ORACLE® アシスト

夜な夜な! なにわオラクル塾
Presented By アシスト #14

入門！Oracleデータベース・解決パフォーマンスチューニング

目次

アシスト

- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

本セミナーのゴール

アシスト

- Oracleデータベースのチューニングの概要と全体的なイメージをつかむ
- ボトルネックやチューニング手法にはどのような種類があるのか、それぞれの特徴を把握する

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

1. Why ～なぜ実施するの？～

アシスト

理由：パフォーマンスが遅いから

・アプリケーションのレスポンスが遅い

・バッチ処理のスループットが遅い　etc.,

その遅さはどの程度問題ですか？

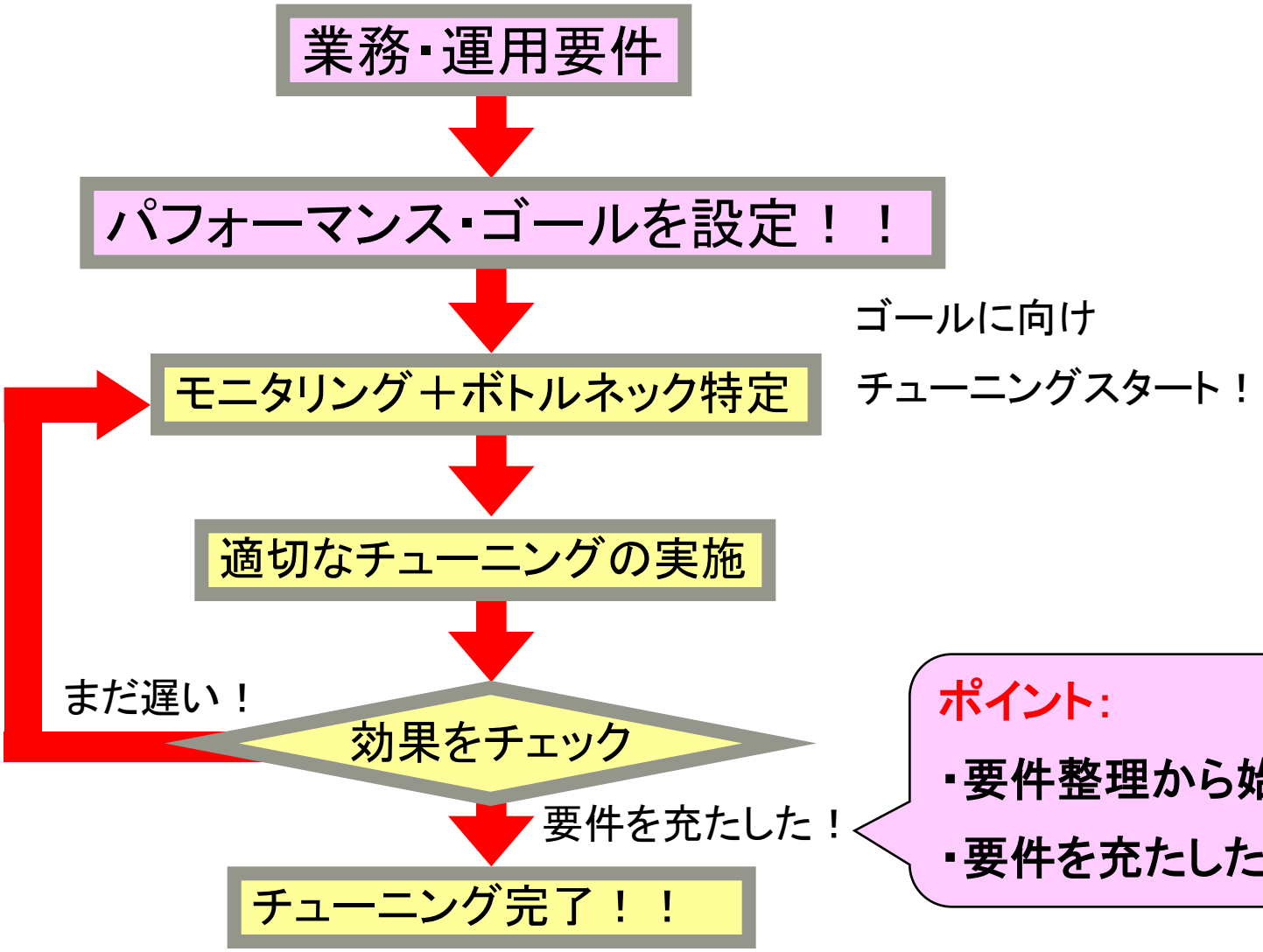
- システムの業務・運用要件（3秒以内にデータ返す、夜間バッチを朝7:00までに完了する etc..）を確認して問題があればチューニングの必要あり！

・・・Goalを明確にして実施しましょう

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

1. Why ～なぜ実施するの？～



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

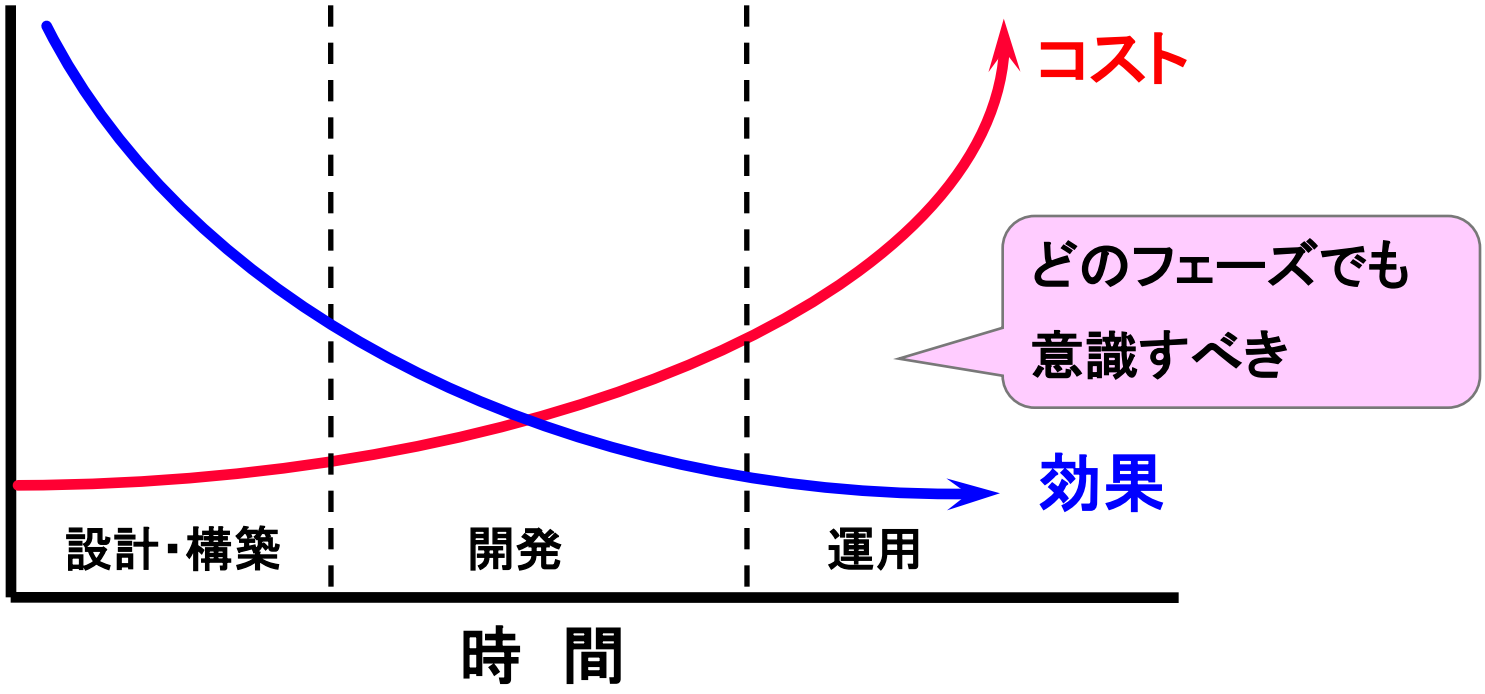
- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

2. When ～いつ実施するの？～

アプリケーションの設計から運用までの間の
チューニングにかかるコストと効果



できるだけ早い時期にチューニングのことを考えた設計、開発を行うことが大切

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

チューニングの対象と役割分担

アシスト

ビジネス・アナリスト	✓業務機能のチューニング
設計者	✓データ設計のチューニング ✓プロセス設計のチューニング
アプリケーション開発者	✓SQL文のチューニング ✓物理構造のチューニング
データベース管理者	✓メモリー割当てのチューニング ✓I/Oのチューニング ✓メモリー競合のチューニング
OS管理者	✓OSのチューニング
ネットワーク管理者	✓ネットワークのチューニング

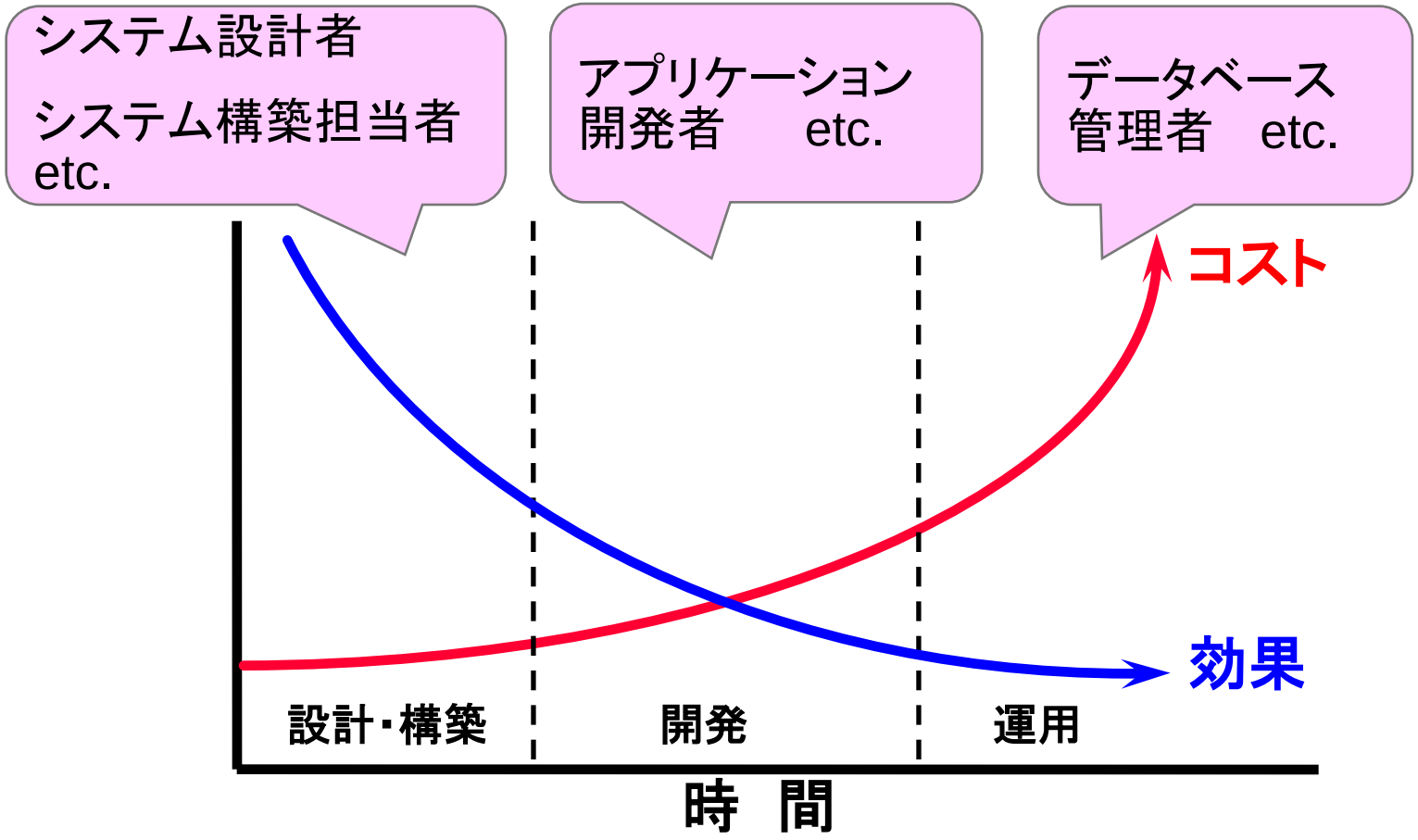
ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

3. Who ～誰が実施するの？～

アシスト

- 各フェーズでそれぞれの担当者が実施すべきです



ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

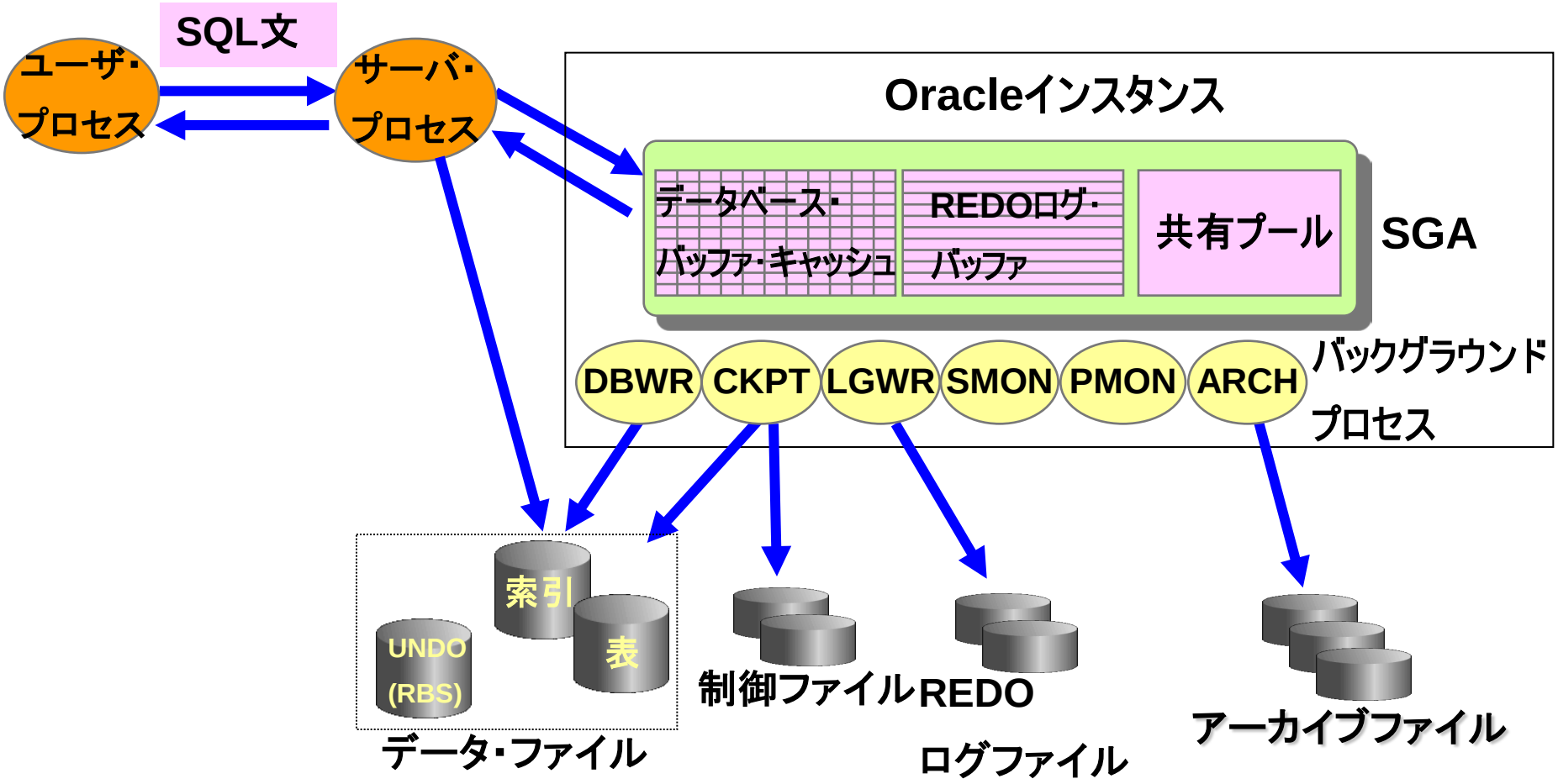
- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

アーキテクチャの復習 ダイジェスト

詳しくは「ここからはじめる、Oracle データベース入門・アーキテクチャー編」をご受講ください！



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

<ご参考>アーキテクチャの復習 ダイジェスト

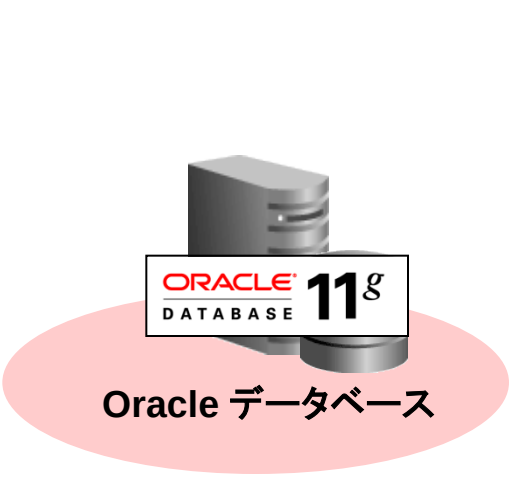
■ユーザ・プロセス：Oracleデータベースに接続し、処理要求(SQL文発行)を行うクライアント・プログラム
■サーバ・プロセス：ユーザ・プロセスからの要求を受付け、処理を行うプロセス
■システム・グローバル領域(SGA)
・データベース・バッファ・キャッシュ：データ・ファイルをコピーし、検索および加工を行うための領域
・REDOログ・バッファ：Redoログ・ファイルにLGWRが書き込む前に、一時的に履歴が置かれる領域
・共有プール：SQL文の情報(実行計画(アクセス・パス)等)や、データ・ディクショナリ情報(DB管理情報)が置かれる領域
■バックグラウンド・プロセス
・DBWR：データベース・バッファ・キャッシュ上で更新されたデータを定期的にデータ・ファイルに書き込む
・LGWR：Redoログ・バッファ上のログデータを定期的にRedoログファイルに書き込む
・CKPT：チェックポイント発生時に制御ファイルやデータ・ファイルのヘッダを更新し同期をとる
・SMON：インスタンス障害後、再起動の際にインスタンス・リカバリを行う
・PMON：サーバ・プロセスの監視を行う
・ARCH：アーカイブ・ログ・モードにおいて、Redoログ・ファイルをアーカイブRedoログ・ファイルにコピーする
■Oracleを構成する物理ファイル
・データ・ファイル：表領域に対応し、ユーザ・データや管理情報を格納
・Redoログ・ファイル：データベース上で行われた変更履歴を格納 (障害時にリカバリ処理で使用される)
・制御ファイル：データベースの現在の状態やファイル群の物理構成を記録
・アーカイブRedoログ・ファイル：Redoログ・ファイルのコピー
・初期化パラメータ・ファイル：SGAの割り当てなどをパラメータで指定する、Oracleの設定ファイル
・サーバー・パラメータ・ファイル：初期化パラメータ・ファイルのバイナリ版。稼動中に変更したパラメータを記録できる(9i以降はこちらが推奨)
・パスワード・ファイル：リモートでDBの起動/停止を行えるようにするためのユーザ名/パスワードを記録

