Oracle Direct Seminar

ORACLE 11g DATABASE

ORACLE®

今さら聞けない!?
パフォーマンス・チューニング入門

日本オラクル株式会社
Oracle Direct

Oracle Direct Seminar

Agenda

- パフォーマンス・チューニングとは
- ボトルネック箇所の特定
- 代表的なチューニング項目
  - メモリ割り当てのチューニング
  - ディスクI/Oのチューニング
  - SQL文のチューニング

Agenda

- パフォーマンス・チューニングとは
- ボトルネック箇所の特定
- 代表的なチューニング項目
  - メモリ割り当てのチューニング
  - ディスクI/Oのチューニング
  - SQL文のチューニング

パフォーマンス・チューニングに必要なこと
要件定義、設計段階から意識すること

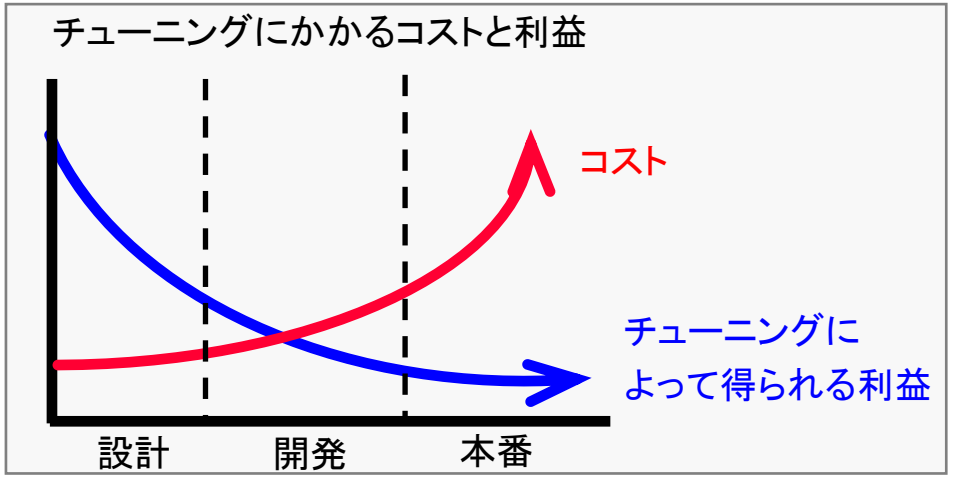
- **要件定義、設計段階からパフォーマンスを意識することが重要**
  - 「〇分以内」等の要件を決定
  - 設計、開発、本番稼動後の格段階で、各担当者がパフォーマンスを考慮

