベルでの設定が可能
- 表領域レベル

```
create tablespace TablespaceName datafile '.....'  
                default COMPRESS FOR OLTP;
```

- 表レベル

```
create table TableName (column1,column2,..  
                        COMPRESS FOR OLTP;
```

非圧縮を明示的に指定する場合は、「NOCOMPRESS」

- Oracle Database 11g R1の表記方法は非推奨
  - COMPRESS FOR ALL OPERATIONS => COMPRESS FOR OLTP
  - COMPRESS FOR DIRECT\_LOAD OPERATIONS  
=> COMPRESS または、COMPRESS BASIC

# 圧縮表の作成方法

## Oracle Database 11g Release 2 以降

- パーティションレベル
  - 表全体/親パーティション/サブ・パーティションの単位で設定可能

```
create table TableName (column1,column2,...)
partition by PartitionType (columnM)
subpartition by PartitionType (columnN)
(partition Partition1 values less than (value1)
(subpartition SubPartition1 values (value1-1)
subpartition SubPartition2 values (value1-2),
partition Partition2... ));
```

優先順位

低

高

- 例えば、表全体の設定は「圧縮」にし、特定のパーティションだけ「非圧縮」に設定する場合は、  
表全体で「COMPRESS FOR OLTP」  
特定の親パーティションで「NOCOMPRESS」  
を指定する

# 圧縮表への変更方法

## alter table文と表のオンライン再定義

- 既存表を圧縮属性に変更する方法は主に3種類
- 1. 既存レコードは非圧縮のまま、新規レコードから圧縮する場合

```
alter table TableName COMPRESS FOR OLTP;
```

- 2. 新規レコードだけではなく、既存レコードも圧縮する場合
  - ただし、このSQL終了後、索引のRebuildが必要となる

```
alter table TableName  
MOVE COMPRESS FOR OLTP;
```

- 一定期間の運用後、既に圧縮済みの表の圧縮効率を高める為に、再圧縮のオペレーションとしても利用可能
- 3. 表のオンライン再定義を使用
  - システム無停止で、既存レコードも圧縮可能

# 圧縮表への変更方法

## 表のオンライン再定義のサンプル

```
BEGIN
  DBMS_REDEFINITION.CAN_REDEF_TABLE('SH','SALES',DBMS_REDEFINITION.CONSTRAINT_NAME);
END;
/

create table SALES_TMP compress for oltp as select * from SALES where 1=2;
alter table SALES_TMP add primary key(col1);

BEGIN
  DBMS_REDEFINITION.START_REDEF_TABLE(
    uname => 'SH', orig_table => 'SALES', int_table => 'SALES_TMP',
    col_mapping => NULL, options_flag => DBMS_REDEFINITION.CONSTRAINT_NAME);
END;
/

BEGIN
  DBMS_REDEFINITION.SYNC_INTERIM_TABLE('SH','SALES', 'SALES2');
END;
/

-- このタイミングで必要に応じて、SALES_TMP側に索引を作成(その後、再度SYNC_INTERIM_TABLEの実行を推奨)

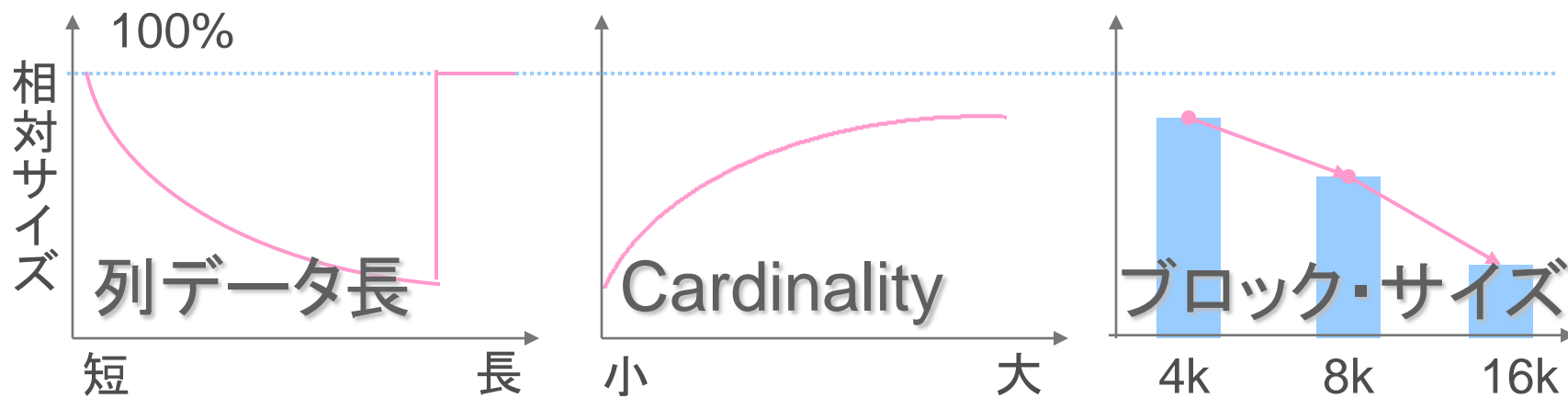
BEGIN
  DBMS_REDEFINITION.FINISH_REDEF_TABLE('SH','SALES', 'SALES2');
END;
/

drop table SALES_TMP;
```