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

監視の目的

アシスト

- パフォーマンス劣化をできるかぎり早期に検出し、問題が深刻化する前に対策が取れるようにします
- システム全体をスローダウンさせかねないリソースの圧迫を、できる限り事前に検知します



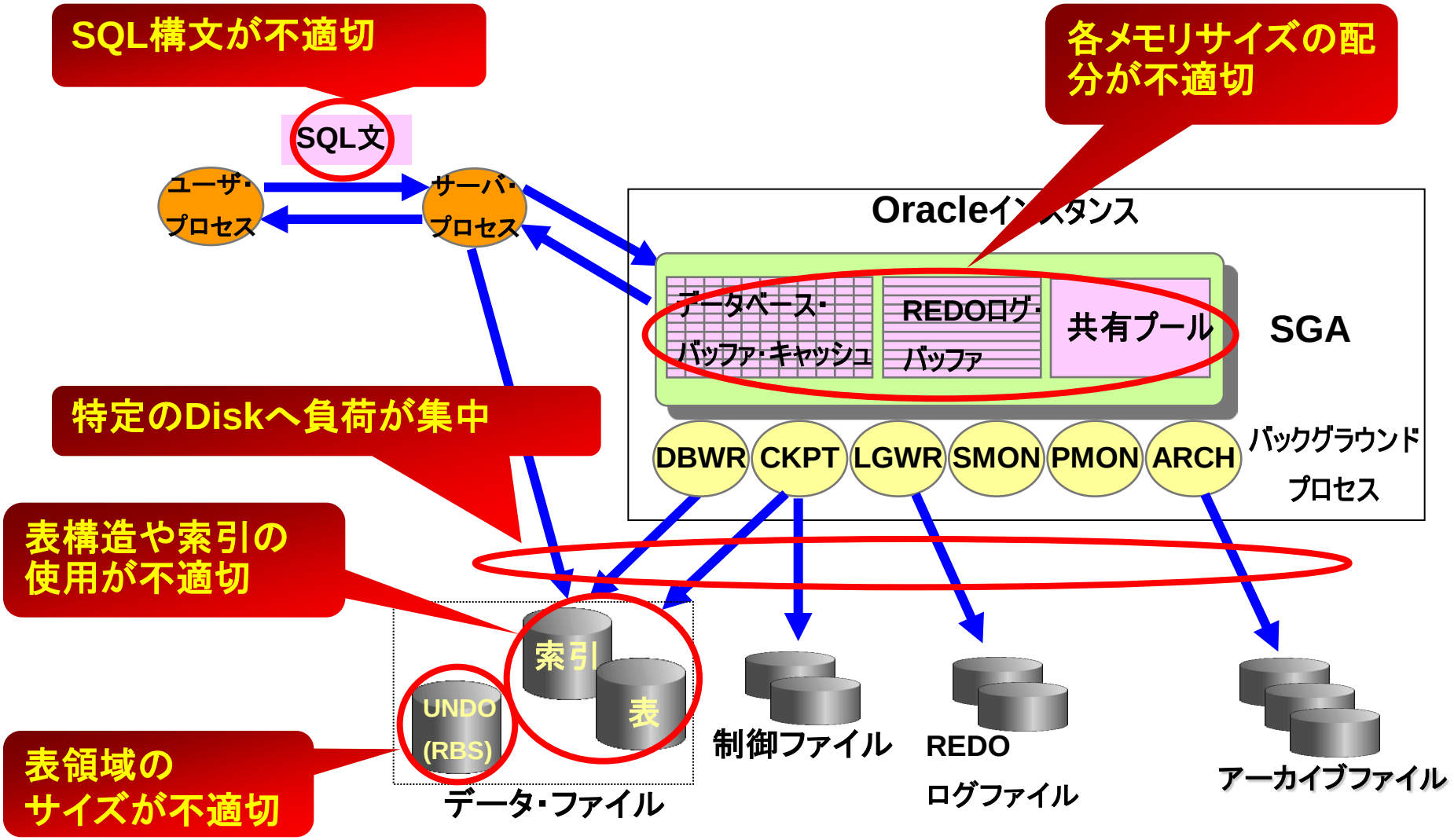
システムの安定稼動・
ダウンタイムの最小化

問題が深刻化する前に
対策を取る

ORACLE

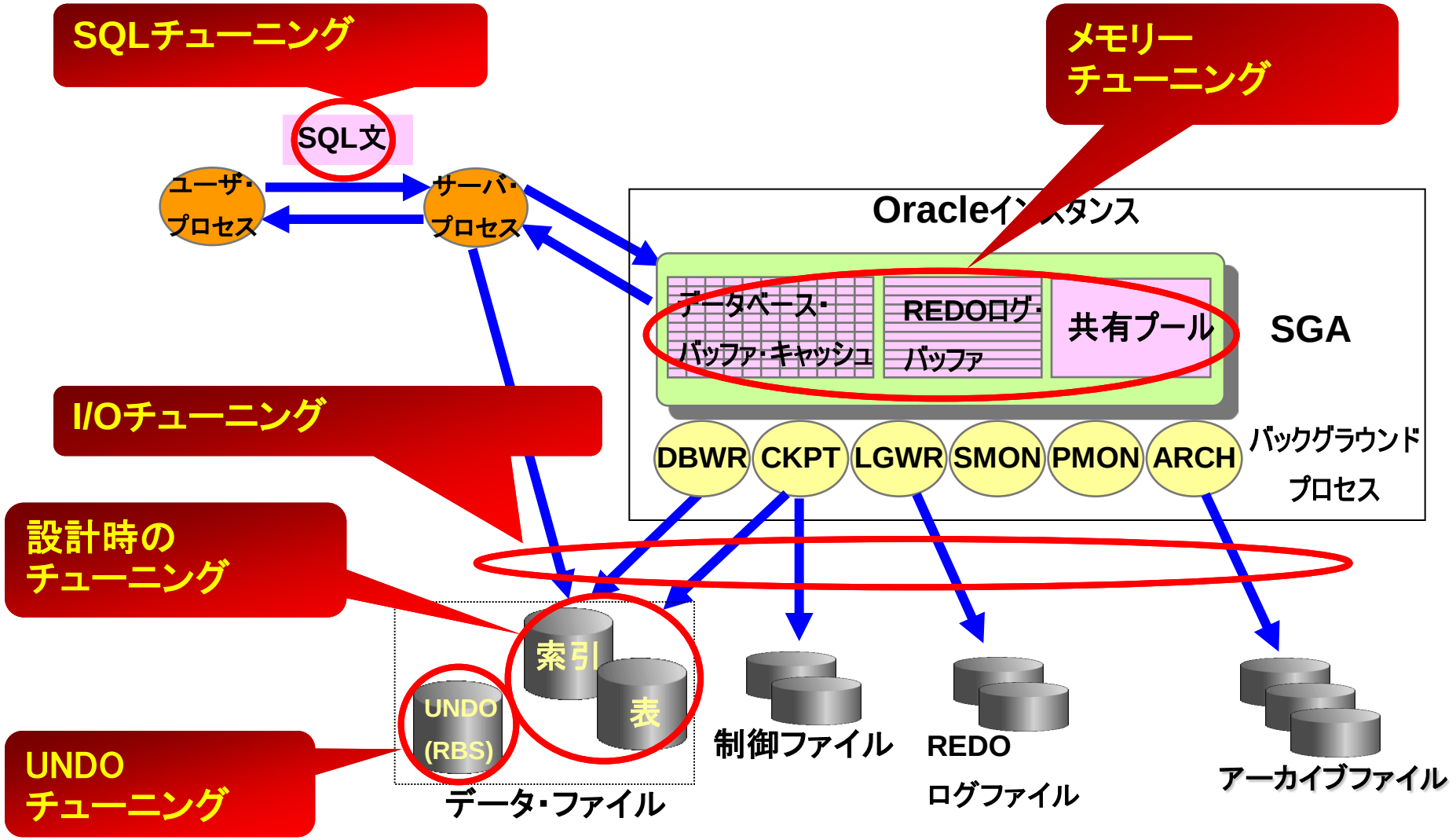
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

ボトルネックとなる可能性のある箇所



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

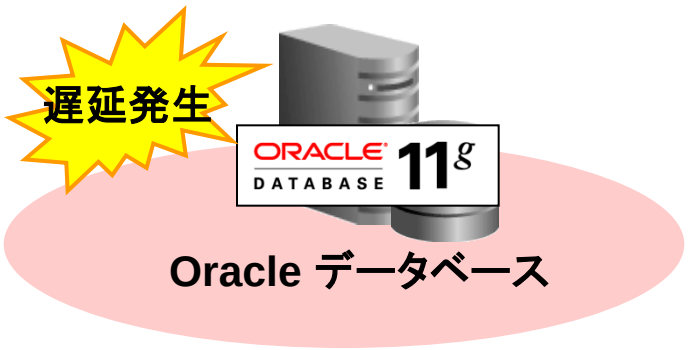
ボトルネックに対するチューニング



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

性能監視の考え方

アシスト



個々のボトルネックを監視するより、
遅延で生じる現象を捉えて監視した方が
管理が楽、かつ、監視漏れを防止できます

遅延を引き起こす要因

Oracle 以外のボトルネック
- CPU ネック
- I/O ネック
- ネットワーク待ち

Oracle 内部のボトルネック
- ロック競合
- ブロック競合
- 処理量増加に伴う遅延
- キャッシュ・フュージョンの遅延

Oracle の遅延を引き起こす

遅延により生じる現象

レスポンス時間の増加
リクエストのレスポンス時間が増える

待機時間の増加
待機イベントの待ち時間が増える

アクティブ・セッション数の増加
DB 側で SQL 実行中のシャドウが積み上がる

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

解決への手順

アシスト

パフォーマンス問題を解決するための、
「分析」と「対処」を実施

1. 性能問題となっている箇所を特定
2. チューニングを行う

今回の範囲

Oracle Database（DB）の
性能問題ではないかを確認

DB側の性能問題
ではない場合

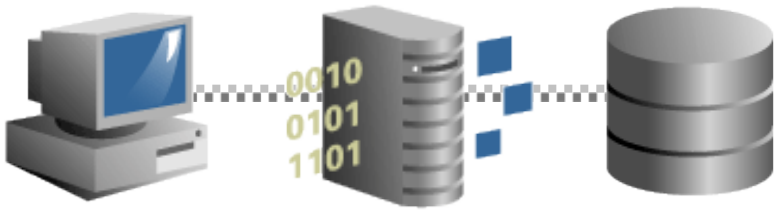
DB側の性能問題
であった場合

Application（AP）か
ネットワークの性能問題かを確認

DBのチューニングを行う

ネットワーク環境を改善

APのチューニングを行う



ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

DBの性能問題の確認

アシスト

DB内部の統計情報を取得し、性能問題となっている箇所がないかを確認する

- DB内部の統計情報とは
  - パフォーマンス統計情報（ex. リソース使用状況や待機の発生情報）
  - 各SQL実行時の詳細情報

＜方法＞

- Statspack（8i~）
- AWRレポート（10g~）
  ※Enterprise EditionでDiagnostic Packオプションが別途必要 **EE** **+** **Diag Pack**
- SQLトレース（主にテスト環境で使用）

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

確認したい情報と診断ツール

アシスト

確認したい情報	診断ツール
ある期間のパフォーマンス統計情報 （メモリー・ヒット率、待機イベント情報、etc.）	STATSPACK Oracle Enterprise Manager
SQL文の実行計画	EXPLAIN PLAN SQL*Plus AUTOTRACE SQLトレース Oracle Enterprise Manager
SQL文のパフォーマンス統計情報	SQL*Plus AUTOTRACE SQLトレース STATSPACK Oracle Enterprise Manager

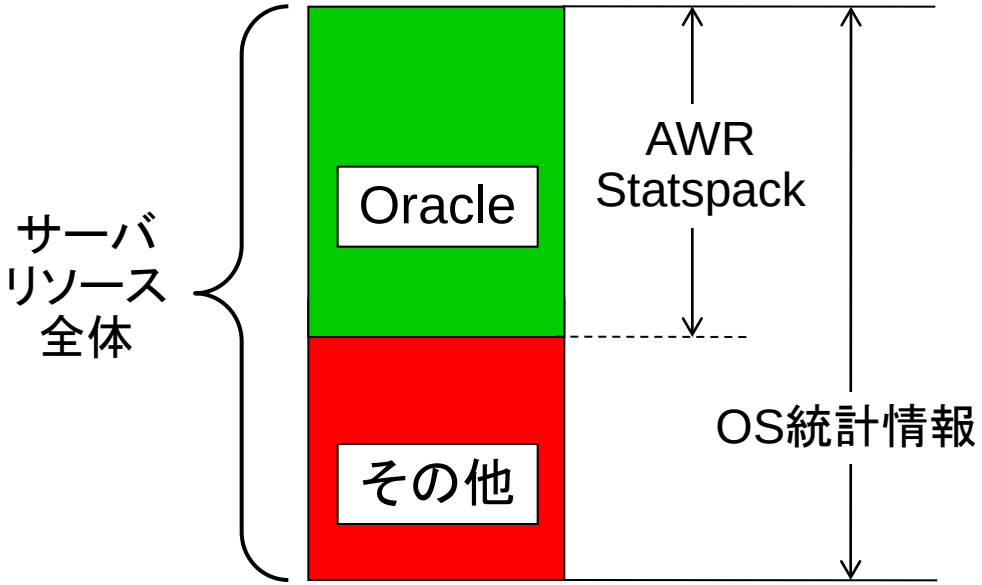
ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Copyright© 2010, Oracle. All rights reserved.

<ご参考>その他取得しておくとよい情報

OS統計情報
- ーCPU統計情報
- ーディスク統計情報
- ーメモリー統計情報
- ーネットワーク統計情報
- ー稼動プロセス統計情報



OS側の統計情報は、Oracle以外の部分を含むサーバ全体の
リソース情報を持つため、原因がDB側にある可能性が高くても取得する

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

分析方法の理解：Statspackとは

アシスト

- DB内部の統計情報を使って分析を行う前に、統計情報を取得する
  各方法の概要を理解する

＜方法＞

- Statspack（8i~）
- AWRレポート（10g~）
  ※Enterprise EditionでDiagnostic Packオプションが別途必要
- SQLトレース（主にテスト環境で使用）

EE ＋ Diag Pack

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Statspack(Statistics Package)とは

アシスト

- Statspack とは
  ーOracle 8i から実装されているオラクル標準の性能分析ツール
- Statspack を利用すると
  ーある期間でのOracle の稼動状況を確認することができる

- 基本的なStatspack利用の流れ
  1. インストール後、定期的に統計情報を取得し蓄積しておく
  2. 問題が発生した場合、蓄積した統計情報からレポートを生成する
  3. 現時点のレポートを（必要なら過去時点のレポートも同時に）比較し分析を行う

Statspack の情報を確認し、DBが行う処理の中で
どこか性能低下の要素となっている箇所はないかを確認する

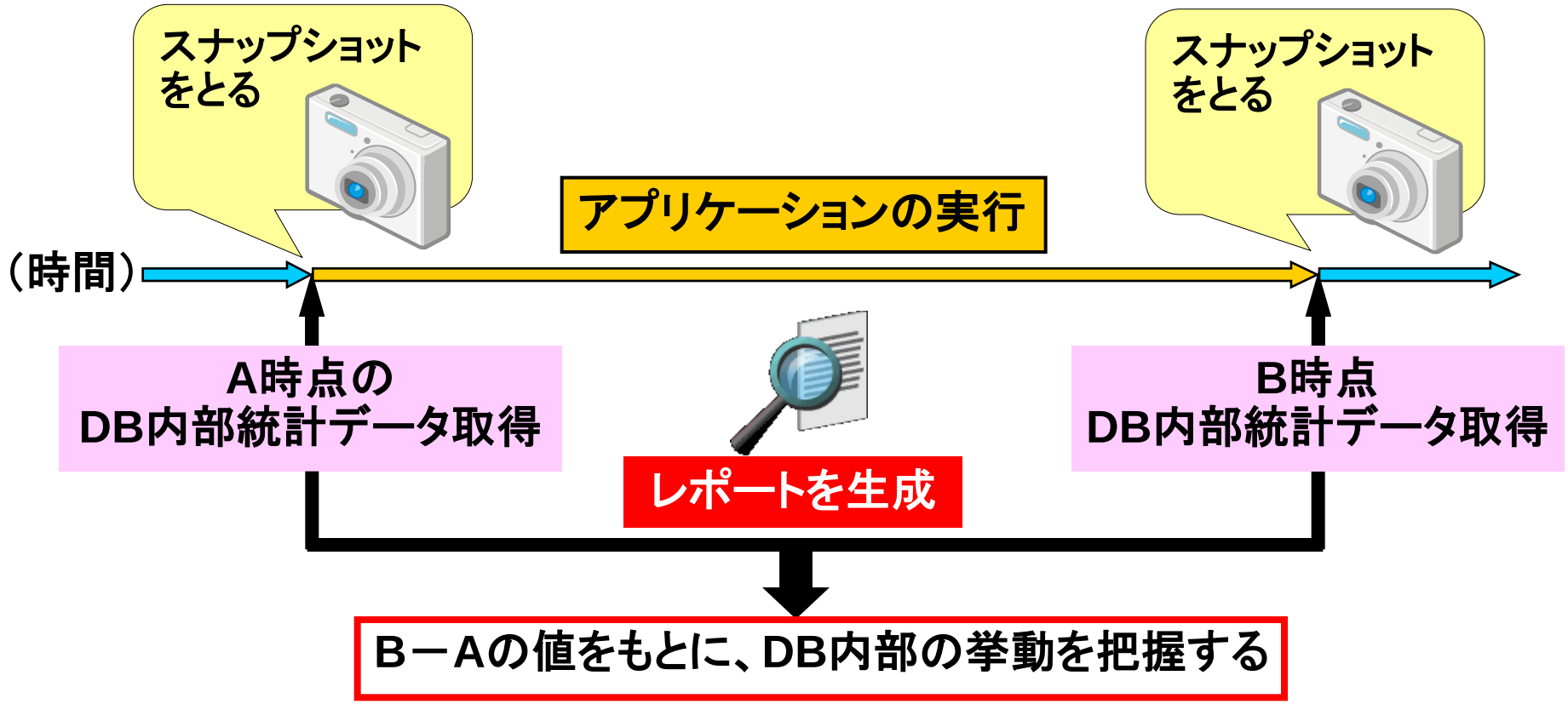
ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Statspackの仕組み