メモリが足りなかった！
サーバーを停止して
メモリを追加しなきゃ・・・

結合のパフォーマンスを
あげるために表を1つにしよう・・・
でもアプリケーションも変えないと

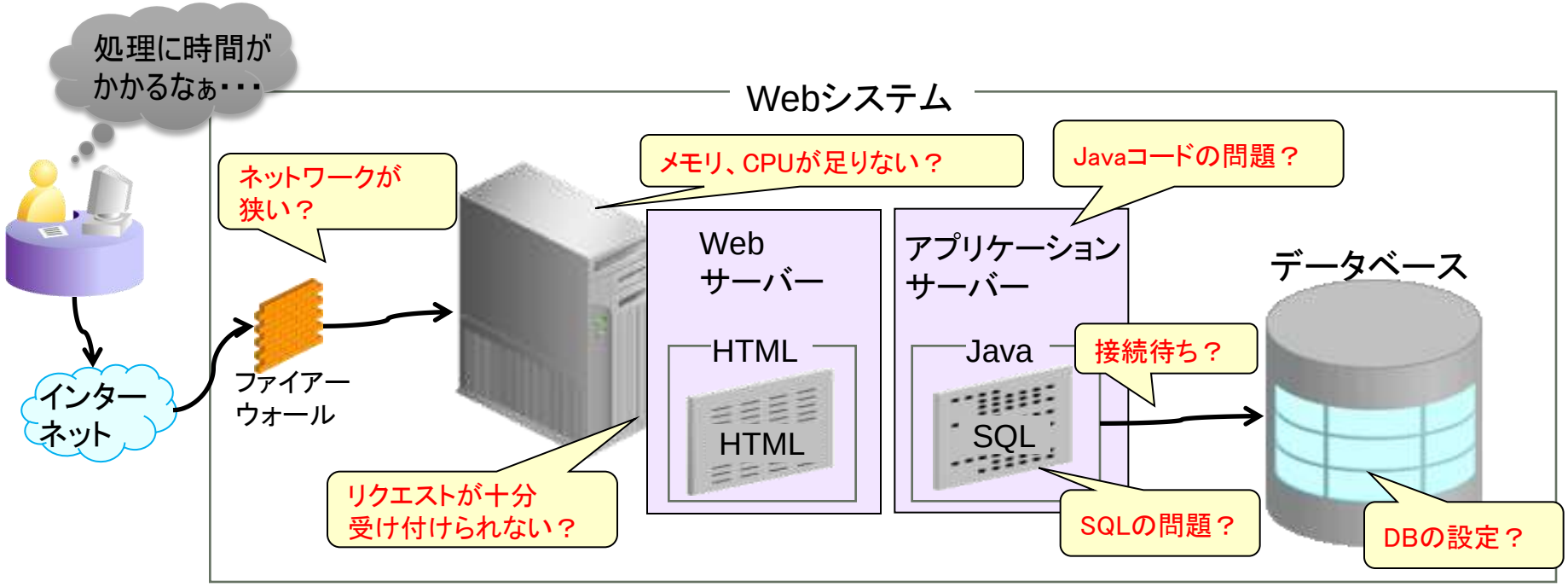
アプリケーションの変更が
必要になったので、
また開発者に依頼しなきゃ・・・

後から実施するチューニング作業は、
労力とコストに対して効果が得にくい傾向にある



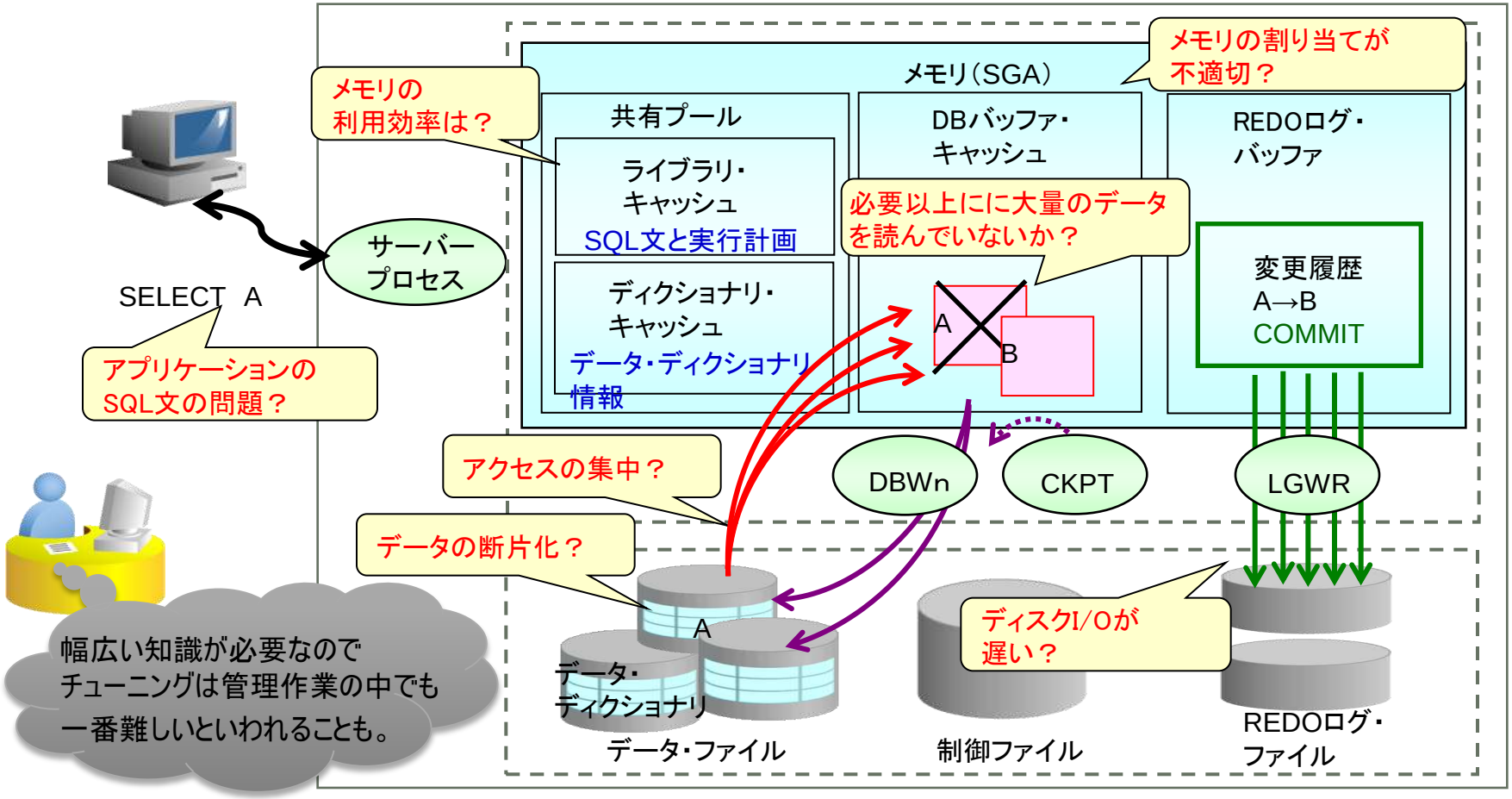
パフォーマンス・チューニングに必要なこと
システム全体からボトルネックを特定して改善すること