# 圧縮率に影響する3大要素

## 列データ長、Cardinality、ブロック・サイズ

- 列データ長が長いほど、圧縮効果は高い
  - ただし、1ブロック内に重複する列データが2つ格納できない場合は例外
- Cardinality (値の種類) が小さいほど、圧縮効果は高い
- ブロック・サイズが大きいほど、圧縮効果は高い

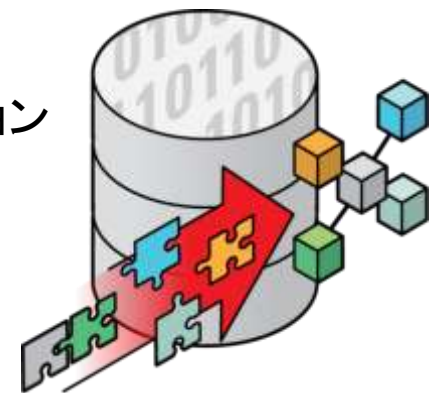




# Advanced Compression Advisor

## Oracle Database 11g Release 2 以降

- DBMS\_COMPRESSION パッケージ
  - DBA権限所有ユーザーでのみ実行可能
- GET\_COMPRESSION\_RATIO プロシージャ
  - 事前に圧縮効果を測定するプロシージャ
  - プロシージャ内部で、実際に圧縮表と非圧縮表を作成
    - Enterprise Editionでのみ使用可能
- GET\_COMPRESSION\_TYPE ファンクション
  - 指定したブロックの圧縮方法を確認できるファンクション



# GET\_COMPRESSION\_RATIO プロシージャ

## 構文

```
set serveroutput on
declare
  SCRATCHTBSNAME          VARCHAR2 (30)      := 'USERS';
  OWNNAME                 VARCHAR2 (30)      := 'SH';
  TABNAME                 VARCHAR2 (30)      := 'SALES';
  PARTNAME                VARCHAR2 (30)      := 'SALES_Q3_2001';
  COMPTYPE_FLG           NUMBER              := 2;
  SAMPLE_BLKCNT_CMP      BINARY_INTEGER;
  SAMPLE_BLKCNT_UNCMP    BINARY_INTEGER;
  SAMPLE_ROWNUM_PER_BLK_CMP  BINARY_INTEGER;
  SAMPLE_ROWNUM_PER_BLK_UNCMP BINARY_INTEGER;
  CMP_RATIO              NUMBER;
  COMPTYPE_STR           VARCHAR2 (100);
begin
  DBMS_COMPRESSION.GET_COMPRESSION_RATIO
    (SCRATCHTBSNAME, OWNNAME, TABNAME, PARTNAME, COMPTYPE_FLG,
     SAMPLE_BLKCNT_CMP, SAMPLE_BLKCNT_UNCMP, SAMPLE_ROWNUM_PER_BLK_CMP,
     SAMPLE_ROWNUM_PER_BLK_UNCMP, CMP_RATIO, COMPTYPE_STR);
  dbms_output.put_line('-----');
  dbms_output.put_line('OBJECT_NAME      => ' || OWNNAME || '.' || TABNAME || ' (PARTITION=' || PARTNAME || ')');
  dbms_output.put_line('COMPRESS_RATIO => ' || CMP_RATIO);
  dbms_output.put_line('-----');
  dbms_output.put_line('COMPRESSED_TYPE                = ' || COMPTYPE_STR);
  dbms_output.put_line('SAMPLE_UNCOMPRESSED_BLOCKS     = ' || SAMPLE_BLKCNT_UNCMP);
  dbms_output.put_line('SAMPLE_COMPRESSED_BLOCKS       = ' || SAMPLE_BLKCNT_CMP);
  dbms_output.put_line('SAMPLE_UNCOMPRESSED_ROWS_PER_BLK = ' || SAMPLE_ROWNUM_PER_BLK_UNCMP);
  dbms_output.put_line('SAMPLE_COMPRESSED_ROWS_PER_BLK  = ' || SAMPLE_ROWNUM_PER_BLK_CMP);
end;
/
```

# GET\_COMPRESSION\_RATIO プロシージャ

## 実行結果

```
-----  
OBJECT_NAME      => SH.SALES (PARTITION=SALES_Q3_2001)  
COMPRESS_RATIO => 2.6  
-----  
  
COMPRESSED_TYPE           = "Compress For OLTP"  
SAMPLE_UNCOMPRESSED_BLOCKS = 318  
SAMPLE_COMPRESSED_BLOCKS  = 123  
SAMPLE_UNCOMPRESSED_ROWS_PER_BLK = 206  
SAMPLE_COMPRESSED_ROWS_PER_BLK  = 535
```