アシスト

- Statspackは、2つの異なる時点での情報をそれぞれ
  スナップショットとして記録しておき、差分からレポートを出力する



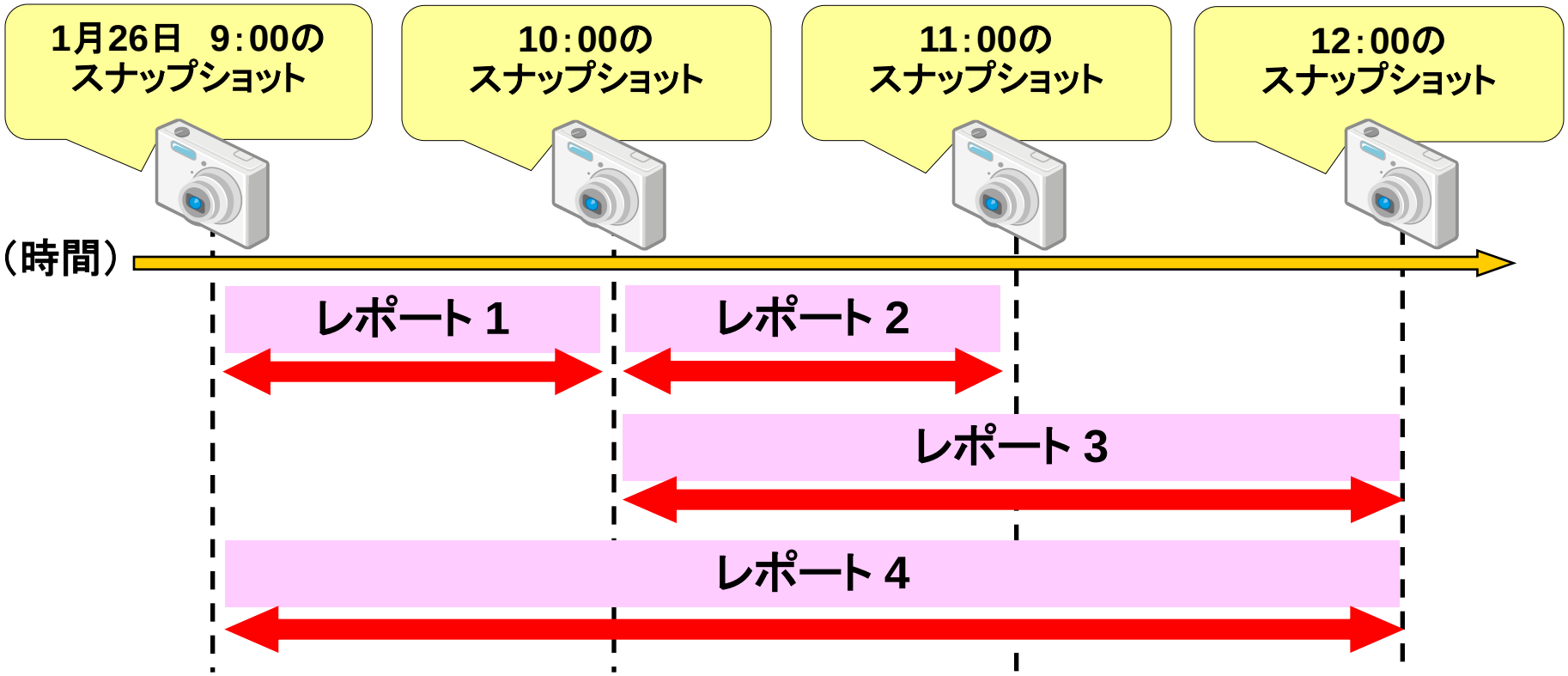
ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

柔軟なレポート出力

アシスト

- レポート出力時にどのスナップショットを使用するか指定するだけで、出力範囲を柔軟に変更することが可能



ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

<ご参考>Statspackで取れる情報(9iR2,10g,11g)

アシスト

レベル(取得する情報の詳細さ)設定参考基準

通常稼動時はLevel 5(デフォルト)、「最近ちょっと遅いなぁ」と感じ始めたらLevel 6
または Level 7、Level 10はOracle サポート等から指示された時のみ(高負荷)

スナップショットレベル	収集データ					
	基本統計情報	アドバイス情報	SQL統計情報	SQL詳細情報	セグメント情報	ラッチ詳細情報
Level 0	○	○				
Level 5	○	○	○			
Level 6	○	○	○	○		
Level 7	○	○	○	○	○	
Level 10	○	○	○	○	○	○

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Statspack 実際のレポート出力例

STATSPACK report for

Database	DB Id	Instance	Inst Num	Startup Time	Release	RAC
	1196126054	orcl1	1	17-10月-08 11:40	11.1.0.7.0	YE

Host	Name	Platform	CPUs	Cores	Sock
	jpintl005.jp.ora	Linux IA (32-bit)			

Snapshot	Snap Id	Snap Time	Se
Begin Snap:	36	21-10月-08 19:46:15	
End Snap:	37	21-10月-08 19:47:51	
Elapsed:		1.60 (mins) Av Act Sess:	

.
.
.

SQL ordered by Elapsed time for DB: ORCL Instance: orcl1 Snaps: 36 -37

-> Total DB Time (s): 42

-> Captured SQL accounts for 195.5% of Total DB Time

-> SQL reported below exceeded 1.0% of Total DB Time

Elapsed Time (s)	Executions	Elap per Exec (s)	%Total	CPU Time (s)	Physical Reads	Old Hash Value
40.69	1	40.69	95.9	40.67	0	3417014992

Module: SQL*Plus

```
declare   v_num number; begin   for i in 1..10000 loop     selec
t c1 into v_num from t1 where c1 = 100;   end loop; end;
```

40.48	10,000	0.00	95.4	40.46	0	2968801179

Module: SQL*Plus

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

分析方法の理解：AWRレポートとは

アシスト

- DB内部の統計情報を使って分析を行う前に、統計情報を取得する
  各方法の概要を理解する

＜方法＞

- Statspack（8i~）
- AWRレポート（10g~）
  ※Enterprise EditionでDiagnostic Packオプションが別途必要
- SQLトレース（主にテスト環境で使用）

EE ＋ Diag Pack

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

AWR（自動ワークロードリポジトリ）

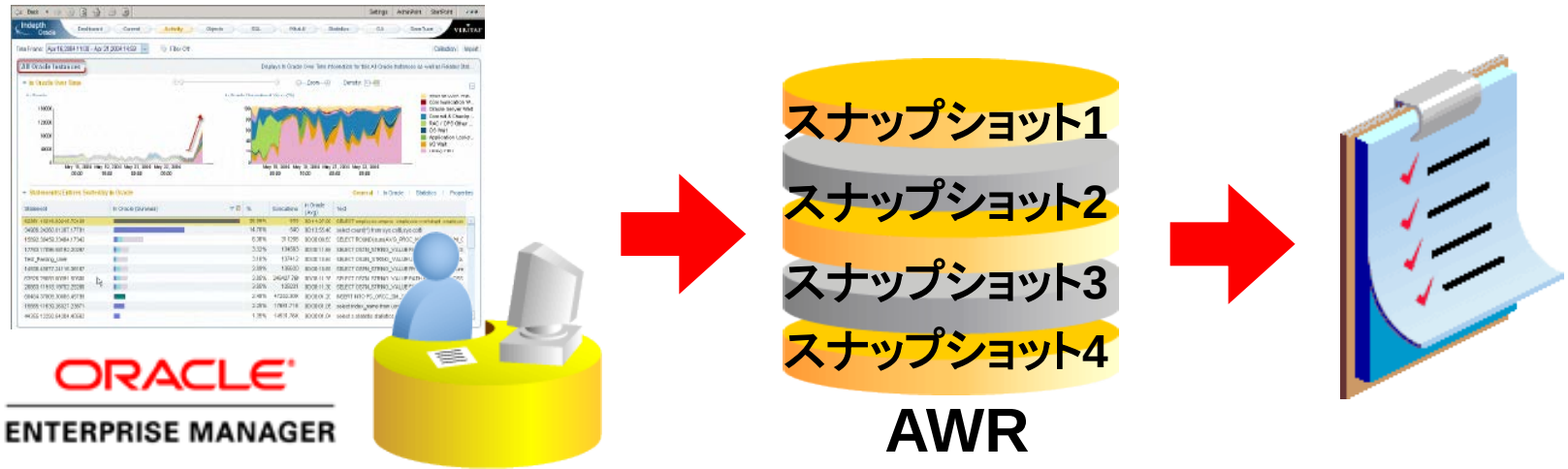
アシスト

EE ＋ Diag Pack

Oracle 10gからDB内部の統計情報（スナップショット）を自動取得

AWRはStatspackを進化させた機能

- Statspackと同様に各種レポートを作成するが低負荷で種類も多く見やすい
- Enterprise Manager(EM)を使って設定内容の変更や確認が操作できる
- ADDMによる自動パフォーマンス監視 / 診断が可能



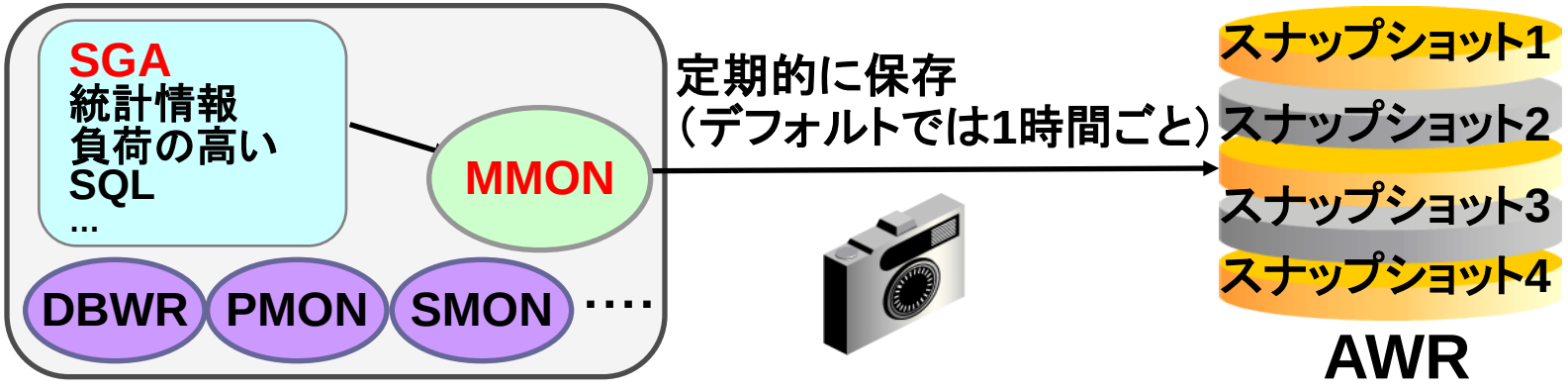
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

AWR スナップショットの自動取得

アシスト

EE ＋ Diag Pack

- AWRはDB作成と同時にSYSAUX表領域に作成されている
  ーインストール作業が不要
- MMON（Memory Monitor）が定期的にSGAの情報を取得する
  ーデフォルトで1時間に1回のスナップショットを一定期間保存（11gでは8日間）
  ー設定内容の確認・変更にはOracle EMを使用可能
- 古いデータの削除も自動で実行される



ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

AWR 設定画面

アシスト

EE + Diag Pack

- 「サーバー」タブの統計管理にある自動ワークロード・レポジトリから設定画面へ

編集
設定の変更

AWRレポート
の実行

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

レポートの作成　開始スナップショットの選択

アシスト

EE ＋ Diag Pack

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

レポートの作成　終了スナップショットの選択

アシスト

EE ＋ Diag Pack

終了スナップショットを
選択してOKをクリック

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

レポートの出力

アシスト

EE + Diag Pack

**Statspack と比較して
より見やすく詳細な
レポートを表示する**

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

StatspackとAWRのレポート比較

アシスト

EE ＋ Diag Pack

SQL ordered by Gets DB/Inst: ORCL/orcl1 Snaps: 38-39

Buffer Gets Executions Gets per Exec %Total Time (s) Time (s) Hash Value

20,071 1 20,071.0 87.7 0.22 0.23 3417014992

Module: SQL*Plus

declare v_num number; begin for i in 1..10000 loop selec

t c1 into v_num from t1 where c1 = 100; end loop; end;

429 1 429.0 1.9 0.01 0.00 688622149

Module: SQL*Plus

INSERT INTO STATS$EVENT_HISTOGRAM (SNAP_ID , DBID , INSTANCE_NUMBER , EVENT_ID , WAIT_TIME_MILLI , WAIT_COUNT) SELECT :B3 , :B2 , :B1 , EN.EVENT_ID , WAIT_TIME_MILLI , WAIT_COUNT FROM V$EVENT_HISTOGRAM EH , V$EVENT_NAME EN WHERE EH.EVENT = EN.NAME AND EH

AWRのメリット

- STATSPACKより見やすい
- STATSPACKより低負荷
- STATSPACKより多くの情報を収集
- ADDMが使用可能

SQL ordered by Gets

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- Total Buffer Gets: 11,852,217
- Captured SQL account for 99.0% of Total

Buffer Gets	Executions	Gets per Exec	%Total	CPU Time (s)	Elapsed Time (s)	SQL Id	SQL Module	SQL Text
11,393,880	20	569,694.00	96.13	24.78	24.80	44nz3b1nk3sxh	OEM.CacheModeWaitPool	SELECT x.STATE_GUID, x...
297,695	20	14,884.75	2.51	8.16	8.71	bjsc9c4gg59jj	OEM.CacheModeWaitPool	BEGIN EMDW_LOG.set_context(MGM...
176,400	40	4,410.00	1.49	4.30	4.35	fyk8b9986ntk7	OEM.CacheModeWaitPool	SELECT EXECUTION_ID, STATUS, ...
86,180	20	4,309.00	0.73	2.86	2.87	7q77fj5a9p1jg	OEM.CacheModeWaitPool	SELECT COUNT(1) FROM MGMT_JOB_...
32,791	59	555.78	0.28	0.98	2.45	6gvch1xu9ca3g		DECLARE job BINARY_INTEGER := ...
28,825	0		0.24	2.13	31.17	b6usrg82hwsa3	DBMS_SCHEDULER	call dbms_stats.gather_databas...
26,626	0		0.22	0.94	30.12	6mcpb06rctk0x	DBMS_SCHEDULER	call dbms_space.auto_space_adv...
21,806	0		0.18	2.05	25.13	b5vd6xtzncv5v	DBMS_SCHEDULER	MERGE /*+ dynamic_sampling(ST ...
18,420	20	921.00	0.16	0.58	0.52	8nn5ta3ydym64	OEM.CacheModeWaitPool	SELECT COUNT(*) FROM MGMT_JOB_...
17,052	1,684	10.13	0.14	1.59	1.96	6v7n0y2bq89n8	OEM.SystemPool	BEGIN EMDW_LOG.set_context(MGM...

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Statspack・AWRレポートを使った分析

アシスト

- Statspack と AWR レポートの概要を理解したら、取得した情報を基に分析を行う

＜方法＞

- Statspack（8i~）
- AWRレポート（10g~）
  ※Enterprise EditionでDiagnostic Packオプションが別途必要
- SQL トレース（主にテスト環境で使用）

EE ＋ Diag Pack

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Statspack レポートの分析

アシスト

- Statspackには多くの情報が含まれているが、パフォーマンスの観点から特に次の箇所に注目して性能問題がないかを確認する

分析ポイント

- スループットと負荷（Load Profile）
- インスタンス効率（Instance Efficiency）
- 待機イベント（Top-5 Timed Events）
- SQLの詳細情報（SQL ordered by ~）

原因箇所を特定するのに役立つ重要な情報

特に本番環境で使用する場合は、業務内容や時間帯によっても統計情報の取得結果が異なってくるので複数のレポートを比較することを推奨します

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Load Profile 負荷状況の読み取り

アシスト

- ディスクアクセスを伴うデータの読み込みや書き込みが行われていないかという点に着目し、DBの処理でボトルネックとなりやすい物理I/Oを確認

Load Profile	Per Second	Per Transaction	Per Exec	Per Call
Redo size:	481.4	4,405.4		
Logical reads:	28.1	257.3		
Block changes:	7.4	67.4		
Physical reads:	0.0	0.3		
Physical writes:	0.2	1.6		
User calls:	1.5	13.3		
Parses:	0.5	4.9		
Executes:	105.6	921.6		

SQLの処理量の確認
数値が高いと非効率なSQLが実行されている可能性がありSQLチューニングの対象となる

スループットの目安

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Instance Efficiency インスタンス効率

アシスト

- DBバッファキャッシュヒット率を確認し、インスタンスが効率よく稼動しているかを確認

参考値に基づいて%を確認
場合によってはOracleの初期化パラメータ、もしくはSQLをチューニング

Instance Efficiency Indicators

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Buffer Nowait %:	100.00	Redo NoWait %:	100.00
Buffer Hit %:	99.88	Optimal W/A Exec %:	100.00
Library Hit %:	97.86	Soft Parse %:	97.38
Execute to Parse %:	58.04	Latch Hit %:	100.00
Parse CPU to Parse Elapsd %:	85.55	% Non-Parse CPU:	94.88

- 全ての値を100%に近づけることが目標です
  - 80%以上・・・%Non-Parse CPU
  - 90%以上・・・Buffer Hit%, In-memory Sort%, Soft Parse%
  - 95%以上・・・Library Hit%, Redo Nowait%, Buffer Nowait%
  - 98%以上・・・Latch Hit%