- システム全体を考慮して、ボトルネック箇所を特定することが必要
  - Webシステムが複雑化するにつれ、問題の切り分けは困難になりがち
  - ボトルネック箇所に対して、適切なチューニングを行う必要がある
  - アプリケーションやネットワークに問題があるのにデータベースをチューニングしても効果はない



パフォーマンス・チューニングに必要なこと
データベース内のボトルネックを特定して改善すること

- データベース内のボトルネックを特定し、ボトルネック箇所を改善

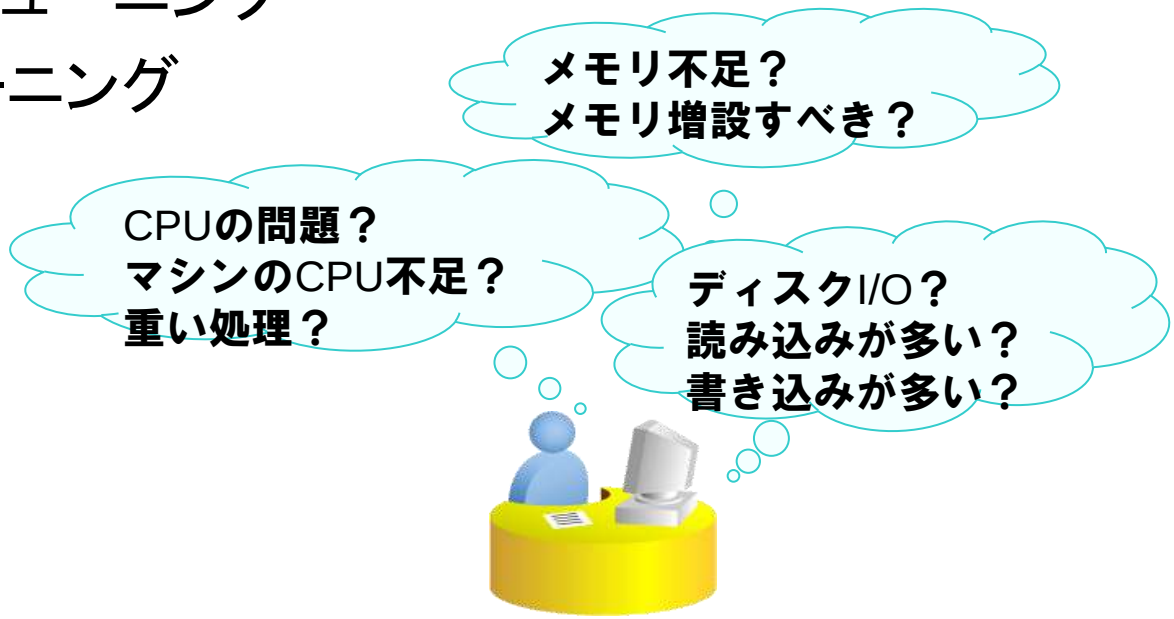


パフォーマンス・チューニングのポイント

- **パフォーマンス・チューニングのポイント**
  - パフォーマンス要件を定義し、要件を満たすような設計、開発、運用をすること
  - ボトルネック箇所を特定し、ボトルネックに応じた改善策をとること
    - サーバの性能以上の処理を行うことはできない
    - 処理待ちが発生している部分（ボトルネック）に対して、待機を減らすことが重要
    - ボトルネックになりやすい個所
      - メモリ割り当ての改善
      - ディスクI/O（ディスクからの読み込み）の改善
      - 実行されるSQL文の効率化