- SH.SALESテーブルのSALES\_Q3\_2001パーティションに対して、OLTP表圧縮を適用した場合、約2.6倍圧縮される見込みであることを確認
  - 約2.6倍圧縮 → データ量が約38%(=100/2.6)まで縮小
- SAMPLE\_UNCOMPRESSED\_BLOCKS : サンプルングしたブロック数
- SAMPLE\_COMPRESSED\_BLOCKS : サンプルングしたブロック数を圧縮後のブロック数
- SAMPLE\_UNCOMPRESSED\_ROWS\_PER\_BLK : サンプルングした1ブロックあたりの行数
- SAMPLE\_COMPRESSED\_ROWS\_PER\_BLK : サンプルングデータを圧縮後の1ブロックあたりの行数

# CPUバウンドなSQLの場合

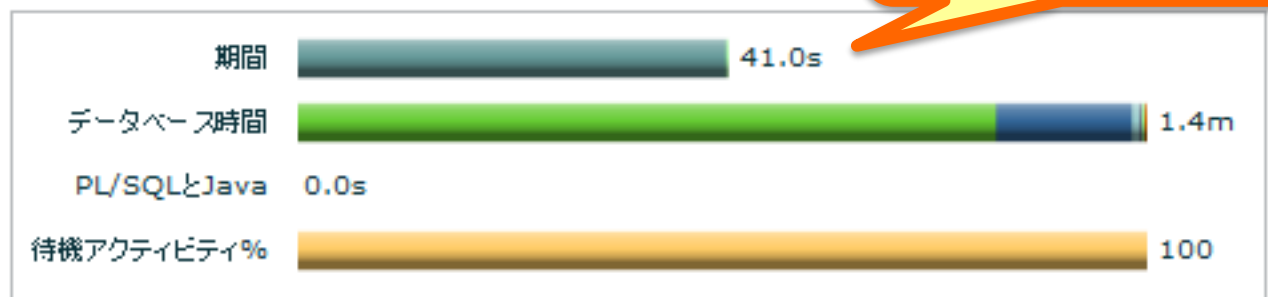
## 圧縮の効果

- OLTP表圧縮によって、I/O量が900MBから300MBに削減
- しかし、CPUがボトルネックのため、処理時間の大幅な改善は無し

### IO統計



### 時間と待機の統計



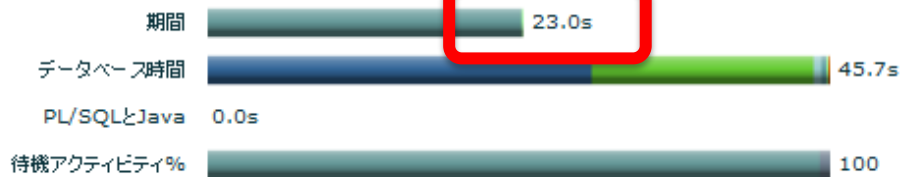
# I/OバウンドなSQLの場合

## 圧縮の効果

- OLTP表圧縮によって、I/O量が900MBから300MBに削減  
→ I/Oボトルネックが改善され、処理速度が高速化

# Before

時間と待機の統計



IO統計

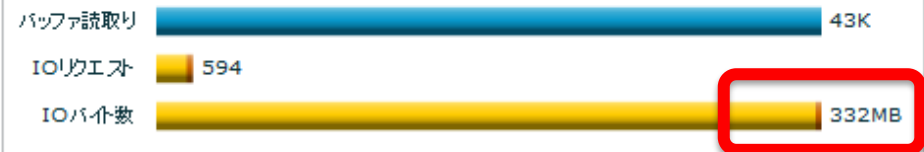


# After

時間と待機の統計



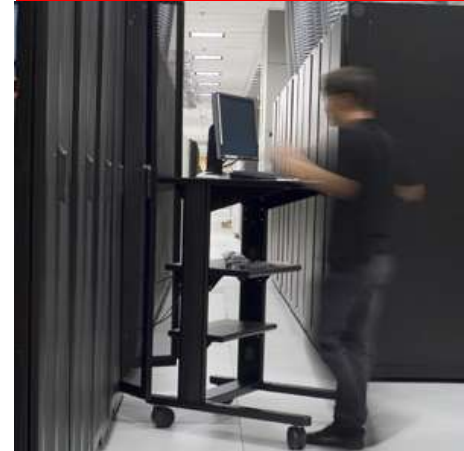
IO統計



# DWH向けチューニング手法

## Summary

適用機能				CPUバウンド	I/Oバウンド
Partitioning	Parallel Query	Compression			
—	—	—	—	150 sec	102 sec
○	—	—	—	<b>84</b> sec	<b>29</b> sec
○	○	—	—	<b>37</b> sec	
○	○	○	—		<b>18</b> sec