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

<ご参考> 各パラメータの説明

アシスト

パラメータ名	説明
Buffer Hit%	必要なデータがバッファ上にあった割合
Library Hit %	必要なSQL、PL/SQLがライブラリ・キャッシュにあった割合
Soft Parse %	全ての解析のうち再利用可能なものの割合
In-Memory Sort%	ソートがメモリ内で行われた割合
Latch Hit%	全てのラッチのヒット率
Parse CPU to Parse Elapsed %	解析CPU時間/ 解析の合計時間
Execute to Parse%	SQL実行に対し解析が行われなかった割合
%non-parse CPU	解析以外で使用されたCPU時間の割合
Buffer Nowait%	バッファに要求を出したときに、即座に使用可能だった割合
Redo Nowait%	redo logに要求を出したときに、即座に使用可能だった割合

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

インスタンス効率　項目別チェックポイント

アシスト

- Buffer Nowait%　→BufferWaitセクション調査
- Buffer Hit%　→ db_cache_size
- Redo Nowait%　→ log_buffer
- Latch Hit%　→ラッチセクション調査
- In-Memory Sort%　→ sort_area_size, pga_aggregate_target
- Soft Parse%　→共有プールの調整
- Execute to Parse%　→共有プール調整、SQL共有を考慮しているか
- Parse CPU to Parse Elasped%→共有プール調整
- %Non-Parse CPU　→共有プール調整

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Top-5 Timed Events 待機イベント

アシスト

- インスタンスレベルで待機時間がもっとも長いイベント上位5つをチェックし、監視対象のインスタンスの性能を低下させていないかを確認

Top 5 Timed Events ~~~~~~~~~~~~~~~~~~ Event	Waits	Time (s)	Avg wait (ms)	%Total Call Time
CPU time		41		96.8
control file sequential read	4,141	1	0	2.8
control file parallel write	36	0	1	.1
ASM file metadata operation	19	0	1	.0
PX Deq: Join ACK	30	0	0	.0

CPU time
他の待機イベントが待機回数や平均待機時間が少ないのでリソースが有効活用されていると判断できる

待機上位イベント
性能低下につながっていそうなイベントがあればチューニング

Waits :イベントのために待機した合計回数
Time(s) :イベントの合計待機時間および合計CPU時間(秒)
Avg wait(ms) :イベントの平均待機時間
% Total Call Time: time for each timed event / total call time
Total call time は、total CPU time + total wait time for non-idle events

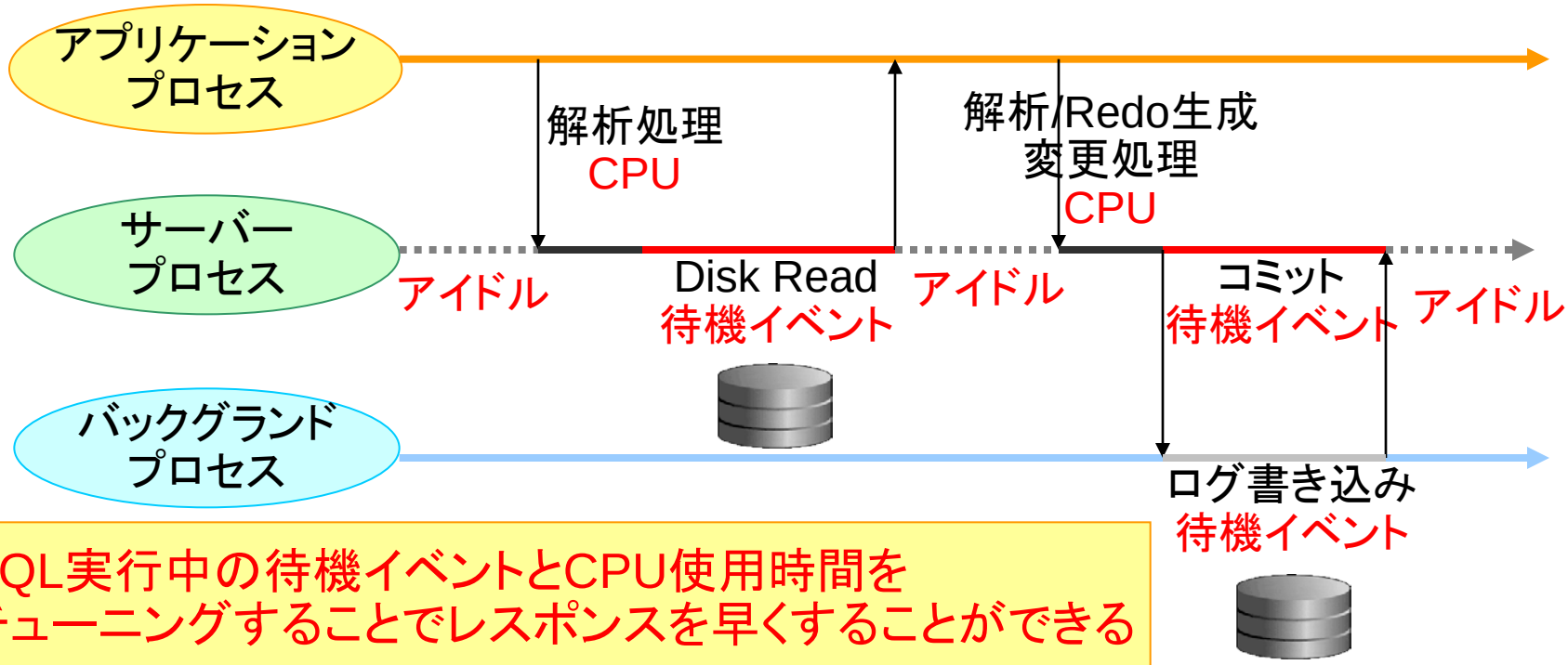
ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

<ご参考> 待機イベントとは

アシスト

- プロセスがCPUを使用していない時間
  - アイドル待機イベント（SQLのリクエスト待ち）
    - ボトルネックが存在する場合に、原因がDBリソースではないことを意味します
  - その他の待機イベント（SQL実行中） **<- DBのチューニングではこちらに注目**
    - DBリソース（バッファ競合、I/O競合、ラッチ競合など）に関連する待機時間



SQL実行中の待機イベントとCPU使用時間を
チューニングすることでレスポンスを早くすることができる

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

SQL ordered by ~ SQLの処理情報

アシスト

- 各SQLが読み込んだバッファ数やディスクへのアクセス回数を確認し、リソース使用率の高いSQLがないかを確認

SQL ordered by CPU DB/Inst: ORCL/orcl1 Snaps: 36-37
-> Total DB CPU (s): 41

CPU Time (s)	Executions	CPU per Exec (s)	%Total	Elapsd Time (s)	Buffer Gets	Old Hash Value
40.46	10,000	0.00	97.9	40.48	33,590,000	2968801179

Module: SQL*Plus
SELECT C1 FROM T1 WHERE C1 = 100

0.57	1	0.57	1.4	1.57	

Module: SQL*Plus
BEGIN statspack.snap; END;

全体の処理時間がかかっているSQLを見つける
「読み込みバッファ数が多い」
「CPUの処理時間が長い」
などに該当するSQLをチューニング対象とする

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

分析方法の理解：SQLトレースとは

アシスト

- DB内部の統計情報を使って分析を行う前に、統計情報を取得する
  各方法の概要を理解する

＜方法＞

- Statspack（8i~）
- AWRレポート（10g~）
  ※Enterprise EditionでDiagnostic Packオプションが別途必要
- SQLトレース（主にテスト環境で使用）

EE ＋ Diag Pack

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

SQLトレースとは

アシスト

SQLトレースを使用すると、SQL実行時のより詳細な情報が取得できる
取得した情報を分析することによって問題のあるSQLの特定を行える

- SQLトレースの取得方法には2通りある
  1. すべてのセッションの情報を取得する方法
  2. 特定のセッションのみの情報を取得する方法
- 取得した情報には、各SQLについて次のような情報が含まれる
  - SQL文の解析、実行、フェッチの実行回数
  - CPU時間、経過時間
  - 物理読み込み（Physical read）、論理読み込み（Logical read）
  - 処理された行数

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

SQLトレース使用手順

アシスト

- 全てのセッション情報を取得する場合

1. 初期化パラメータ SQL_TRACE をTRUEに設定
2. Oracleを再起動する

- 再起動後から設定をFALSEに変更して再起動するまでの間、セッションが開始される度にトレースファイルが作成され、各セッションの全ての情報が出力される

- 特定のアプリケーションについてのみトレースを取得する方法

1. アプリケーション内でSQLトレースを開始したいポイントに次のSQL文を追加

```
ALTER   SESSION SET SQL_TRACE=TRUE;
```

- この場合のトレース取得期間は、アプリケーションを終了するか、またはSQL_TRACE を FALSE に設定した時点までとなる

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

TKPROFによるフォーマット

アシスト

- SQLトレースの出力結果は、TKPROFを使用することによって見やすい形にフォーマットすることが出来る

＜UNIX/Linux環境でTKPROFを実行する場合の例＞

- TKPROFコマンドでorcl1_ora_12059.trc というトレースファイルをtkprof_1.prfへフォーマットする

```
$ tkprof orcl1_ora_12059.trc tkprof_1.prf
```

※ TKPROFの実行モジュール名は環境によって異なるので詳しくはマニュアルを参照ください

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

TKPROFの出力例

SQL ID: ck7jaztn6503v
Plan Hash: 3617692013
select * from t1 where c1 = 100

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	1	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	2	0.01	0.00	0	3360	0	1
total	4	0.02	0.01	0	3361	0	1

Misses in library cache during parse: 1
Optimizer mode: ALL_ROWS
Parsing user id: 32

Rows Row Source Operation
---
1 TABLE ACCESS FULL T1 (cr=3360 pr=0 pw=0 time=0 us cost=922 size=98649 card=97)

チューニング対象の特定
「処理時間が長い」
「ディスクI/Oが多発」
「読み取りブロック数が多い」
「極端に実行回数が多い」
などに該当するSQLを
チューニング対象とする

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

- 直積結合の回避
- SQL文の共有
- 列名の明示指定
- 結合は原則6表まで
- ソート処理の削減
- 不要なアクセスの削減

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

直積結合の回避

アシスト

- 直積結合とは、結合対象の表の全ての行の組み合わせを戻す処理
  - 結合条件に抜けがあるか、SQL文の書き方に無理がある場合がほとんどであるため、SQL文を見直し、書き直す

tableA = 1000件、tableB = 2万件 の直積結合

1,000 * 20,000 = 20,000,000

2000万回処理されてしまうが、結果が正しいと気付かない事も。

- 回避確認方法:
  - SQLトレースに「**CARTESIAN**」がない事を確認する

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

SQL文の共有

アシスト

- Oracleは、空白、大文字小文字、も含め全て一字一句同じSQLを同一とみなし、SQL解析情報を共有します
- 非同一の場合、毎回解析が行われてしまったり、SQLをキャッシュするメモリが枯渇する事もあるため、バインド変数を使います

非同一 SELECT ename FROM emp WHERE empno = 135;
SELECT ename FROM emp WHERE empno = 137;

同一 SELECT ename FROM emp WHERE empno = 変数;（変数値A）
SELECT ename FROM emp WHERE empno = 変数;（変数値B）

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

SQL文の共有

アシスト

- JDBCの例
  - PreparedStatement の利用
  - LOOPする場合は2行目以降を繰り返します

```
PreparedStatement pst =
 conn.prepareStatement(
 "UPDATE tableA SET ID1 = ? WHERE code = ?");
pst.setString(1, name);
pst.setString(2, code);
int result = pst.executeUpdate();
```

- Pro*Cの例
  - ホスト変数の利用

```
EXEC SQL INSERT INTO emp(empno, ename, deptno)
VALUES(:emp_number, :emp_name, :dept_number);
```

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

列名の明示指定

アシスト

- 指定対象
  - SELECT文の明示列名指定
  - INSERT文の明示列名指定
  - ORDER BY の明示列名指定
- 不要な列をFETCHさせない（性能向上）
- 解析処理が削減され、性能が向上します
- 保守性の向上
  - 列の追加、列順番の変更に対応

× SELECT * FROM TABLEA;
○ SELECT A1.選択列1、A1.選択列2 FROM TABLEA A1;

× INSERT INTO TABLEA VALUES (‘1’, ‘AA’);
○ INSERT INTO TABLEA (列1, 列2) VALUES (‘1’, ‘AA’);

× SELECT A1.列1, A1.列2 FROM TABLEA A1 ORDER BY 2;
○ SELECT A1.列1, A1.列2 FROM TABLEA A1 ORDER BY A1.列2;

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

結合は原則6表まで

アシスト

- 結合は原則6表までを推奨
  - OracleはSQL実行時、考えうる表の結合順序を評価し最適な結合順序を選ぼうとします
  - 結合表数が7を超えると、解析処理時間が急激に増加します。
    - 表の結合組み合わせが6つの表の時は720通りに対して、表が7つの時は5,040通りと大きく増加するためです
    - 4! = 24, 5! = 120, 6! = 720, 7! = 5,040, 8! = 40,320

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

ソート処理の削減

アシスト

- 不要な ORDER BY、GROUP BY の削除します
- GROUP BY ＋ HAVING を用いる場合、必ず事前にWHERE句で件数を絞ります
- 可能な限り暗黙的にソートが行われる処理を避けます
  - DISTINCT
  - GROUP BY
  - ORDER BY
  - UNION
  - INTERSECT
  - MINUS

  ※ UNIONではなく UNION ALLを使うなど

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

不要なアクセスの削減

アシスト

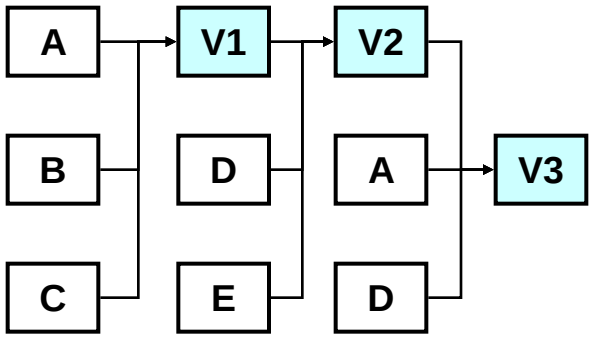
- LOOP中のSQLの削減（非常にトラブルが多い）
  - ロジック見直しは非常に高コスト！

    必ず事前チェックする
  - 外に出せるSQL文、prepare処理はLOOP外に出す
  - INSERT … SELECT 文の有効利用
- 複数のSQLを1つにまとめて、通信量を減らす
  - Merge 文
  - JDBC「バッチ更新」機能

- VIEWはネストさせない

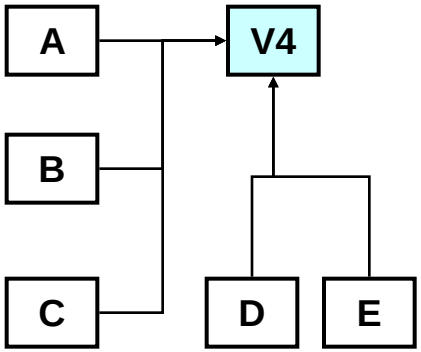
×

ビューをネストすると、非効率な実行計画になったり、同じテーブルに2回以上アクセスしてしまう可能性がある



○

V4を明示的に作る事で最適な実行計画になる



ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

統計情報を必ず取得

アシスト

- 開発・テスト時には統計情報を必ず取得します
  - 表・索引を追加したり、データを再投入した場合も再実行
  - 統計情報が不十分だと、最適な実行計画にならない可能性が高い
- C/O後は統計情報取得の運用を確定させます
  - 統計情報と実行計画を管理
    - 統計情報の取得タイミング、取得対象

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

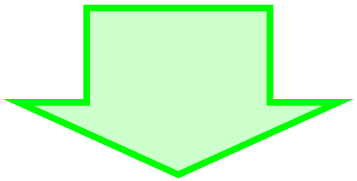
- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引の使用

- 索引の使用により、表データへの高速なアクセス・パスが提供されます
- 索引が存在しないか、存在するがOracleが使用できない場合、全表走査となります



索引が使われないと、表データサイズに比例して大幅に性能が劣化

基本的に選択行数が少ない方が索引走査が有利になる。

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引を利用するために

アシスト

- 索引列の値とNULLの比較やNOTを使わない
- 索引列には計算をさせない
- LIKE句を使った中間一致・後方一致検索
- 索引を作成したのに検索が遅い時

- 注意事項
  - 使わない不要な索引は作成しない
  - 索引の格納効率の悪化(断片化)には注意
  - ビットマップ索引は対象データの更新量に留意

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引列の値とNULLとの比較やNOTを使わない (B*Tree Index利用時)

アシスト

×

```
1. SELECT  ENAME FROM  EMP
      WHERE  COMM  IS  NULL  ;
2. SELECT  ENAME FROM  EMP
      WHERE  COMM  IS  NOT  NULL
```

×

```
1. SELECT  ENAME FROM  EMP
      WHERE  DEPTNO  ! =  30  ;
```

索引を使用しないで全表走査となる

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引列には計算をさせない

アシスト

×

```
1. SELECT  ENAME FROM  EMP
       WHERE  SAL *  1.1 > 950
2. SELECT ENAME  FROM  EMP
       WHERE  TO_CHAR (  HIREDATE ,'DDMMYY ') = '280595 '
 3. SELECT ENAME  FROM  EMP
       WHERE  ENAME = NVL(:ename_param,ENAME)
```

索引を使用しないで全表走査となる

○

```
1. SELECT  ENAME FROM  EMP
       WHERE  SAL > 950 / 1.1
2. SELECT ENAME  FROM  EMP
       WHERE  HIREDATE  =  TO_DATE( '280595 ' , 'DDMMYY ')
3. SELECT ENAME  FROM  EMP
       WHERE  ENAME like NVL(:ename_param,'%')
```

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

LIKE句を使った中間一致・後方一致検索

アシスト

索引は先頭キャラクタから順にキーとしてツリーを
作成しているので前方一致は索引が使われるが、
中間一致・後方一致では索引が使われない

索引が使われる

○

```
<前方一致>
  SELECT empno , ename  FROM  emp  WHERE ename  LIKE 'MI%' ;
```

索引が使われない

×

```
<中間一致>
  SELECT empno , ename  FROM  emp  WHERE ename  LIKE '%MIT%' ;
<後方一致>
  SELECT empno , ename  FROM  emp  WHERE ename  LIKE '%NER' ;
```

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引を作成したのに検索が遅い時

アシスト

1. 索引を使用する事で速くなる処理か？
   - フルスキャンの方が有利な検索ではないか
2. SQL文が索引を利用できない文になっていないか？
   - likeの中間一致、後方一致をしていないか
   - 索引を利用できない書き方をしていないか
3. オプティマイザが索引利用を選択していないのではないか？
   - まず実行計画を確認する
   - 索引を利用していなかったら
     - オプティマイザモードは？
     - ANALYZEを実行したか？