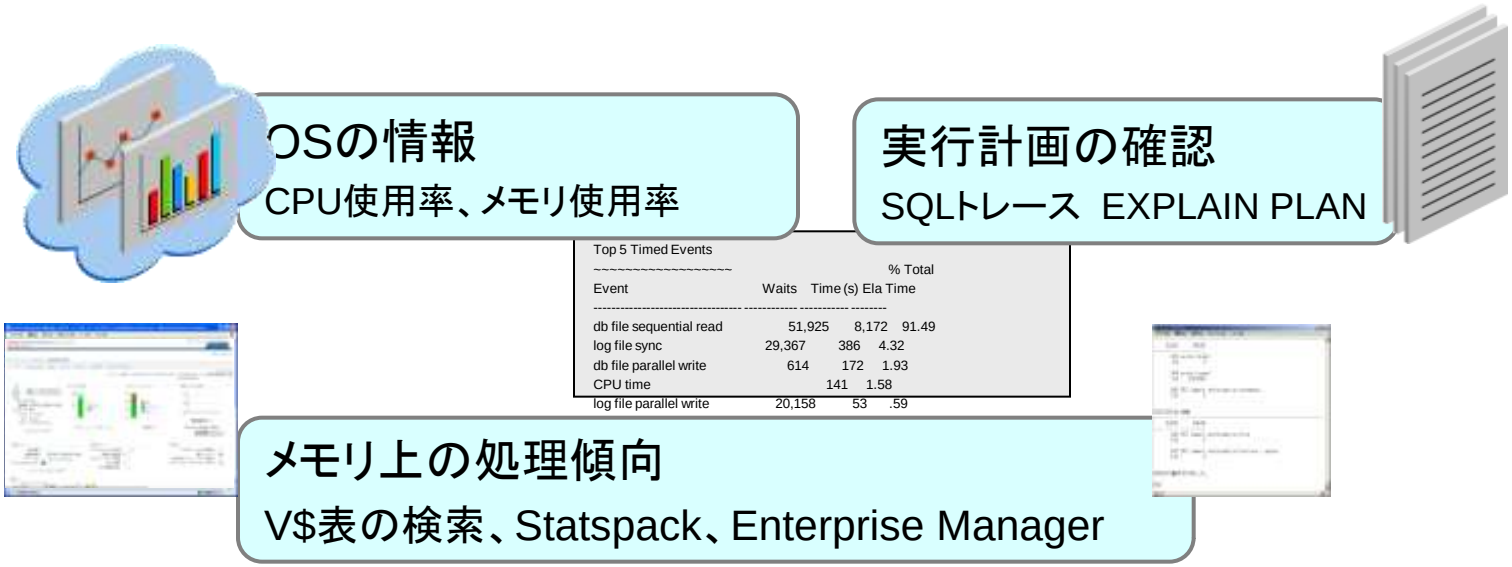
Agenda

- パフォーマンス・チューニングとは
- ボトルネック箇所の特定
- 代表的なチューニング項目
  - メモリ割り当てのチューニング
  - ディスクI/Oのチューニング
  - SQL文のチューニング



ボトルネック箇所の特定方法

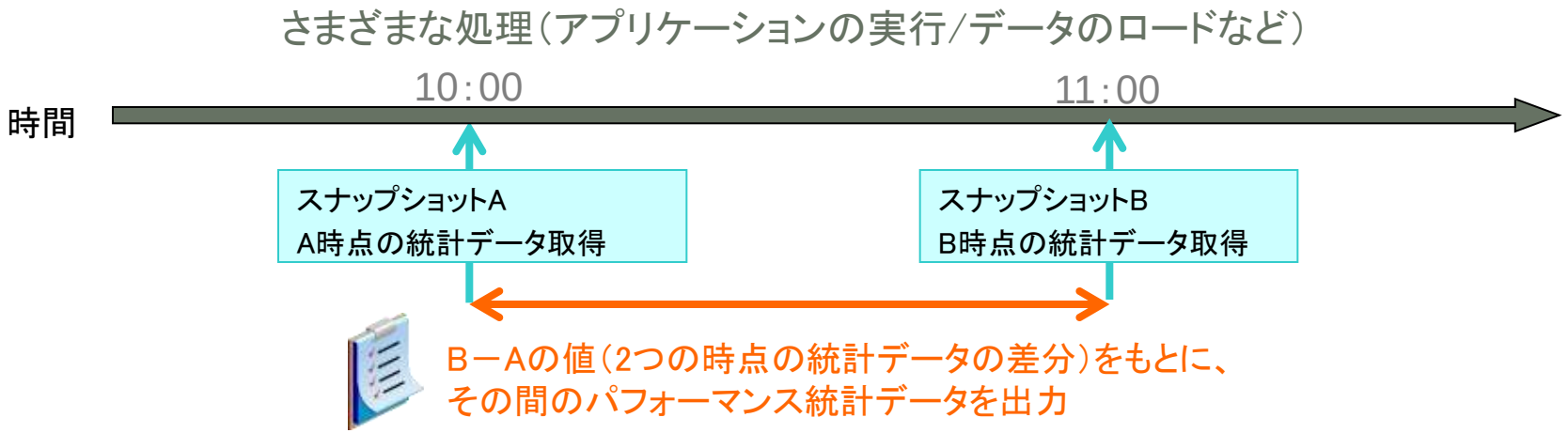
- データベース内の統計情報を収集
  - メモリ上の情報をSELECT文で検索
  - Statspackによる必要な情報の一括収集
  - Oracle Enterprise Managerの自動診断機能の活用