## 5. In-Memory Parallel Executionの効果

Oracle 11g Enterprise Editionの新機能である、In-Memory Parallel Execution(In-Memory PX) を用いることで、HDDからのブロック読み込みを極小化 + CPU性能を最大限活用することで、SQLを高速化を実現します。

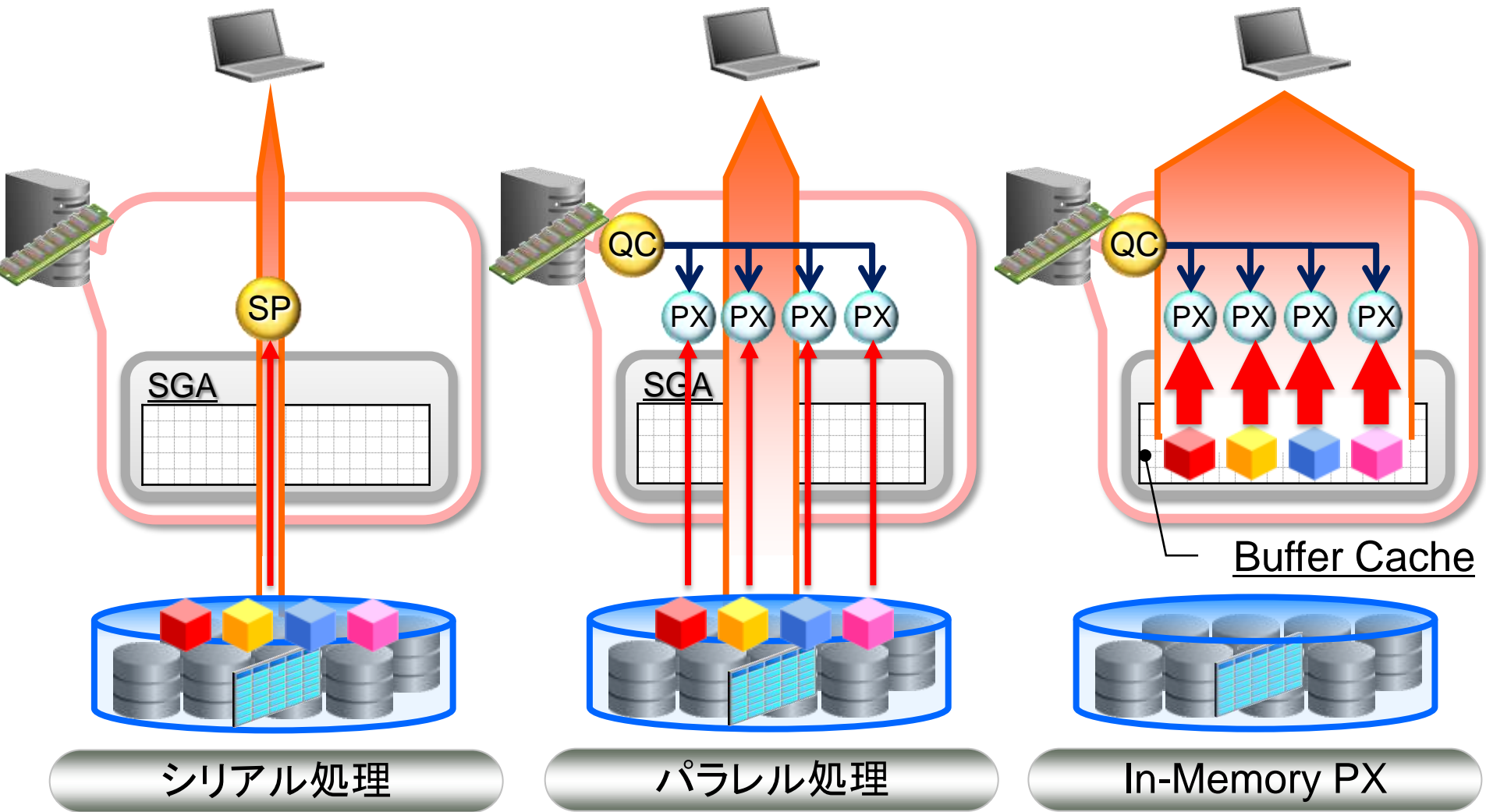
Partitioning + Compression + In-Memory PXの効果とは？