     (USER_TABLES の LAST_ANALYZEカラムで確認)
     - ヒントを使ってみる
4. 索引のメンテナンスをしているか？
   - DROP INDEX→ CREATE INDEX
   - ALTER INDEX 索引名 REBUILD
   - ALTER INDEX 索引名 SHRINK SPACE

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引の監視(未使用索引の検出)

アシスト

V$OBJECT_USAGEビューに索引の使用状況に関する統計情報を格納できます。

- 索引の使用状況の監視を開始

```
ALTER INDEX 索引名 MONITORING USAGE
```

- 監視を終了

```
ALTER INDEX 索引名 NOMONITORING USAGE
```

- 索引の使用状況を検索

```
SELECT index_name,used
    FROM V$OBJECT_USAGE;
```

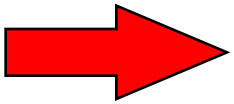
ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引の監視(索引構造の分析)

アシスト

```
ANALYZE INDEX 索引名 VALIDATE STRUCTURE
```



INDEX_STATS データディクショナリ
に分析結果が格納

主な項目	NAME	索引の名前
	PCT_USED	領域の使用比率(%)
	DEL_LF_ROWS	削除されたリーフの行
	HEIGHT	Bツリーの高さ

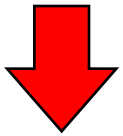
- **領域の使用比率(PCT_USED)を定期的に監視し、使用率が平均的な使用率より低下したら再構築を検討**

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引構造の再構築

索引の削除 (DROP INDEX)



索引の作成 (CREATE INDEX)

ALTER INDEX
索引名 REBUILD ONLINE

EE

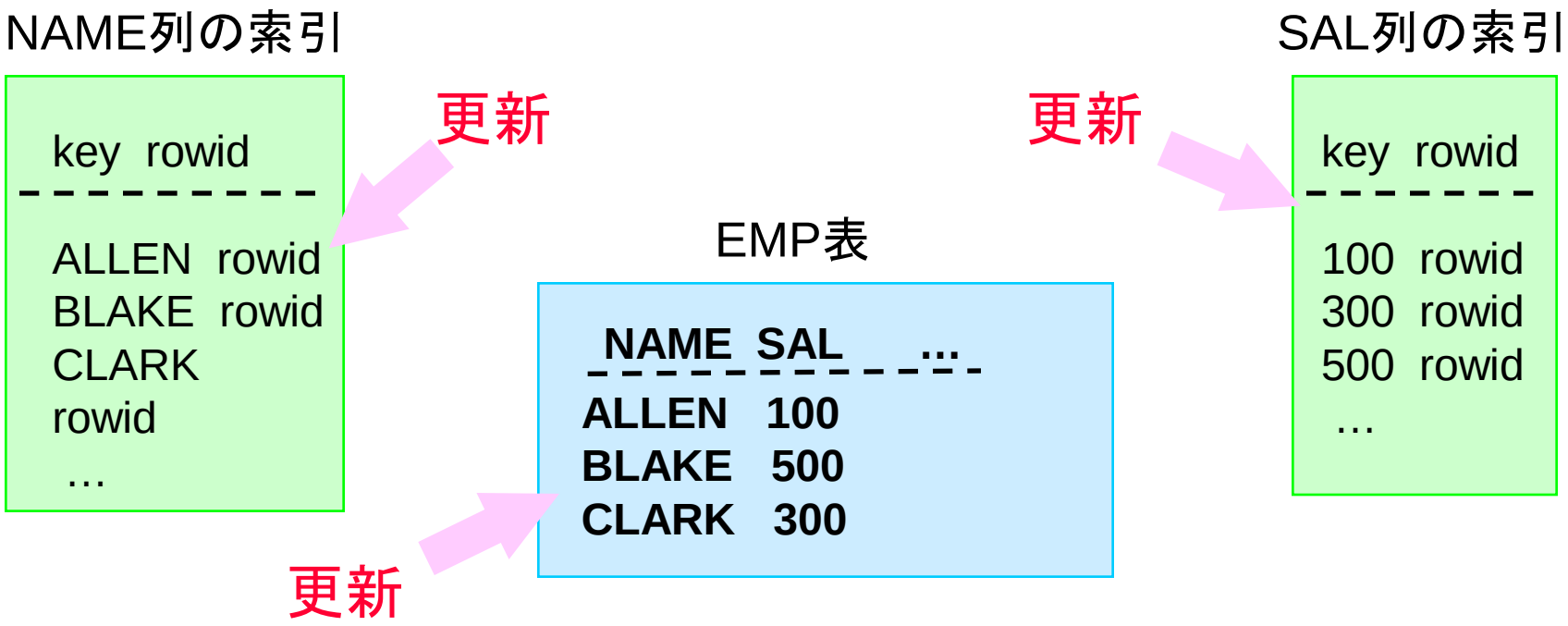
REBUILD中でも、索引にアクセス可能
新索引を作成してから旧索引を削除するので、
一時的に2倍の領域が必要
領域が足りない場合は、別の領域を指定可能

既にソートされているデータ
を利用するので高速

ex) ビットマップ索引のメンテナンス
夜間に大量のデータローディング
索引の削除 → データローディング → 索引の作成

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

索引を使う欠点



表データと索引データを更新しなければならない
→ 更新処理においては、負荷がかかる

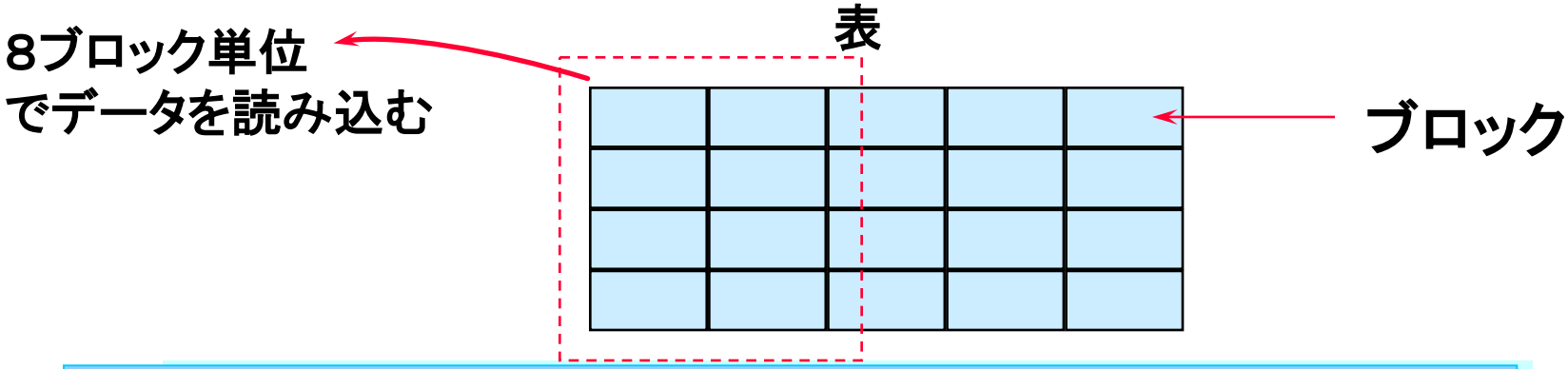
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

マルチブロックREAD

アシスト

1回のI/Oで複数のブロックを読み込む機能

→　DISKへのI/O回数を減らす事ができます



初期化パラメータ　db_file_multiblock_read_count = 8

現在のシステムは一般的に、DISK I/O ネックになる事が多いので効果大

全表走査　→　マルチブロックREAD
索引走査　→　シングルブロックREAD

選択率が高くなる程、
全表走査が有利になる

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

チューニング効果の確認

アシスト

SQL文が複雑になるにつれてチューニングの方法も何通りも考えられるので、目指す効果が表れるまでチューニングを繰り返すこともあるが、SQL*PlusのAUTOTRACE機能を使用すれば大まかな効果を即座に確かめることができる

- AUTOTRACEを使用すると以下項目を確認できる
  - SQL*Plusから実行したSQL文
  - 実行計画の結果
  - 統計情報
    - ー物理読み込み　　ー論理読み込み
    - ーREDOサイズ　　ーメモリソートサイズ
    - ー処理件数　　ーディスクソートサイズ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

AUTOTRACE機能の設定手順

アシスト

- AUTOTRACEの設定手順は以下の通り

Step1　PLUSTRACEロールを作成（初回のみ）

```
SQL> connect sqlplus / as sysdba
SQL> @$ORACLE_HOME/sqlplus/admin/plustrce.sql
```

Step2　作成したPLUSTRACEロールを使用ユーザに付与

```
SQL> grant plustrace to ＜ユーザ名＞
```

Step3　使用ユーザでAUTOTRACE機能が使用する作業用の表を作成

```
SQL> @$ORACLE_HOME/sqlplus/admin/utlxplan.sql
```

AUTOTRACE機能を使用させたいユーザごとに、Step2とStep3を繰り返す

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

AUTOTRACEの取得

アシスト

1. 使用ユーザでSET AUTOTRACEコマンドを実行

```
SQL> set autotrace on
```

2. SQLを実行する

```
SQL> select * from t1 where c1 = 100;
```

※AUTOTRACE機能を終了したい場合は

```
SQL> set autotrace off
```

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

AUTOTRACEの取得例

実行計画

```
| Id  | Operation                    | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |          |     1 |  2005 |     2   (0)| 00:00:01 |
|   1 | TABLE ACCESS BY INDEX ROWID| T1| 1 |2005 |     2   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN| C1_INDEX |    1  |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("C1"=100)
```