ボトルネック箇所の特定方法
Statspackによる統計情報の取得

- Statspack（STATISTICS PACK）
  - パフォーマンス・チューニングに役立つ情報を収集し、レポート形式で表示するツール
  - ある期間でOracleが行なった処理の統計情報を収集
    - メモリのキャッシュヒット率
    - 待ち時間の内訳
    - トランザクション統計
    - 処理に時間のかかったSQL文

データベース内部で、
どんな処理が行われ
どんな待機が生じているか検出



ボトルネック箇所の特定方法

Statspackの結果レポート例

report1.lst – ワードパッド

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ヘルプ(H)

```
STATSPACK report for

Database    DB Id    Instance     Inst Num  Startup Time   Release     RAC
~~~~~~~~ ---------- ------------ -------- --------------- ----------- ---
          1193534749 orcl                1 23-10月-08 14:5 11.1.0.6.0  NO
                                           9

Host Name            Platform                 CPUs Cores Sockets   Memory (G)
~~~~~ ---------------- ------------------------ ----- ----- ------- ------------
     TSG             Microsoft Windows IA (     2     2       1          2.0

Snapshot       Snap Id     Snap Time      Sessions Curs/Sess Comment
~~~~~~~~    ---------- ------------------ -------- --------- -------------------
Begin Snap:          1 23-10月-08 15:20:50       35       4.1
  End Snap:         11 23-10月-08 17:25:32       34       4.3
   Elapsed:        124.70 (mins)
   DB time:         42.72 (mins)     DB CPU:        7.51 (mins)

Cache Sizes           Begin        End
~~~~~~~~~~~       ---------- ----------
    Buffer Cache:        80M              Std Block Size:         8K
     Shared Pool:       120M                  Log Buffer:     6,167K

Load Profile              Per Second    Per Transaction    Per Exec    Per Call
~~~~~~~~~~~~         ---------------  ---------------- ----------- -----------
      DB time(s):                0.3               1.2        0.03        0.00
       DB CPU(s):                0.1               0.2        0.00        0.00
       Redo size:            2,373.8           8,506.0
   Logical reads:              477.4           1,710.7
   Block changes:               14.6              52.4
  Physical reads:              391.1           1,401.3
 Physical writes:                1.0               3.4
      User calls:              477.8           1,712.1
          Parses:                5.3              19.1
     Hard parses:                0.3               1.1
W/A MB processed:                0.1               0.3
          Logons:                0.1               0.2
        Executes:               13.0              46.6
       Rollbacks:                0.0               0.0
    Transactions:                0.3
```

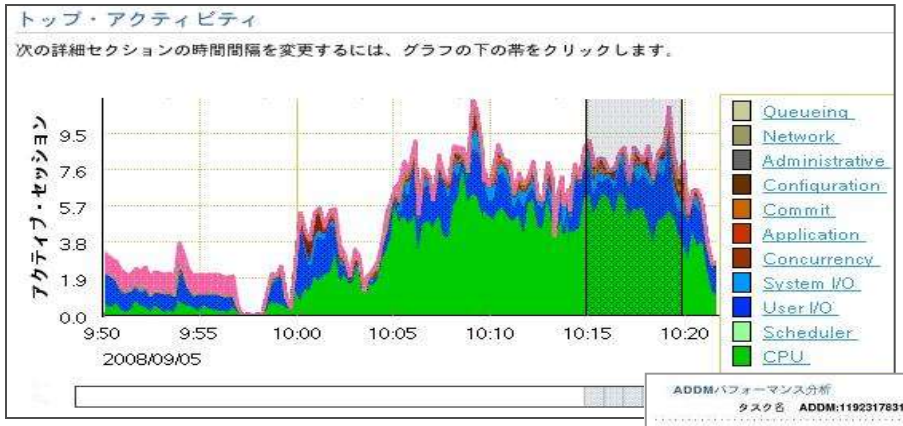
report1.lst – ワードパッド

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ヘルプ(H)

```
Instance Efficiency Indicators
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

            Buffer Nowait %:  100.00       Redo NoWait %:  100.00
            Buffer  Hit   %:   98.54  Optimal W/A Exec %:  100.00
            Library Hit   %:   94.84        Soft Parse %:   94.32
        Execute to Parse %:   89.05         Latch Hit %:  100.00
Parse CPU to Parse Elapsd %:   90.61     % Non-Parse CPU:   92.02

 Shared Pool Statistics        Begin    End
                               ------   ------
             Memory Usage %:   92.85   92.69
    % SQL with executions>1:   63.94   85.65
  % Memory for SQL w/exec>1:   69.71   88.90