# バッチ&DWH処理の高速化ソリューション

マルチコアCPUの処理能力を最大限に活用



ORACLE

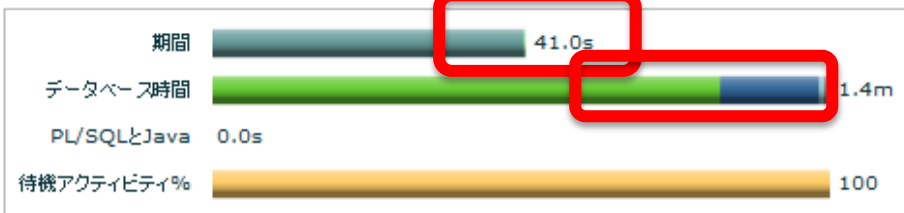
# CPUバウンドなSQL

## In-Memory PXの効果

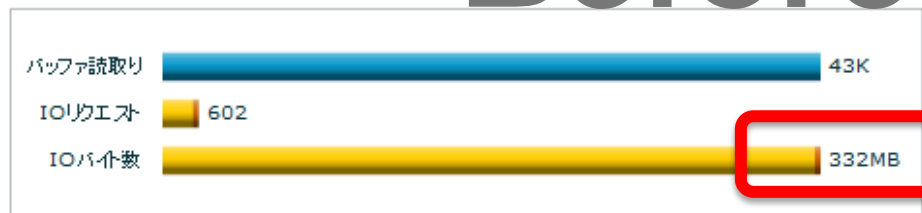
- メモリ上のデータで平行実行が行われていることで、I/O量がほとんどなくCPU時間で占められている状態
- しかし、元々CPUボトルネックな為、処理時間の改善率は小さい

# Before

時間と待機の統計

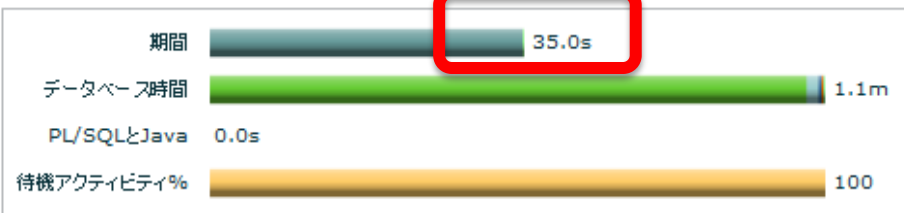


IO統計



# After

時間と待機の統計



IO統計



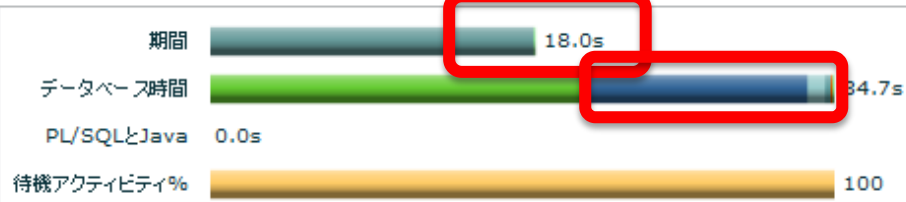
# I/OバウンドなSQLの場合

## In-Memory PXの効果

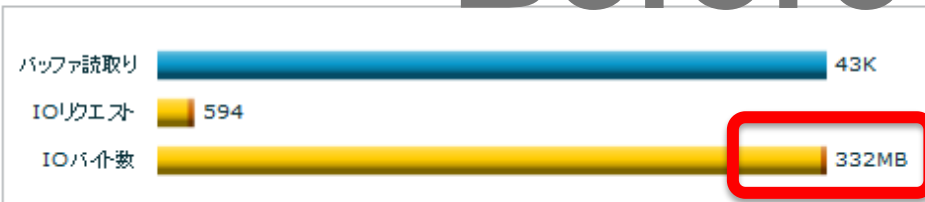
- メモリ上のデータでパラレル実行が行われていることで、I/O量がほとんどなく大半がCPU時間で占められている状態
- In-Memory PXにより、I/Oボトルネックが解消してCPU性能を限界まで使用

# Before

### 時間と待機の統計

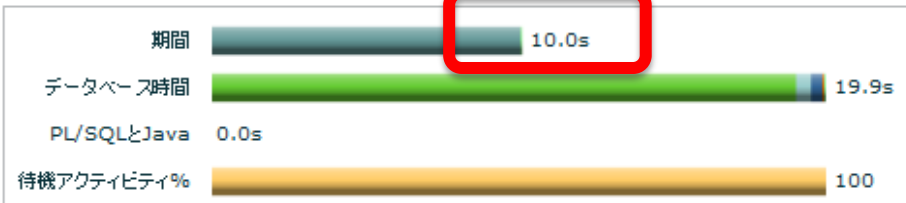


### IO統計

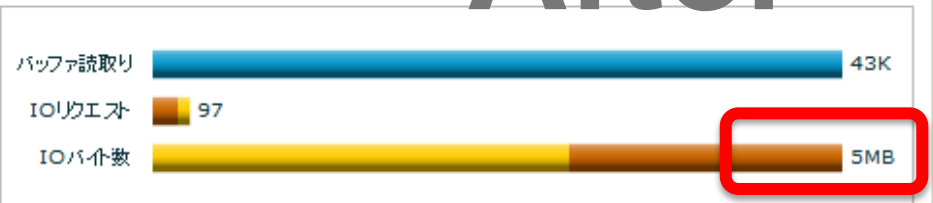


# After

### 時間と待機の統計



### IO統計



# DWH向けチューニング手法

## Summary

適用機能				CPUバウンド	I/Oバウンド
Partitioning	Parallel Query	Compression	In-Memory PX		
—	—	—	—	150 sec	102 sec
○	—	—	—	<b>84</b> sec	<b>29</b> sec
○	○	—	—	<b>37</b> sec	
○	○	○	—		<b>18</b> sec
○	○	○	○		<b>10</b> sec