統計

```
----------------------------------
        338  recursive calls
          0  db block gets
         51  consistent gets
          9  physical reads
          0  redo size
       2558  bytes sent via SQL*Net to client
        420  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          6  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

統計の出力

論理読み込み：db block gets + consistent gets
物理読み込み：physical reads
REDOサイズ：redo size
メモリソート回数：sorts(memory)
ディスクソート回数：sorts(disk)
処理件数：rows processed

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

アシスト

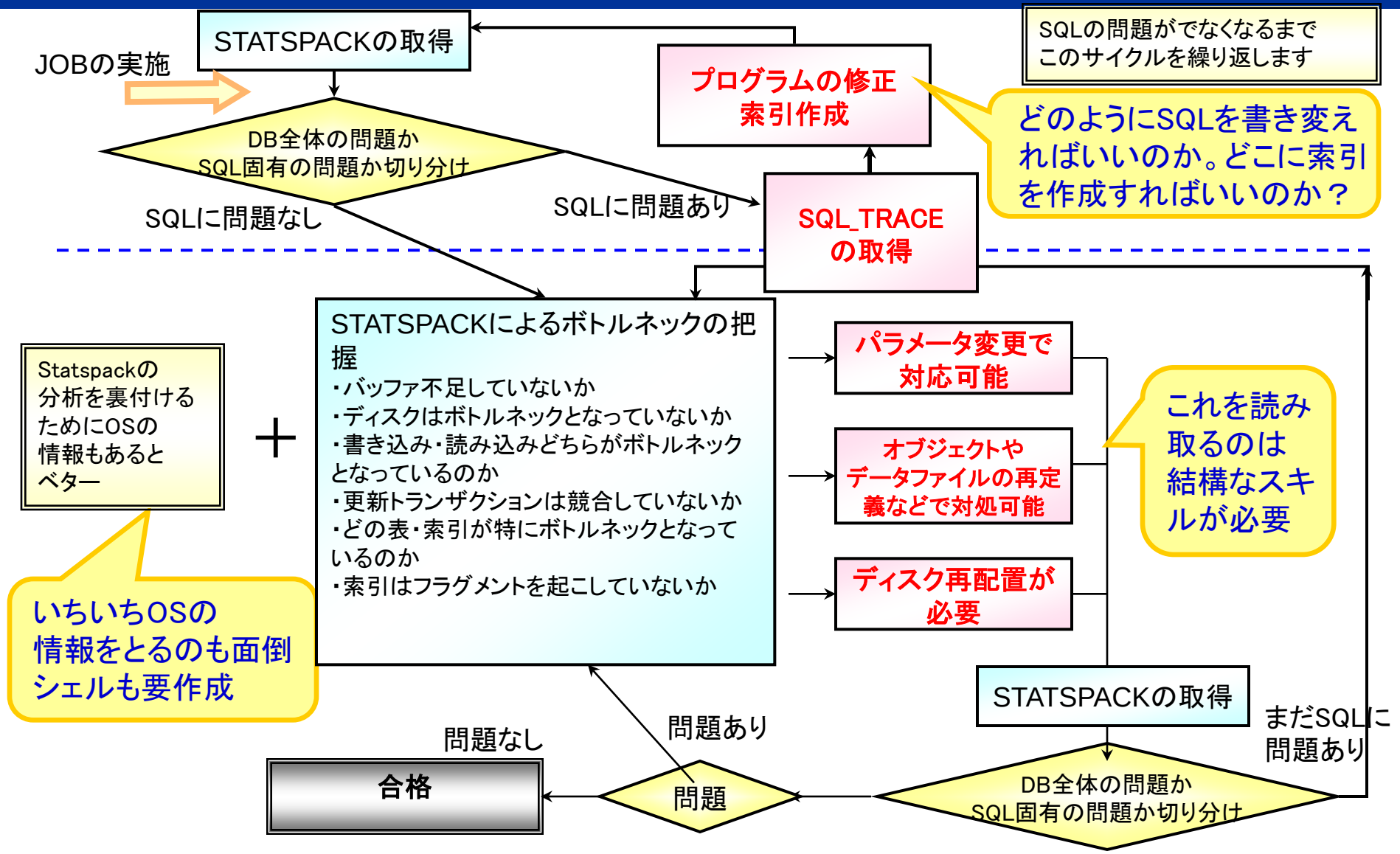
- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Statspackを利用したボトルネックの把握とチューニング例

アシスト



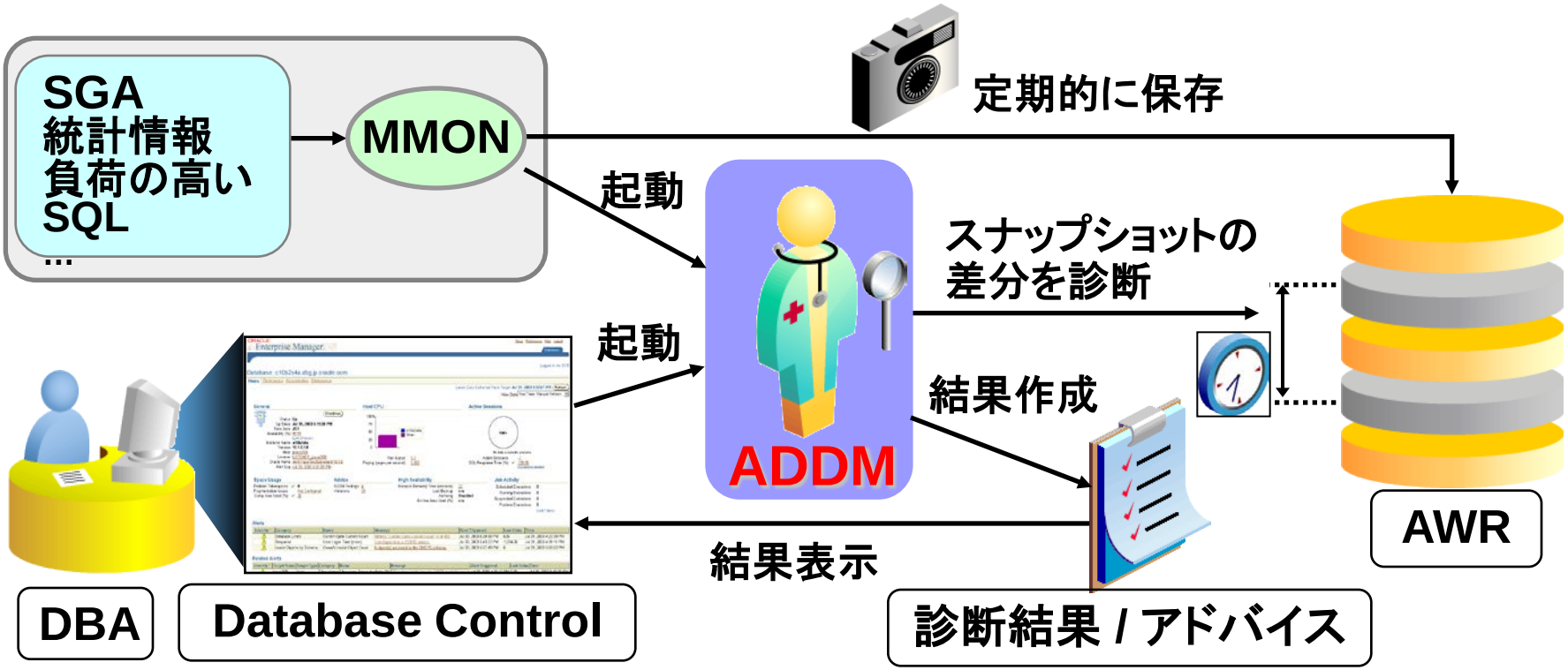
ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

ADDM 自動データベース診断モニター (Automatic Database Diagnostic Monitor)

EE + Diag Pack

- AWRに収集された統計情報をもとに、定期的なDBのパフォーマンス監視 / 診断をDB管理者（DBA）向けに行ってくれる機能



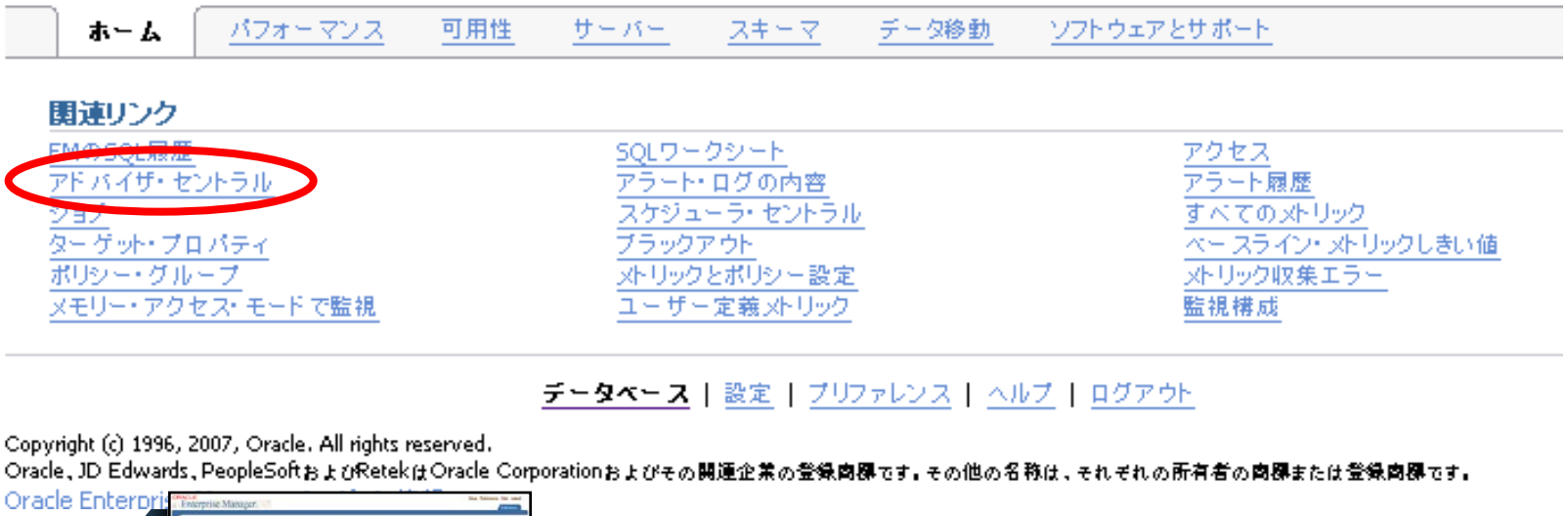
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

診断結果とアドバイスの表示方法

アシスト

EE + Diag Pack

- ADDMに関する情報は、Database Control の「ホーム」タブの「関連リンク」グループ内にある「アドバイザ・セントラル」リンクから検索
- 手動実行の場合は、「アドバイザ」グループの「ADDM」を実行する



ORACLE
ENTERPRISE MANAGER

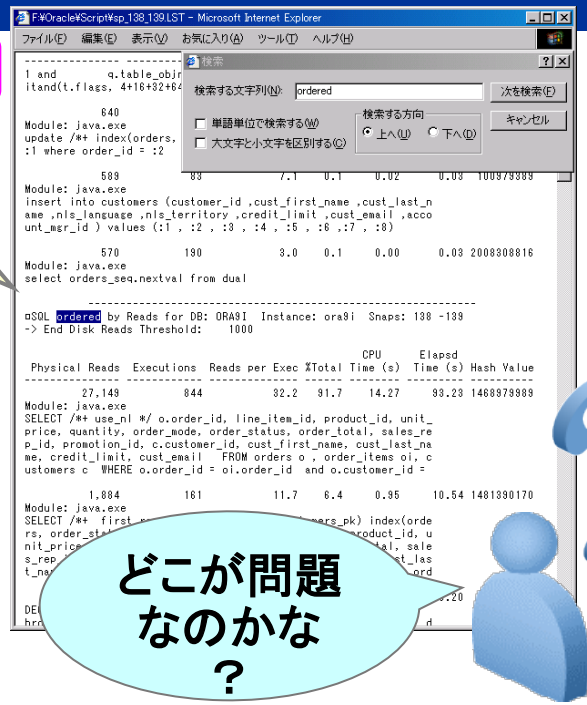
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

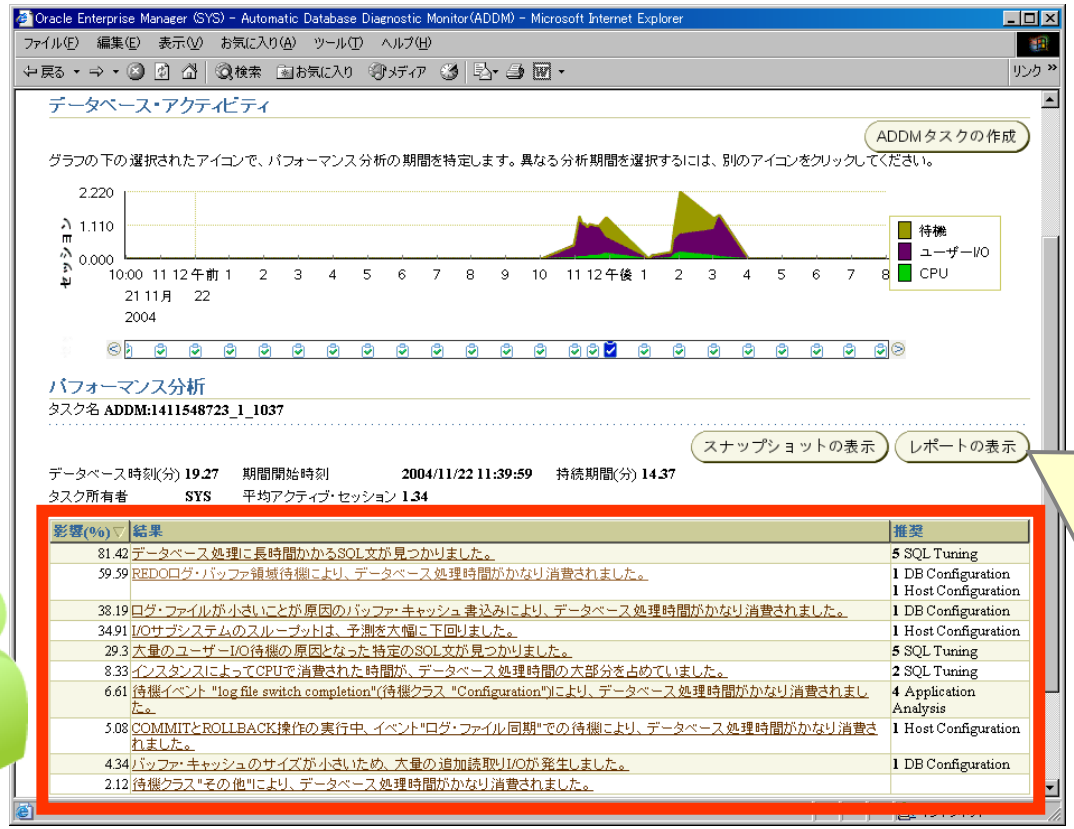
自動で診断レポートを作成

アシスト

EE ＋ Diag Pack

STATSPACKでは、管理者がレポートを解析し、DBAがチューニングをしていましたが…





ADDMでは自動で診断レポートを作成、パフォーマンスをはじめとした分析結果をブラウザ上でドリルダウン！

ORACLE

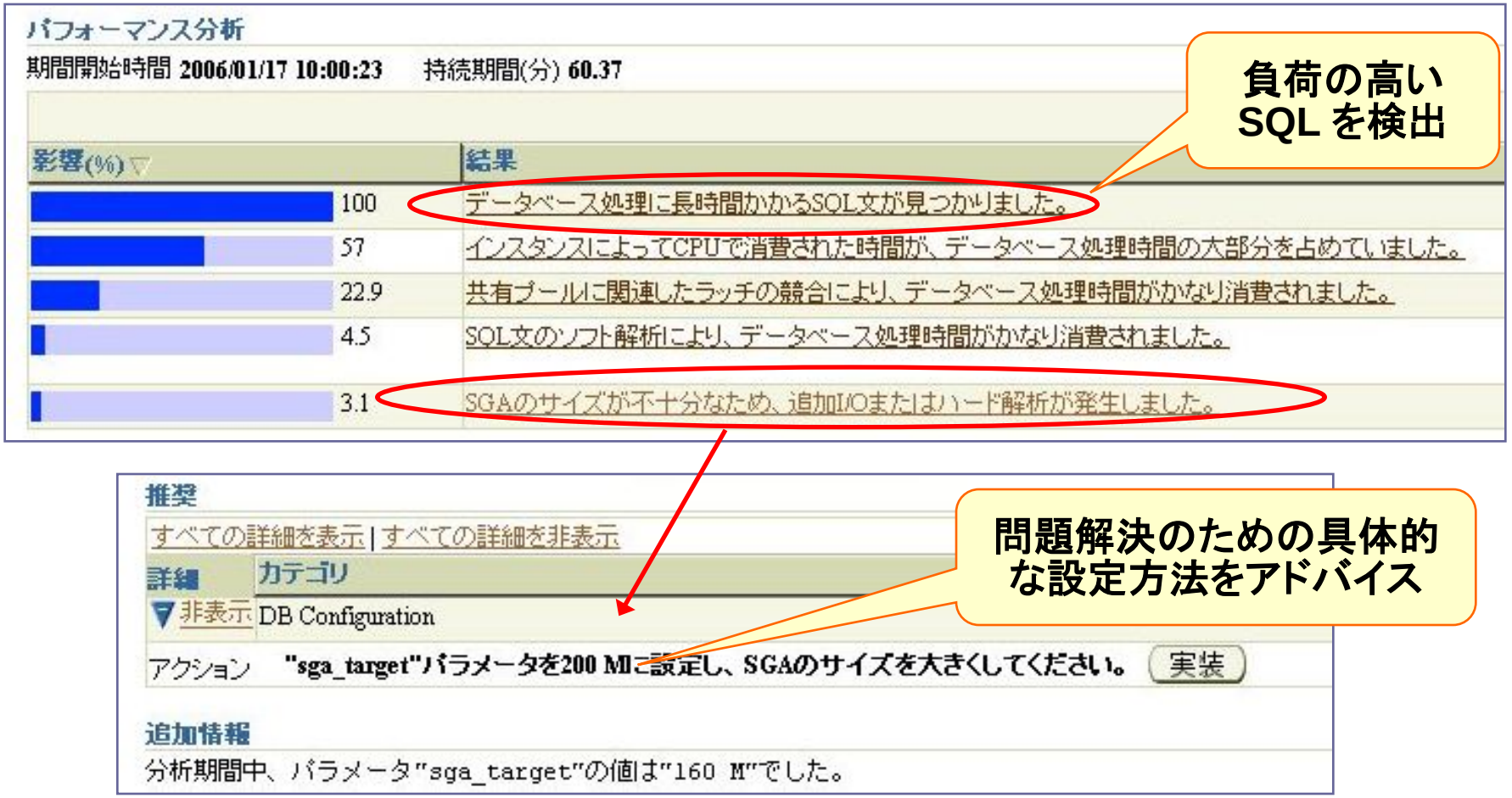
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

ADDM によるアドバイス

アシスト

EE + Diag Pack

- AWRに収集されたデータを分析し、定期的に診断を実行
- 診断結果として、アドバイザの実行などの解決方法を Webコンソール に表示



ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

ADDM診断レポート例

アシスト

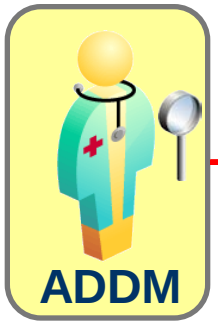
ボトルネック切り分け
チューニングアドバイス

EE + Diag Pack

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

アドバイザ機能

EE ＋
Diag Pack



DB全体の
監視・診断

- SQLチューニング・アドバイザ：特定のSQL文の診断・チューニング（SQL文の変更、プロファイルの作成 等の推奨）
- SQLアクセス・アドバイザ：システム全体の索引やマテリアライズド・ビュー、パーティショニング(11g～)に関する診断
- メモリー・アドバイザ：PGA、バッファ・キャッシュ、共有プールのサイズ変更をアドバイス
- セグメント・アドバイザ：セグメントの断片化レベルを特定し、縮小すべきセグメントをアドバイス
- UNDOアドバイザ：最適なUNDOの保存期間やUNDO表領域のサイズを推奨

・・・ Tuning Pack
・・・ Pack不要(SE,SE Oneで利用可)

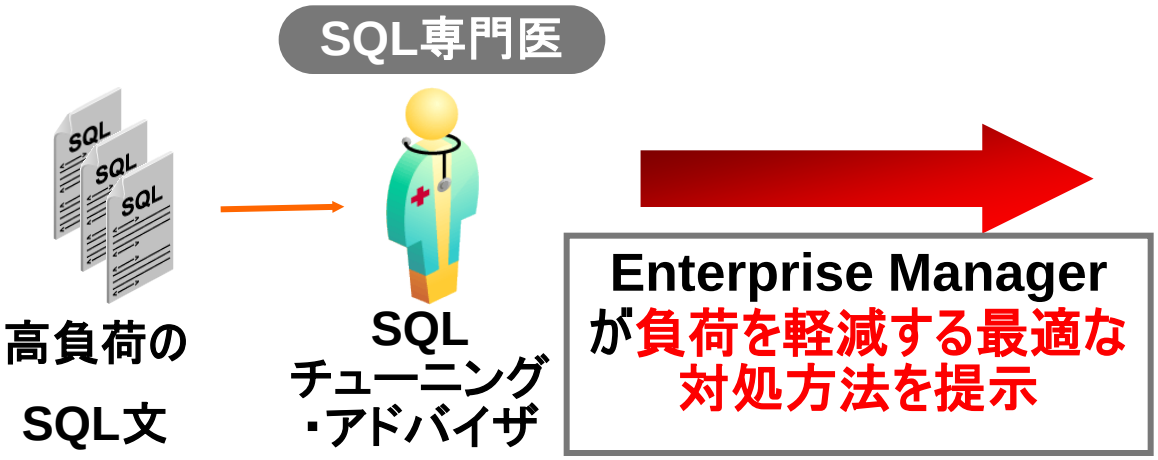
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

SQLチューニング・アドバイザ

アシスト

EE ＋
Diag Pack
＋
Tuning Pack

- 高負荷で問題となるSQL文や実行計画を診断する
- 診断をもとにアドバイス



失効・欠落している
統計の収集

索引の作成

SQL文の
再構成

SQLプロファイ
ルの作成

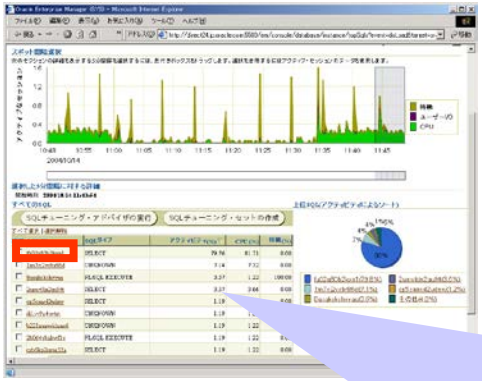
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

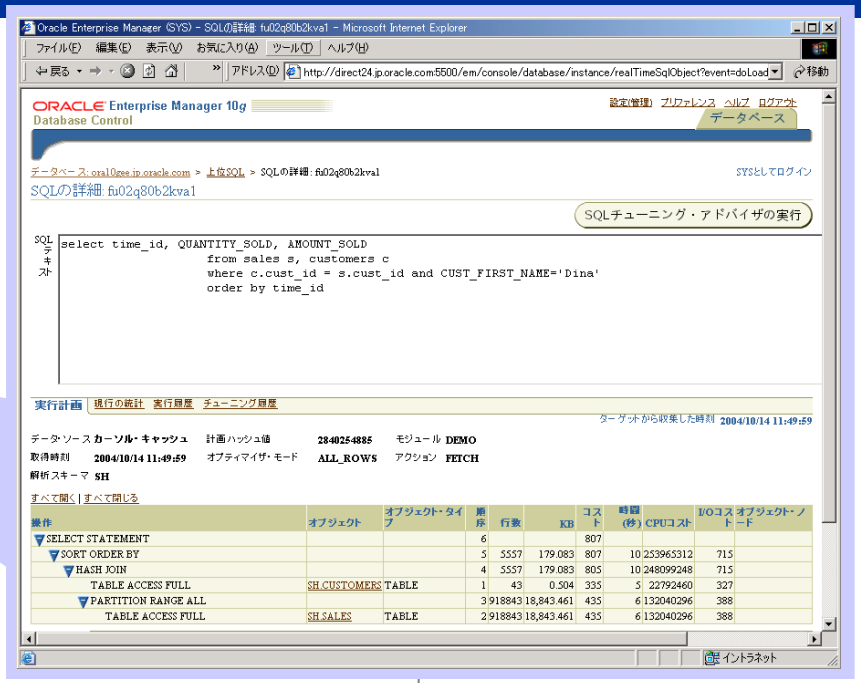
実行例

アシスト

負荷の高いSQLの

実行計画を確認し・・





実行計画 | 現行の統計 | 実行履歴 | チューニング履歴

データ・ソース カーソル・キャッシュ　計画ハッシュ値 2840254885　モジュール DEMO

取得時刻 2004/10/14 17:35:16　オプティマイザ・モード FIRST_ROWS　アクション FETCH

解析スキーマ SH

すべて開く | すべて閉じる

操作	オブジェクト	オブジェクト・タイプ	順序	行数	KB	コスト	時間(秒)	CPUコスト	I/Oコスト
SELECT STATEMENT			6			806			
SORT ORDER BY			5	822	26.49	806	10	251209328	715
HASH JOIN			4	822	26.49	805	10	248099248	715
TABLE ACCESS FULL	SH.CUSTOMERS	TABLE	1	43	0.504	335	5	22792460	327
PARTITION RANGE ALL			3	918843	18,843.461	435	6	132040296	388
TABLE ACCESS FULL	SH.SALES	TABLE	2	918843	18,843.461	435	6	132040296	388

新規実行計画

すべて開く | すべて閉じる

操作	ラインID	順序	行数	KB	コスト	時間(秒)	CPUコスト	I/Oコスト	オブジェクト	オブジェクト・タイプ
SELECT STATEMENT	0	6	4	0.129	7	1	76434	7		
NESTED LOOPS	1	5	4	0.129	7	1	76434	7		
TABLE ACCESS BY GLOBAL INDEX ROWID	2	2	918843	18,843.461	4	1	29895	4	SH.SALES	TABLE
INDEX FULL SCAN	3	1	3		3	1	21964	3	SH.SALES_TIME_IDX	INDEX