Top 5 Timed Events                                                    Avg %Total
~~~~~~~~~~~~~~~~~~                                                    wait   Call
Event                                            Waits    Time (s)    (ms)   Time
------------------------------------------ ------------ ----------- ------ ------
db file sequential read                          59,310       1,109     19   29.6
control file sequential read                      8,087         637     79   17.0
buffer busy waits                                 4,499         524    117   14.0
CPU time                                                        343           9.1
control file parallel write                       2,497         340    136   9.1
-------------------------------------------------------------

Host CPU  (CPUs: 2  Cores: 2  Sockets: 1)
~~~~~~~~                Load Average
                        Begin     End      User  System    Idle     WIO     WCPU
                       ------- -------  ------- ------- ------- ------- --------
                                        41.93   27.66   30.41

Instance CPU
~~~~~~~~~~~~

              % of total CPU for Instance:    3.07
              % of busy  CPU for Instance:    4.41
  %DB time waiting for CPU - Resource Mgr:

Memory Statistics                       Begin          End
~~~~~~~~~~~~~~~~~                  ------------ ------------
                    Host Mem (MB):      2,038.4      2,038.4
                     SGA use (MB):        231.5        231.5
                     PGA use (MB):         63.8         66.4
```

ボトルネック箇所の特定方法
Enterprise Managerによる自動情報取得

- Enterprise Manager