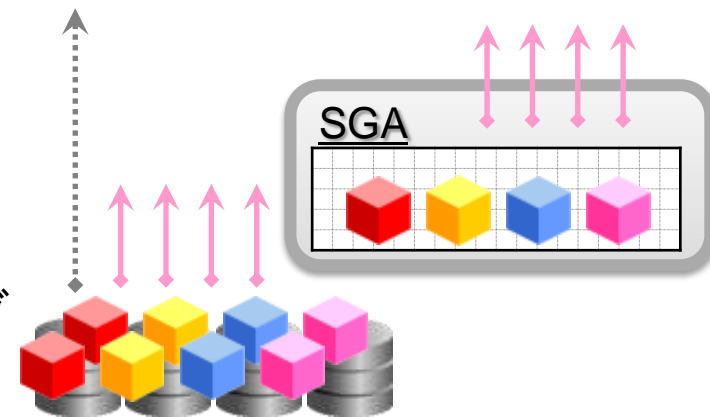
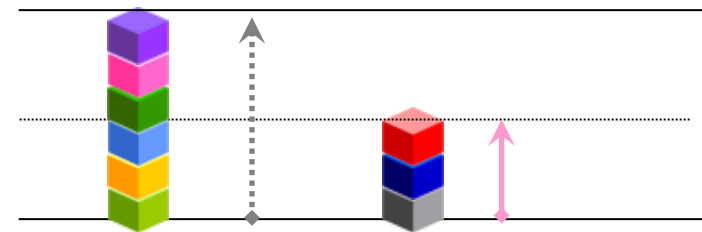
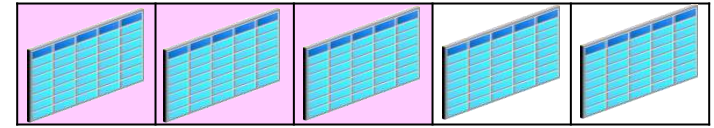