TABLE ACCESS BY INDEX ROWID	4	4	1	0.012	1	1	15512	1	SH.CUSTOMERS	TABLE
INDEX UNIQUE SCAN	5	3	1		0	1	8171	0	SH.CUSTOMERS_PK	INDEX (UNIQUE)

SQLチューニングを

実行！

EE ＋

Diag Pack ＋

Tuning Pack

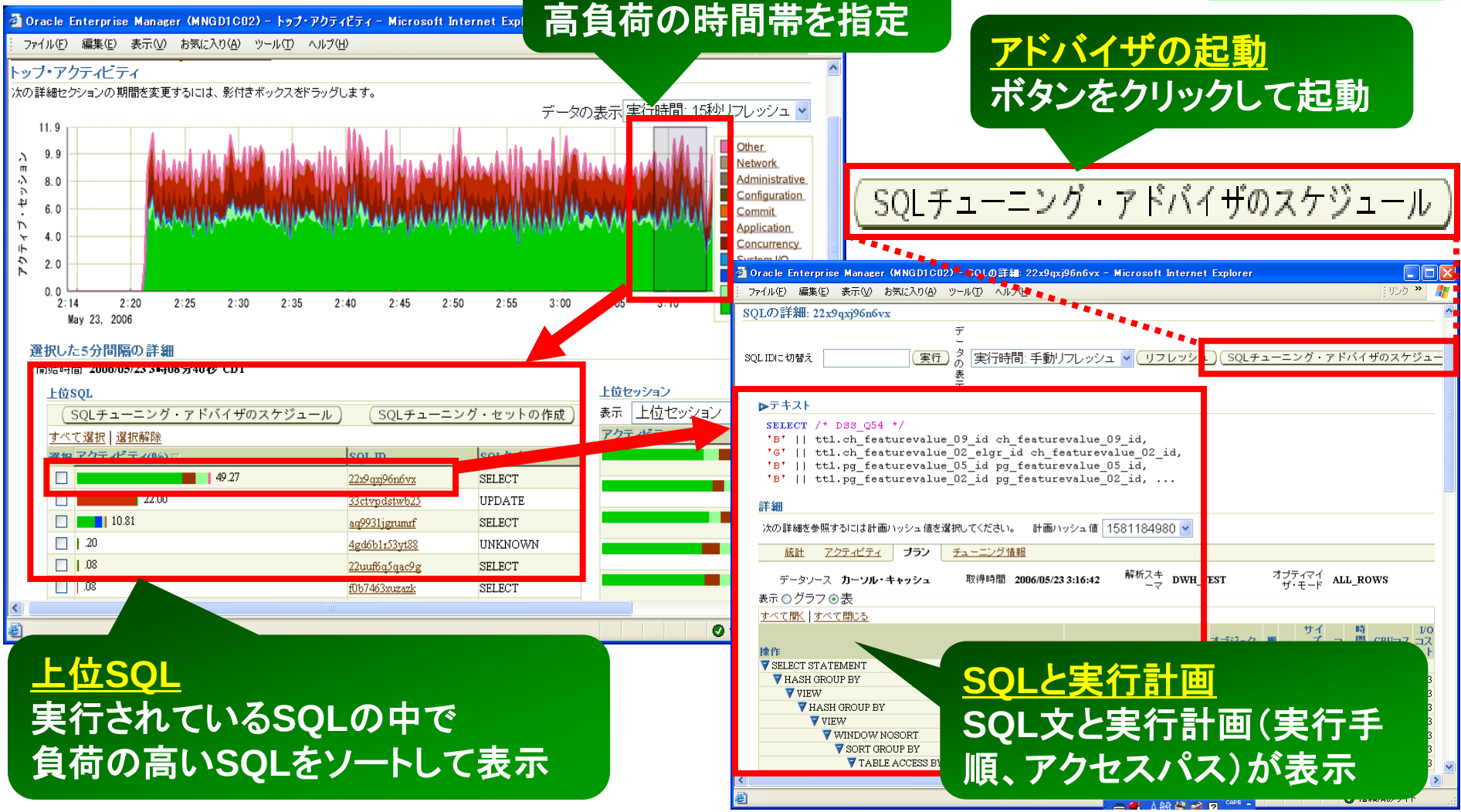
ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

負荷の高い時間帯で実行されているSQLの特定 EE + Diag Pack + Tuning Pack

負荷状況の確認
高負荷の時間帯を指定

アドバイザの起動
ボタンをクリックして起動



上位SQL
実行されているSQLの中で
負荷の高いSQLをソートして表示

SQLと実行計画
SQL文と実行計画（実行手順、アクセスパス）が表示

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

チューニングの流れ

EE + Diag Pack + Tuning Pack

自動ワークロード・リポジトリ（AWR）
データベース稼動状況の履歴
アドバイザなどが、診断 / 分析時に参照

①DBの稼動状況を収集

Automatic Database Diagnostics Monitor（ADDM）
データベース全体の稼動状況をモニタ、診断

②収集情報をもとに、全体としての診断
時間モデル統計に基づき、全体の中で時間がかかっている個所を把握し、対応策を提示

各種アドバイザ
データベースを診断し、最適な設定や問題点を解消する方法を提示、または設定に有用な見積りを表示（SQL、メモリー、セグメント、UNDOなど）

③さらに個々の点について診断

解決策の適用（管理者）
アドバイザが提示した解決策を適用するか決定

④解決策を適用
最終的には管理者が判断できる

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

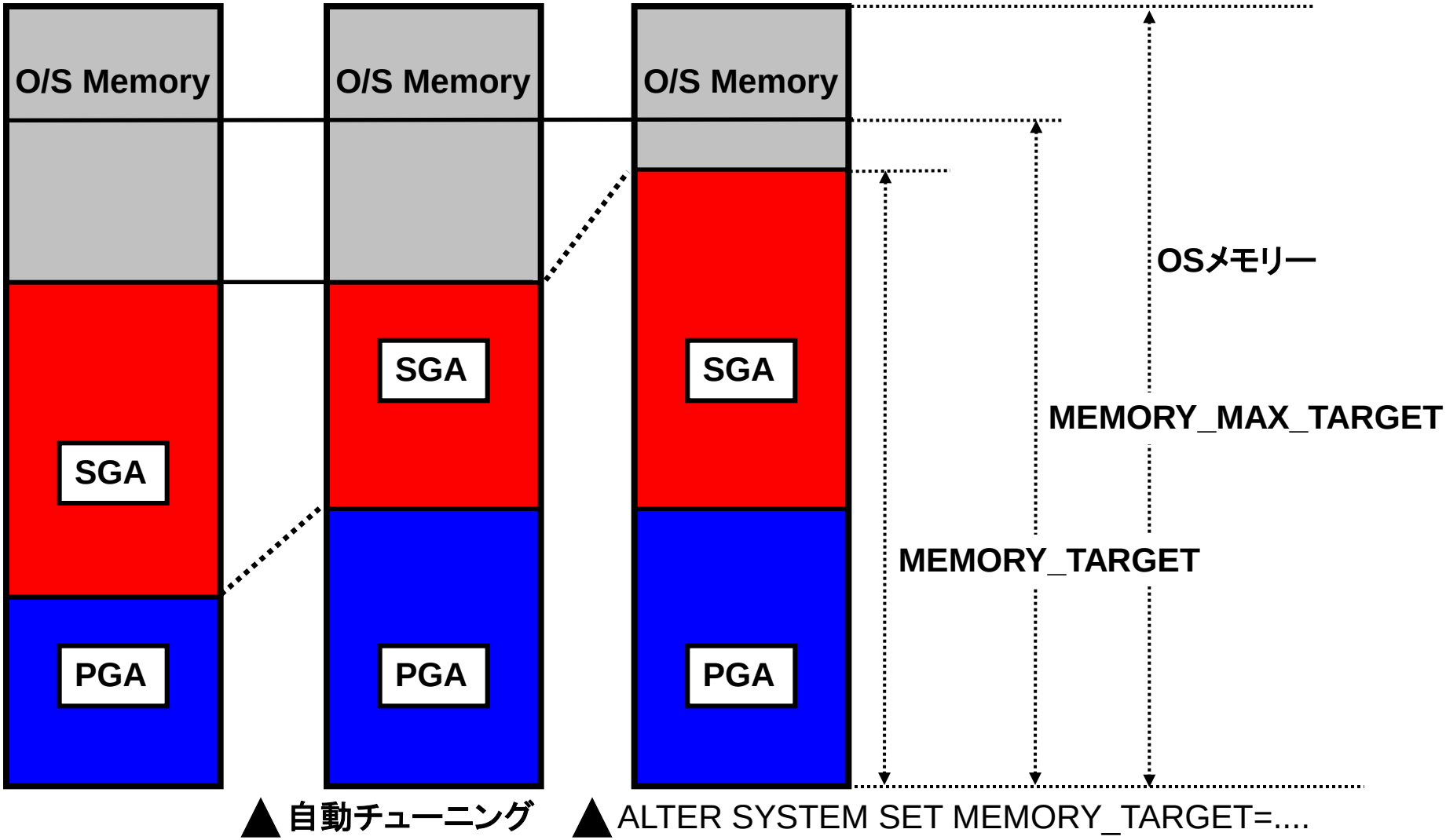
メモリアドバイザ

アシスト

- 最適なメモリ量の割り当て（バッファ・キャッシュ・サイズ）

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

自動メモリー管理（11g）



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

「リアルタイムSQL監視」

EE + Diag Pack + Tuning Pack

- 「リアルタイムSQL監視」とは
  - 実行中のSQLを**自動で監視**し、詳細な統計を取得
  - EMの**グラフィカルなレポート画面**から分析ができる
  - Oracle Database 11gからの新機能
  - Tuning Packで提供

- 特長
  - GUIから**簡単にボトルネックを突き止められる**
  - 再現待ちや特別な設定をせず**すぐに分析を始められる**
  - レポートをエクスポートして外部で参照可能
  - オーバーヘッドがほとんどない

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長1: GUIからのボトルネック分析
分析するSQLの特定

EE + Diag Pack + Tuning Pack

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長1: GUIからのボトルネック分析
SQLの実行状況の詳細確認(1)

EE + Diag Pack + Tuning Pack

SQL実行全体の期間(経過時間)のほか、DB時間や待機イベントの内訳を把握できる

バッファ読取りよりI/Oバイト数が大幅に多い場合は、ストレージがボトルネックになる傾向あり

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長1: GUIからのボトルネック分析
SQLの実行状況の詳細確認(2)

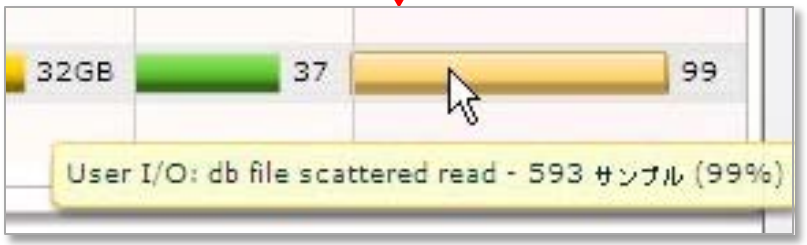
EE + Diag Pack + Tuning Pack

各ステップごとのメモリ(PGA)、一時表領域の使用状況

各ステップごとの実行タイミングや実行時間など
(ここではITEM表、DATE_DIM表、STORE_SALES表の順に読み取りながら結合している)

実行計画で予想された行数と実際に返された行数の比較も容易

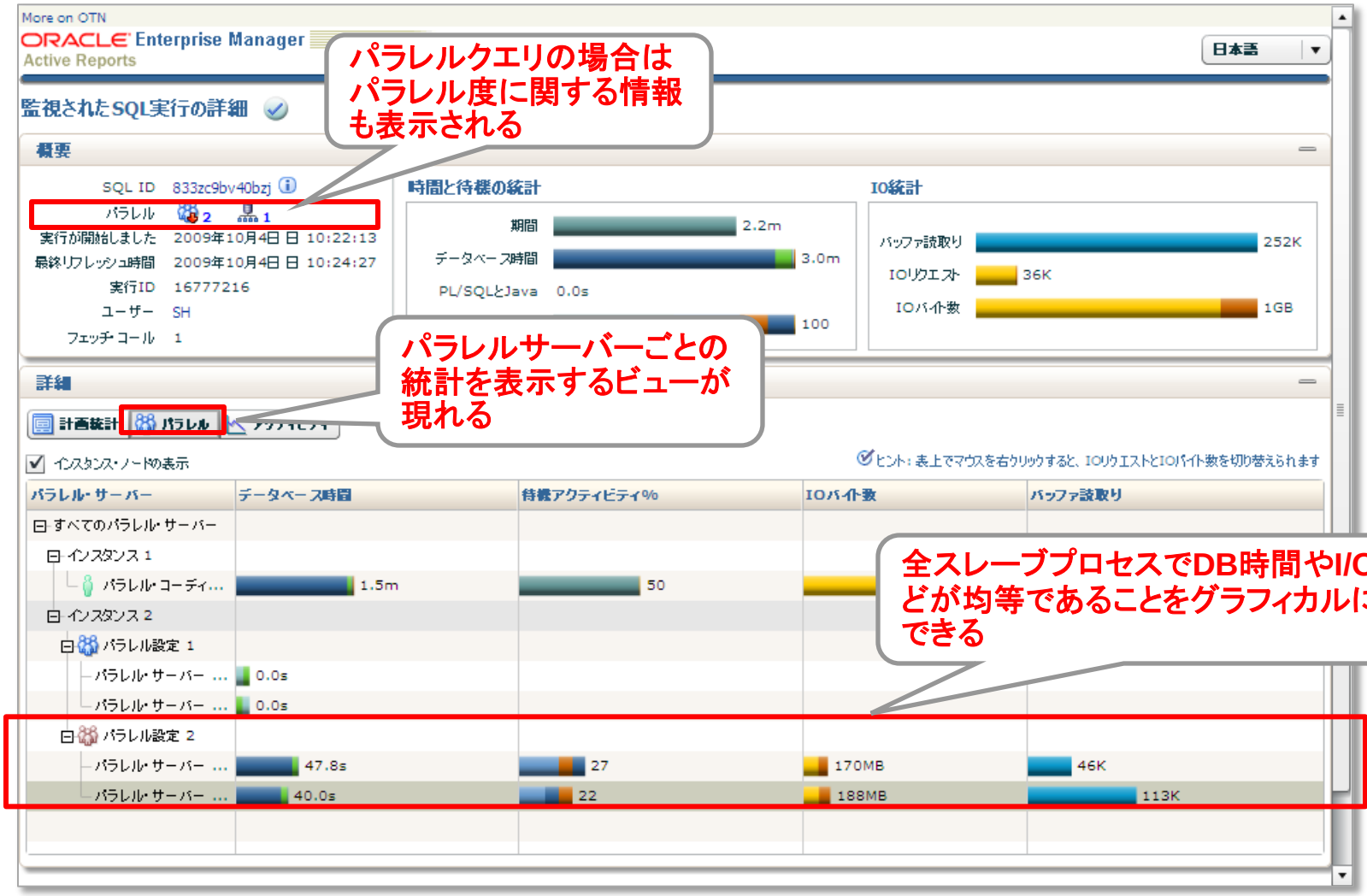
各ステップごとの待機イベントの発生個所やその内訳も簡単にわかる



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長1: GUIからのボトルネック特定 パラレルクエリの実行状況

EE + Diag Pack + Tuning Pack



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長1: GUIからのボトルネック特定
実行中のデータ参照(「今ここ！」マーク)

EE + Diag Pack + Tuning Pack

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

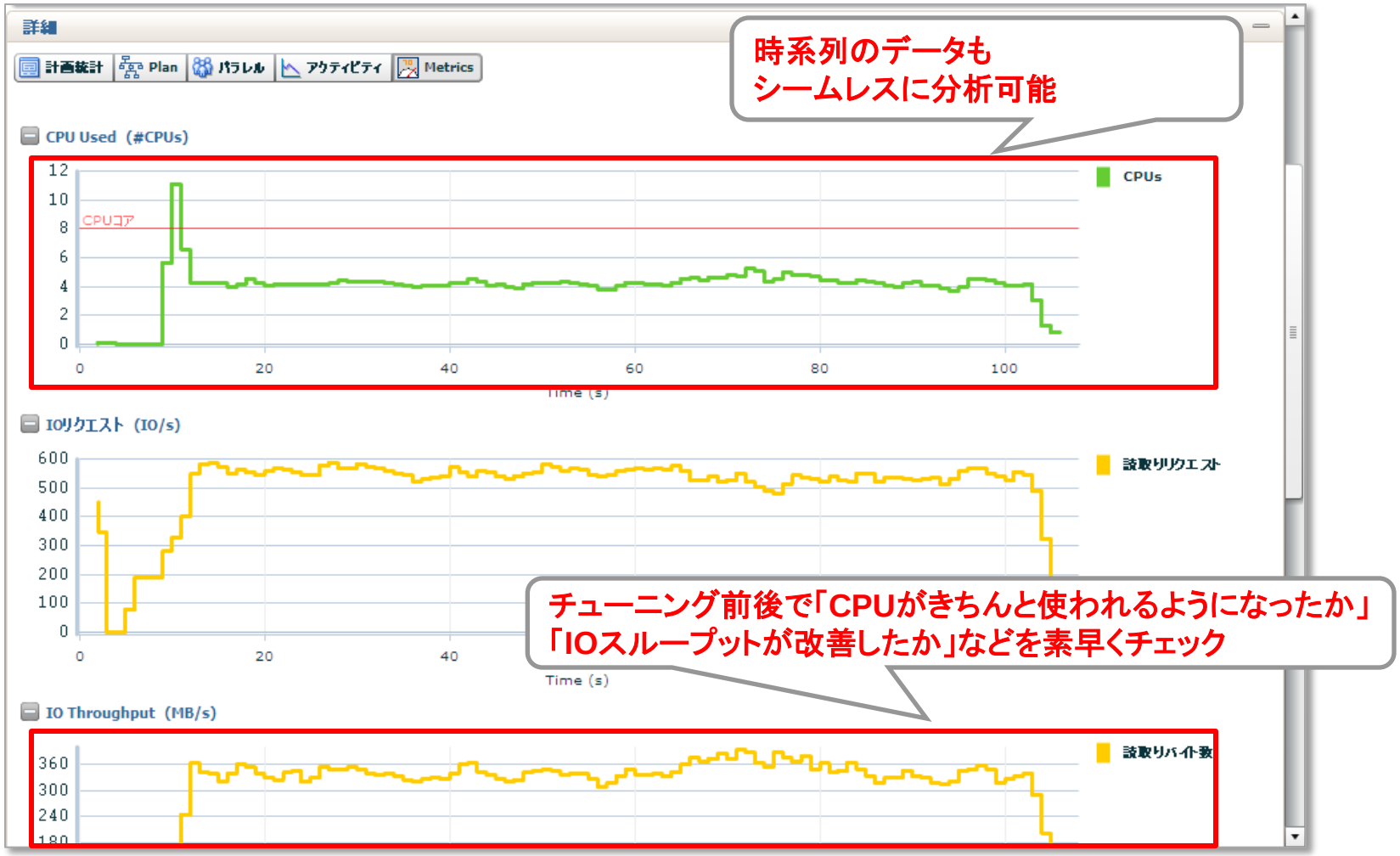
特長1: GUIからのボトルネック特定
SQL全文とバインド変数の参照

EE + Diag Pack + Tuning Pack

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長1: GUIからのボトルネック特定
時系列グラフ

EE + Diag Pack + Tuning Pack



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長2: 再現待ちや特別な設定が不要

EE + Diag Pack + Tuning Pack

- SQL監視は自動的に実行される
- 自動実行される条件
  - 5秒以上のCPU時間またはI/O時間を消費しているSQL
    または
  - パラレル実行されているSQL
- 手動実行する方法
  - ヒント句 /*+MONITOR*/

[参考] Oracle Databaseパフォーマンス・チューニング・ガイド
10.4 リアルタイムSQL監視
http://download.oracle.com/docs/cd/E16338_01/server.112/b56312/instance_tune.htm#CACGEEIF

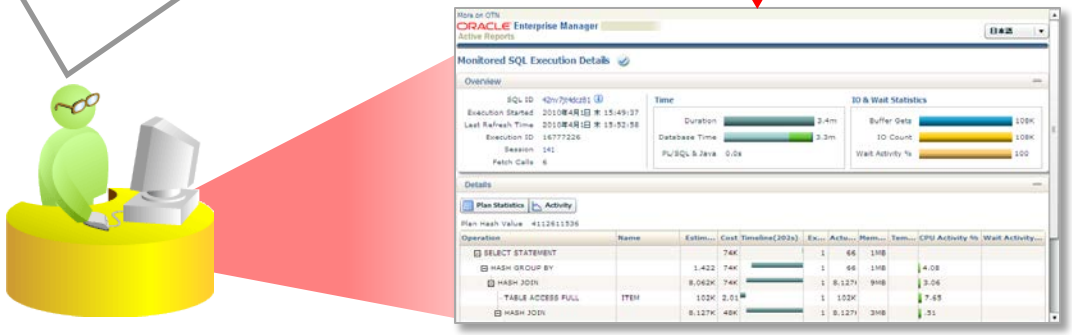
Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長3: HTMLでのエクスポート

EE + Diag Pack + Tuning Pack

HTMLファイルにエクスポートすることで
環境に直接アクセスできない場所でも
分析可能

ローカル(ブラウザを起動しているマシン)にHTMLで保存可能



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

特長4: オーバーヘッドがほとんどない

EE + Diag Pack + Tuning Pack

- 監視情報はV$表に保存される
  - V$SQL_MONITOR, V$SQL_PLAN_MONITOR (11gで追加)
  - V$SQLのサブセット
  - 単一実行の統計(複数実行の累積や平均ではない)
  - ディスクI/Oを伴うSQLトレースと違い、SGA上に格納される