# DWH向けチューニング手法のまとめ

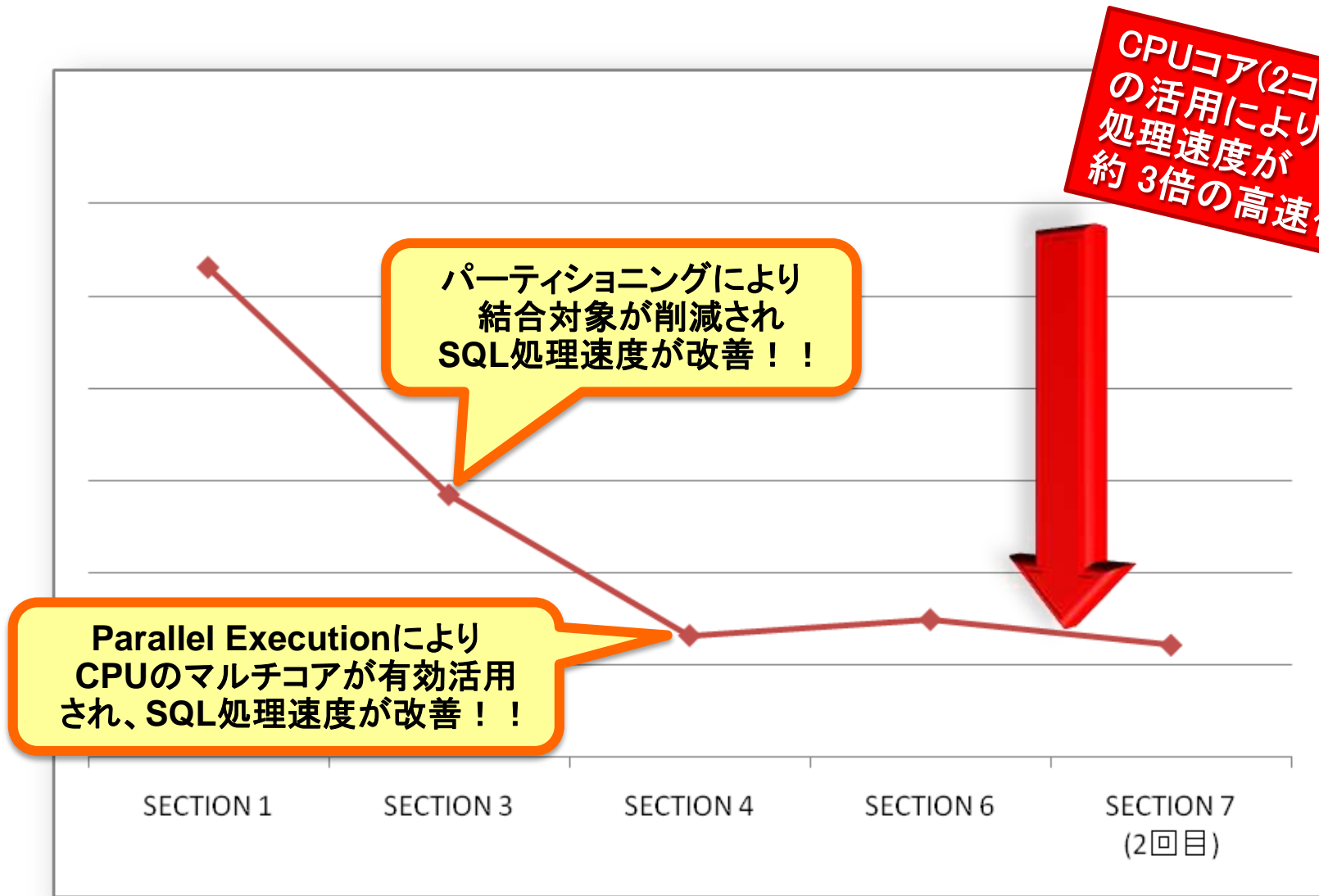
# DWH向けチューニング手法

## まとめ

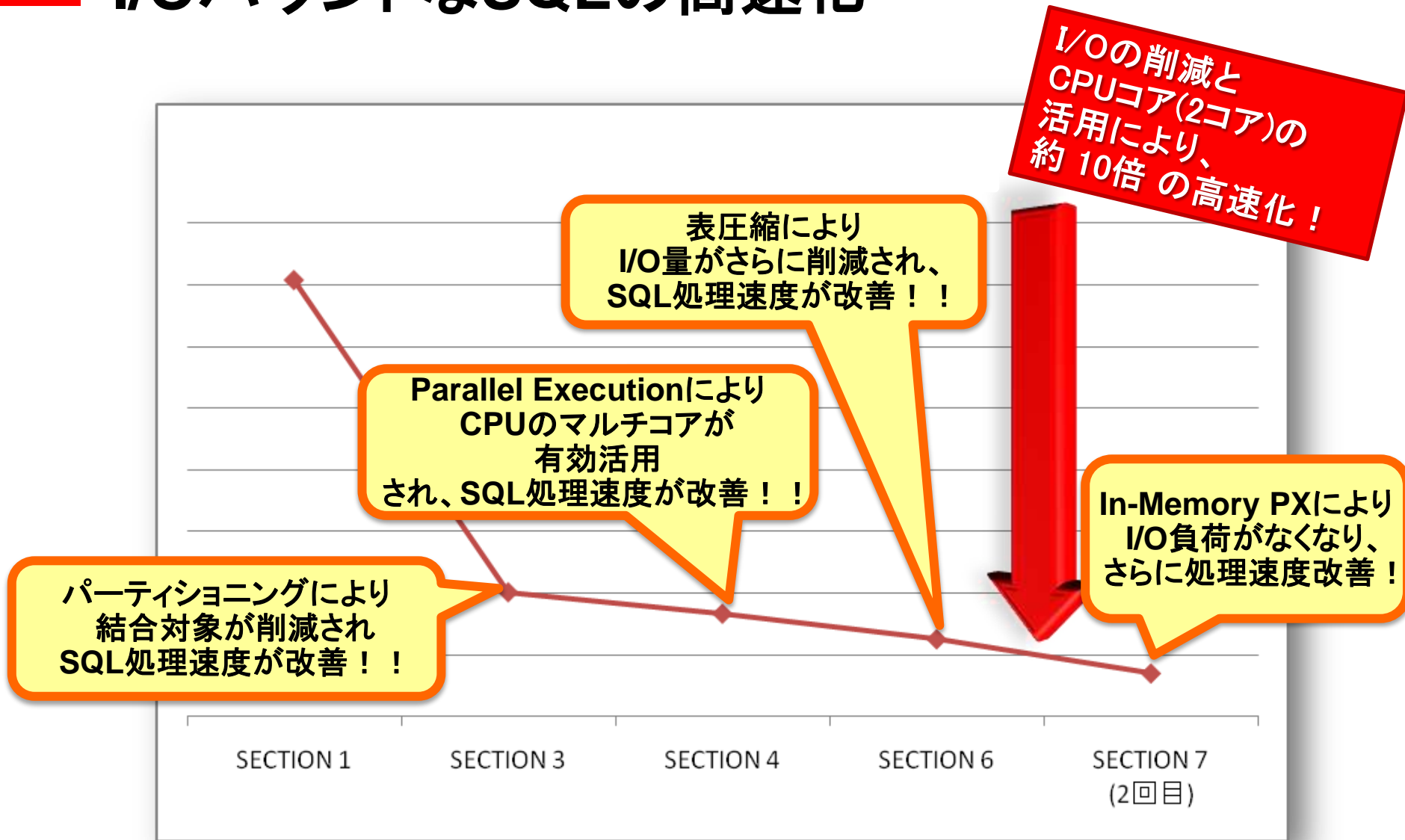
- Oracle Partitioning Option
  - データアクセス範囲の限定
  - 無駄な処理を削減し、CPU及びI/Oコストの軽減
- Oracle Advanced Compression Option
  - データ圧縮
  - I/Oボトルネックの改善
- Parallel Query
  - マルチコアの有効活用
  - 1CPUコアのボトルネックを排除
- In-Memory Parallel Execution
  - 複数CPUの活用 + 大量データのキャッシング
  - I/Oボトルネックを改善