```
SQL> select sql_id, sql_exec_start, last_refresh_time, sql_exec_id,
sid, fetches from v$sql_monitor where sql_id= '42nv7jt4dcz81' ;

SQL_ID         SQL_EXEC LAST_REF SQL_EXEC_ID    SID FETCHES
-------------- -------- -------- ------------ ------ --------
42nv7jt4dcz81  10-04-01 10-04-01     16777216    170        6
```

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

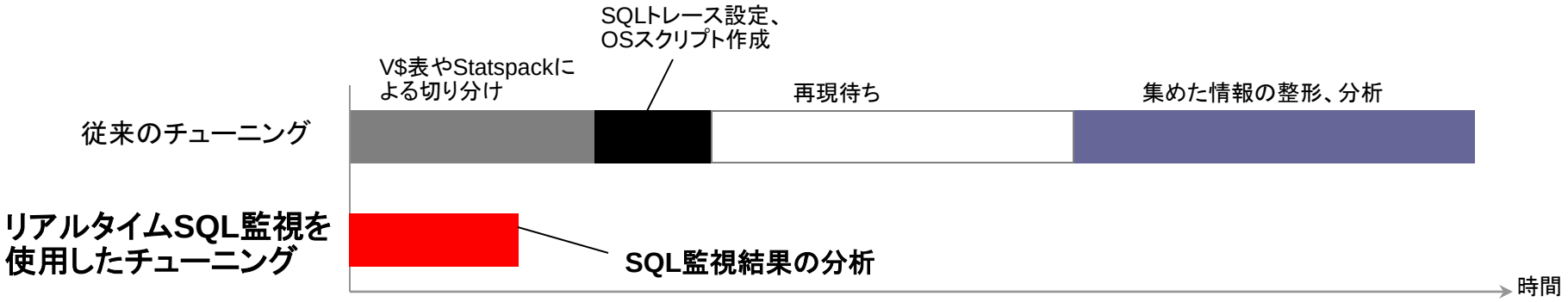
期待される導入効果

EE + Diag Pack + Tuning Pack

開発・テスト時からのチューニング精度の向上

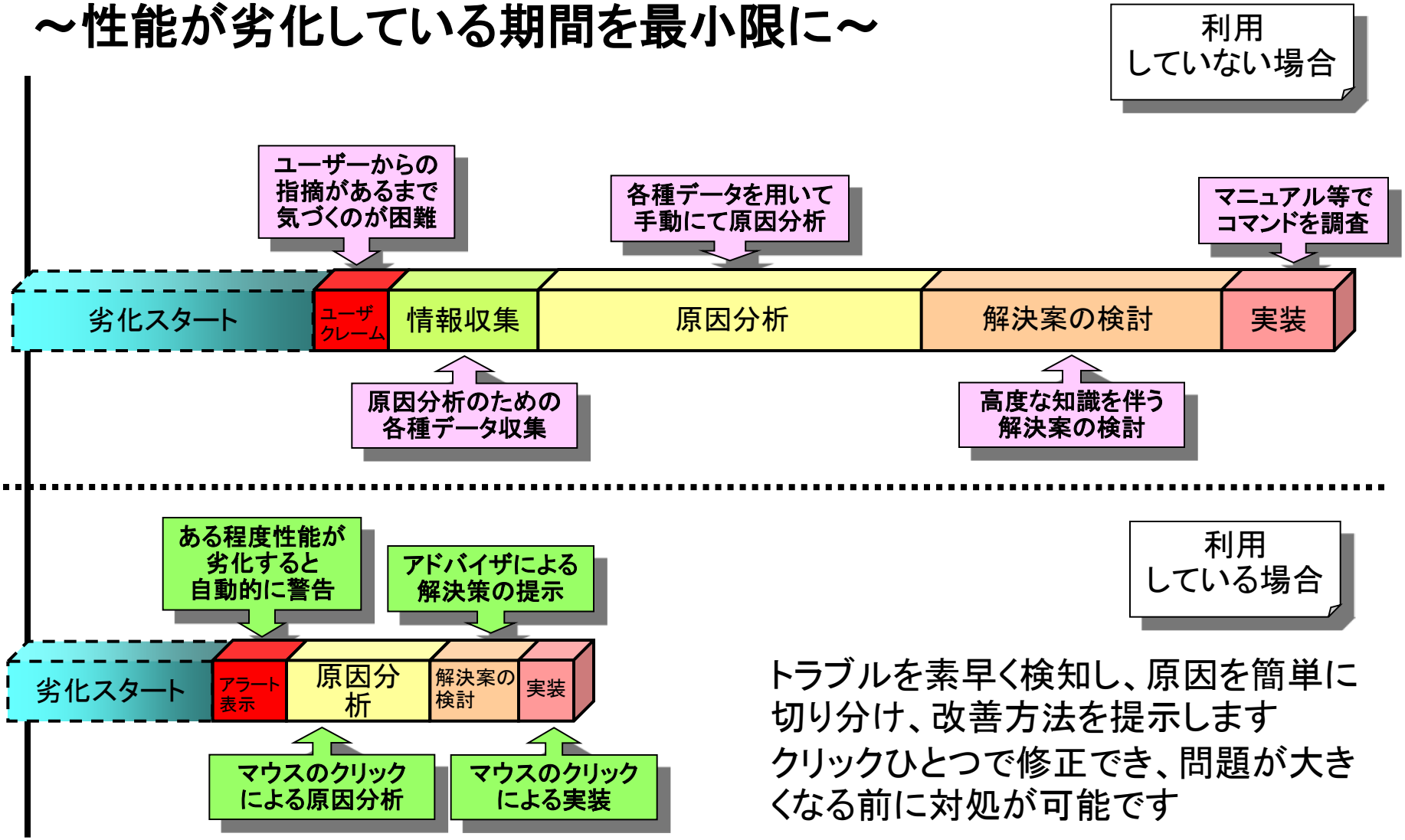
カットオーバー後に発生した性能問題の迅速な解決

長時間かかるSQL(バッチなど)への正確な対処



Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

～性能が劣化している期間を最小限に～



トラブルを素早く検知し、原因を簡単に切り分け、改善方法を提示します
クリックひとつで修正でき、問題が大きくなる前に対処が可能です

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

目次

- パフォーマンス・チューニングの5W+1H
  1. Why ～なぜ実施するの？～
  2. When ～いつ実施するの？～
  3. Who ～誰が実施するの？～
  4. Where ～どこでボトルネックがおこるの？～
  5. What&How ～何をどうすれば早くなるの～
     - 5-1. ボトルネックの特定
     - 5-2. 対処
       - SQLチューニング
       - 索引の使用
       - チューニング効果の確認
     - 5-3. 分析の自動化とアドバイス機能
- まとめ

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

まとめ

アシスト

パフォーマンス問題を解決するための、
「分析」と「対処」を実施

1. 性能問題となっている箇所を特定
2. チューニングを行う

Statspack の使用や
AWR やSQL トレースの活用

Oracle Database（DB）の
性能問題ではないかを確認

DB側の性能問題
ではない場合

DB側の性能問題
であった場合

Application（AP）か
ネットワークの性能問題かを確認

DBのチューニングを行う

ネットワーク環境を改善

APのチューニングを行う

SQL チューニング、索引
ADDM やアドバイザの活用
・・・・

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Appendix

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

Statspack使用手順

アシスト

- Statspackの使用手順は以下の通り

Step1　インストール・設定

  - Statspack初回利用時のみ、実行用ユーザを作成し、同時にパスワードや表領域を設定

Step2　スナップショット取得

  - 状況によって自動取得やスナップショットレベルを変更する

Step3　レポート作成

  - 開始と終了スナップショットを選び、ファイル名を入力する

Statspackを使用するためのスクリプトは、DBを導入した環境にすでに用意されているため、DBに接続しスクリプトを実行するだけで使用できる

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

インストールスクリプトを実行

アシスト

Step1 インストールと設定 | Step2 スナップショット取得 | Step3 レポート作成

Statspackに関する操作は、全て実行用ユーザ（PERFSTAT）で行うため、
インストール・スクリプトを実行してPERFSTATユーザを作成する

1. SQL*PlusでSYSDBA権限を持つユーザとしてDBに接続
2. インストール・スクリプト（spcreate.sql）を実行

（例）

```
# SQL*PlusにSYSDBA権限で接続
SQL> connect sqlplus / as sysdba

# インストール・スクリプトを実行
SQL> @$ORACLE_HOME/rdbms/admin/spcreate.sql
```

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

パスワードと表領域の設定

アシスト

Step1 インストールと設定 | Step2 スナップショット取得 | Step3 レポート作成

PERFSTATユーザのパスワードとStatspackで使用するデフォルト表領域と一時表領域を設定する

1. スクリプトを実行するとPERFSTATユーザのパスワード、デフォルト表領域、一時表領域の入力を求められるので適宜入力

（例）

```
# 実行ユーザのパスワードを設定
perfstat_passwordに値を入力してください: <passwordを入力>

# デフォルト表領域を設定
default_tablespaceに値を入力してください: <表領域名を入力>

# 一時表領域を設定
temporary_tablespaceに値を入力してください: <表領域名を入力>
```

デフォルト表領域は余裕を持ち300MBに設定することを推奨

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

スナップショットの取得

アシスト

Step1 インストールと設定 | Step2 スナップショット取得 | Step3 レポート作成

PERFSTATユーザでDBに接続し、スナップショットで統計情報を取得する

1. PERDSTATユーザでstatspack.snapを実行
2. 必要なら取得する統計情報の詳細さ(レベル)を i_snap_level => で　設定(ユーザがレベルを指定しない場合には、デフォルト値のレベル5で取得される)

(例)

```
# スナップショットの取得
SQL> execute statspack.snap(I_snap_level => 6);
```

- 繰り返しスナップショットを記録し、ある2時点でのスナップショットの差分をレポートとして出力することで、特定の期間のDBの統計情報を取得できる

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

レポートの作成

アシスト

Step1 インストールと設定 | Step2 スナップショット取得 | Step3 レポート作成

レポート作成スクリプトを実行し、レポートを生成する

1. PERDSTATユーザでspreport.sqlを実行

（例）

```
SQL> @?/rdbms/admin/spreport.sql     # レポート作成スクリプトを実行
                                     :
        ～省略～                                                    Snap
Instance   DB Name      Snap Id   Snap Started                      Level   Comment
---------- ------------ ---------- ------------------------------ --------- --------------
orcl1      ORCL              1    17 10月 2008 12:17                     5
                            11    20 10月 2008 09:29                     5
                                     :
Specify the Begin and End Snapshot Ids
~~~
```

スナップショットIDの入力

アシスト

Step1 インストールと設定 | Step2 スナップショット取得 | Step3 レポート作成

統計情報を取得する範囲を開始および終了時点のスナップショットID指定し、レポートを出力する

1. Snap Idを参考に開始スナップショットと終了スナップショットIDを入力
2. レポート名を入力

（例）

```
Specify the Begin and End Snapshot Ids
~~~
```

スナップショットの削除

アシスト

取得したパフォーマンス統計情報は、明示的に削除しない限り
いつまでも残り、場合によっては領域を圧迫してしまう

よって不要になった統計情報は適宜削除を行う

1. PERFSTATユーザでsppurge.sqlを実行
2. 削除したい範囲の開始と終了点にあたるスナップショットIDを指定

（例）

```
SQL> @?/rdbms/admin/sppurge.sql      # 削除するスクリプトを実行

Specify the Lo Snap Id and Hi Snap Id range to purge
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

losnapidに値を入力してください: 14          # 開始点のスナップショットIDを入力

hisnapidに値を入力してください: 19       # 終了点のスナップショットIDを入力
```

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

アシストのOracleビジネスのご紹介

アシスト

Oracleの実績

【Oracle受賞歴】 Oracle Award受賞

Oracle Award 2009

[DODAI] Platform Solution Award

【過去受賞歴】

パートナー別販売実績第1位
Excellent Partner受賞(9年連続 9回)
Best Partner賞
Support of The Year賞
Show case of the Year賞
Oracle Real Application Clusters部門
Best Area Performance of the Year賞
Oracle Database 11g Award賞
KUDOS for Oracle Support Partners賞

【ACSP認定】高レベルサポート認定企業

【Oracle販売実績】

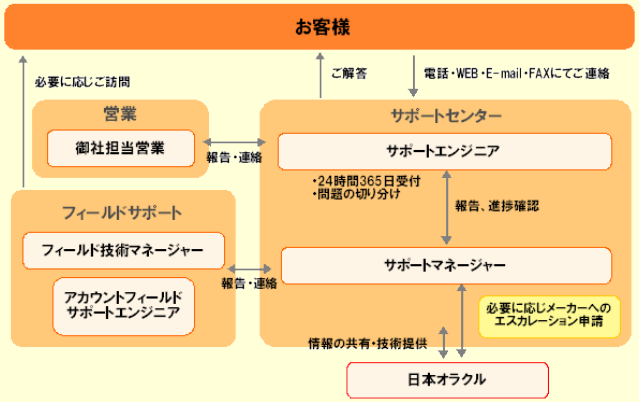
年間売上	約80億円(国内トップクラス)
お取引き企業数	8,000社以上(国内トップクラス)
サポート契約継続率	90%以上
過去3年のサポート切替実績	1,200件以上

【歴史】

1987年、アシストグループとして株式会社オラクルを設立し、国内で最初にOracleのライセンス販売、サポートの提供をはじめました。
誠実にお客様の声に耳を傾け、真心を持ってお答えするという姿勢と、万全のサポート体制でお客様の期待に答え続け、多くのお客様より厚くご支持を頂けております。

Oracleのサポート

24時間365日、約200名のOracle専属技術者による全国サポート体制(国内最大級)



○ 信頼のフィールドサポート

サポートセンターで解決できない問題は、Oracle専任技術者が直接お客様のもとをお伺いし、解決にあたります。

過去3年(年間平均)の対応状況

サポート件数	約10,000件/年
フィールド対応件数	約400件/年
顧客満足度	約90%

○ マルチベンダー対応

アシストではすべてのOSで実機を用意し、構築スキルを蓄え、検証環境を整えております。

redhat AIX intel solaris HP-UX 11i

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

アシストのOracleビジネスのご紹介

アシスト

豊富な支援実績

西日本支社のみで約25名のOracle専門部隊がお客さまの課題解決にあたり、年間100件を超える技術支援を実施しております。

さまざまな課題やご要望に対して、柔軟かつ高品質なサービスをご提供することが可能です。

技術支援の一例

A社(卸売業)	RHEL/10g R2	基幹系システム再構築	Real Application Clusters、バックアップ/リカバリ、データ移行
B社(陸運業)	Win/10g R2	業務系システム再構築	Oracle Fail Safe、データ移行、バックアップ/リカバリ
C社(電気・ガス業)	AIX/10g R2	基幹系システム再構築	HA、バックアップ/リカバリ、教育
D社(電気機器)	Win/11g R1	情報系システム構築	Single、DB設計、チューニング
E社(証券業)	RHEL/9i R2	災対サイト構築	Data Guard、バックアップ/リカバリ
F社(情報・通信)	HP-UX/10g R2	情報系システム再構築	Real Application Clusters、バックアップ/リカバリ、パーティション設計
G社(電気機器)	AIX/10g R2	基幹系システム再構築	パーティション設計、バックアップ/リカバリ

技術支援サービスメニュー

- 導入作業
- バージョンアップ作業
- データベース移行
- Oracle データベース設計/実装
- バックアップ/リカバリ運用設計
- バックアップ/リカバリ検証
- データベース診断
- 監視設計
- SQL チューニング
- パーティショニング設計
- Real Application Clusters(RAC)環境構築
- Data Guard 環境構築
- その他技術支援(オンサイト技術支援)
- 各種オンサイト教育

ORACLE

Copyright© 2012, Oracle. & K.K.Ashisuto All rights reserved.

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Siebelは、米国オラクル・コーポレーション及びその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標の可能性があります。

ご清聴ありがとうございました。

株式会社アシスト　西日本支社