# CPUバウンドなSQLの高速化



# I/OバウンドなSQLの高速化





# OTNセミナーオンデマンド

コンテンツに対する  
ご意見・ご感想を是非お寄せください。

OTNオンデマンド 感想



[http://blogs.oracle.com/oracle4engineer/entry/otn\\_ondemand\\_questionnaire](http://blogs.oracle.com/oracle4engineer/entry/otn_ondemand_questionnaire)

上記に簡単なアンケート入力フォームをご用意しております。

セミナー講師/資料作成者にフィードバックし、  
コンテンツのより一層の改善に役立てさせていただきます。

是非ご協力をよろしくお願いいたします。

# OTNセミナーオンデマンド

日本オラクルのエンジニアが作成したセミナー資料・動画ダウンロードサイト

## 掲載コンテンツカテゴリ(一部抜粋)

Database 基礎

Database 現場テクニック

Database スペシャリストが語る

Java

WebLogic Server/アプリケーション・グリッド

EPM/BI 技術情報

サーバー

ストレージ



超入門! Oracle データベースって何

再生時間: 60分

100以上のコンテンツをログイン不要でダウンロードし放題

データベースからハードウェアまで充実のラインナップ

毎月、旬なトピックの新作コンテンツが続々登場

## 例えばこんな使い方

- 製品概要を効率的につかむ
- 基礎を体系的に学ぶ/学ばせる
- 時間や場所を選ばず(オンデマンド)に受講
- スマートフォンで通勤中にも受講可能



毎月チェック!



コンテンツ一覧 はこちら

<http://www.oracle.com/technetwork/jp/ondemand/index.html>

新作&おすすめコンテンツ情報 はこちら

<http://oracletech.jp/seminar/recommended/000073.html>

OTNオンデマンド



# オラクルエンジニア通信

オラクル製品に関わるエンジニアの方のための技術情報サイト

## オラクルエンジニア通信 - 技術資料、マニュアル、セミナー

Oracleエンジニアのための技術情報サイト by Oracle Japan

新着情報を知りたい

技術資料を探したい

セミナーを受けたい

**About**

Oracleエンジニアの方がスキルアップしていただくために、厳選した情報をお届けしています

技術資料



インストールガイド・設定チュートリアルetc. 欲しい資料への最短ルート

特集テーマ  
Pick UP



性能管理やチューニングなど月間テーマを掘り下げて詳細にご説明

アクセス  
ランキング



他のエンジニアは何を見ているのか？人気資料のランキングは毎月更新

技術コラム



SQLスクリプト、索引メンテナンスetc. 当たり前運用/機能が見違える!?

<http://blogs.oracle.com/oracle4engineer/>

オラクルエンジニア通信



The screenshot shows the top section of the oracletech.jp website. On the left is the 'oracletech.jp' logo with the tagline '好奇心が、エンジニア人生を豊かにする。'. On the right is the 'ORACLE' logo, a search bar, and social media icons for Twitter, Facebook, Ustream, YouTube, and RSS. Below these is a red navigation bar with five buttons: '製品/技術情報', 'スキルアップ', 'セミナー', 'キャンペーン', and 'ちょっと一息'.

製品/技術  
情報



Oracle Databaseっていくら？オプション機能も見積れる簡単ツールが大活躍

セミナー



基礎から最新技術までお勧めセミナーで自分にあった学習方法が見つかる

スキルアップ



ORACLE MASTER ! 試験頻出分野の模擬問題と解説を好評連載中

Viva!  
Developer



全国で活躍しているエンジニアにスポットライト。きらりと輝くスキルと視点を盗もう

<http://oracletech.jp/>

oracletech



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

Oracle Direct



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  
ご登録の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

# **Hardware and Software Engineered to Work Together**

**ORACLE®**