  - ブラウザからアクセスするデータベース管理ツール
  - GUIの画面から負荷状況をグラフィカルに表示
  - ボトルネック項目をリスト

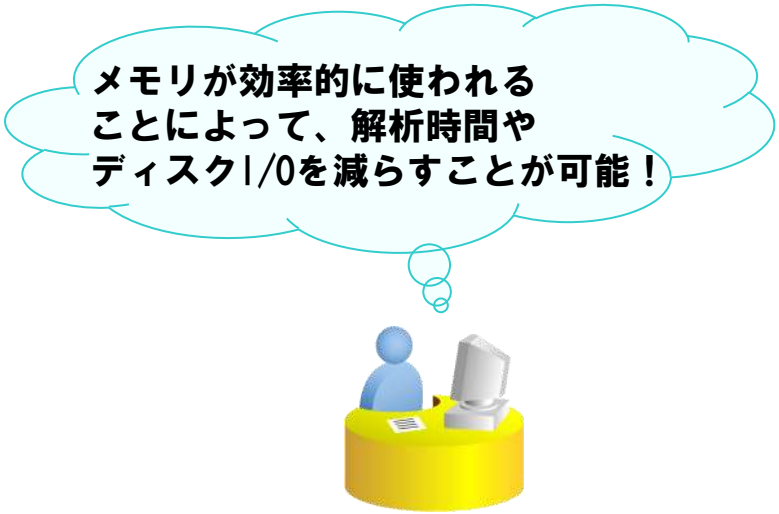


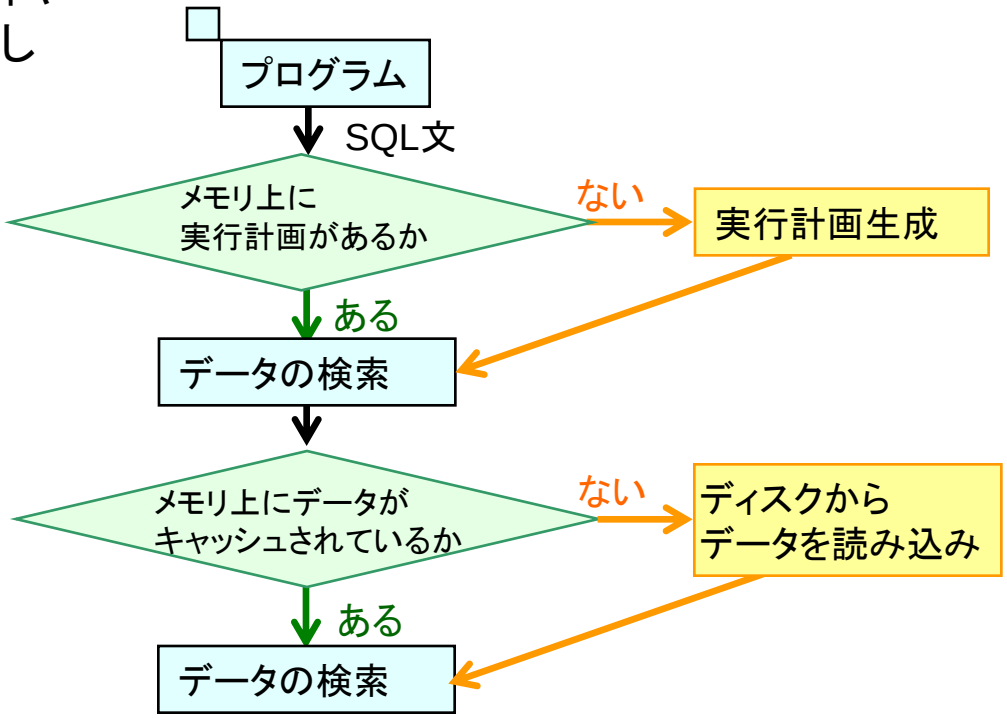
Agenda

- パフォーマンス・チューニングとは
- ボトルネック箇所の特定
- 代表的なチューニング項目
  - メモリ割り当てのチューニング
  - ディスクI/Oのチューニング
  - SQL文のチューニング

メモリ割り当てのチューニング

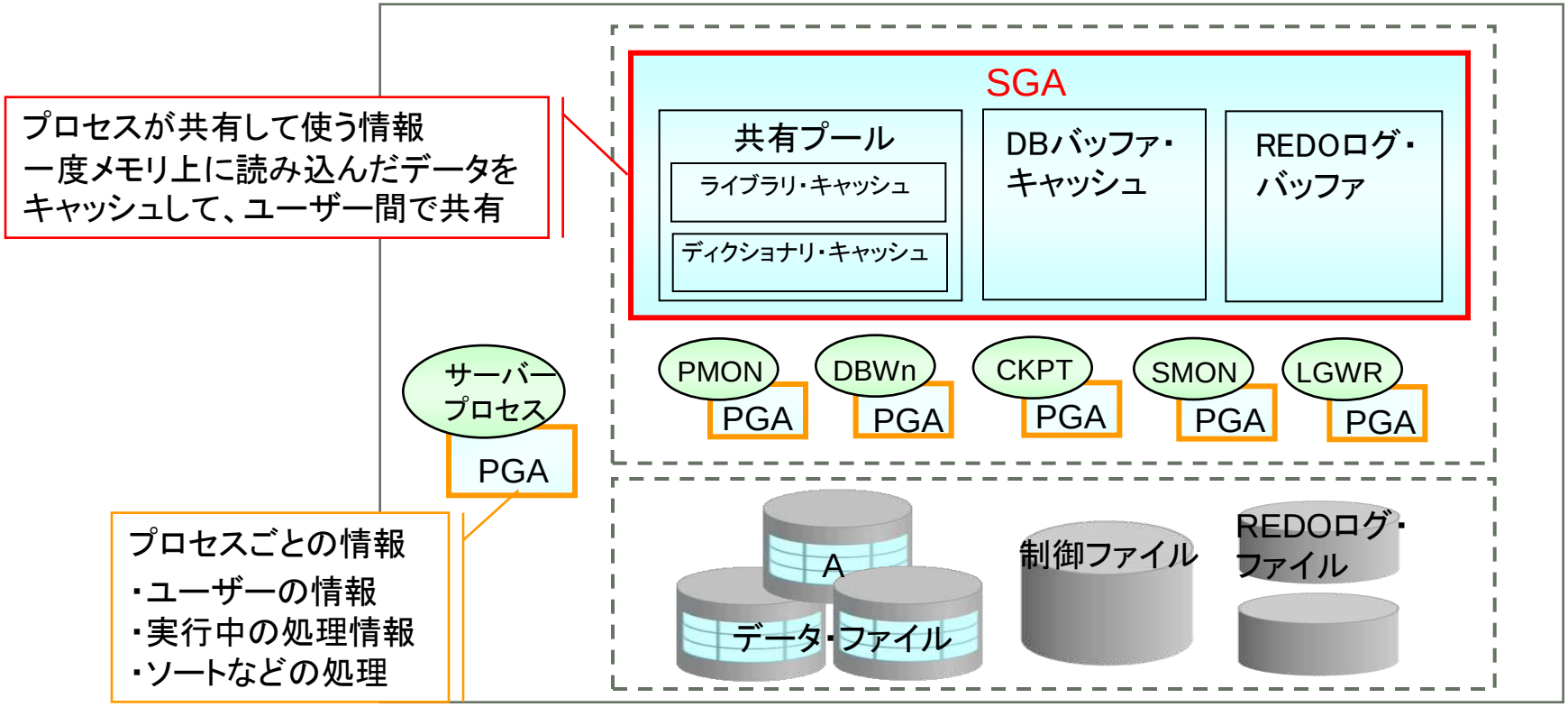
- メモリ・チューニングの重要性
  - Oracle Databaseでは、データの検索、更新などの処理をメモリ上で実行
  - メモリに必要な情報をキャッシュし、複数の処理で共有
  - メモリが不足すると、SQLの解析やディスクからの読み込みが多発しボトルネックになる可能性





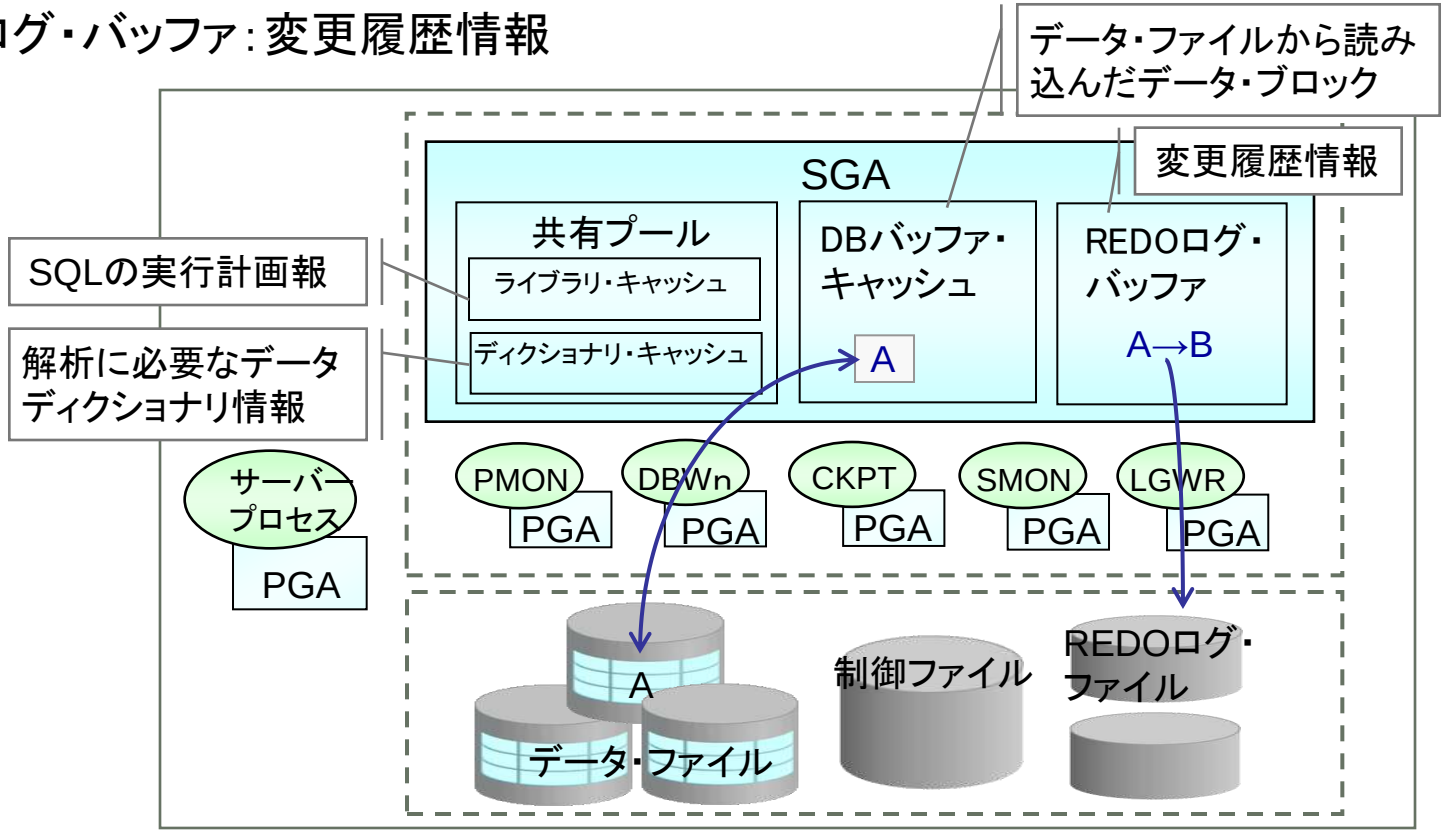
Oracleが使用するメモリの種類

- Oracleが使用する2種類のメモリ
  - SAG：プロセスが共有して使うメモリ（ユーザー間で共有）
  - PGA：プロセスごとに固有に持つメモリ



Oracleが使用するメモリの種類
SGA（System Global Area）

- キャッシュする情報によって、いくつかのコンポーネントに分類
  - 共有プール：SQLの解析情報
  - データベース・バッファ・キャッシュ：ユーザーが使うデータ・ブロック
  - REDOログ・バッファ：変更履歴情報



メモリ不足によって生じる問題

- **SGA（共有プール、DBバッファ・キャッシュ）のメモリ不足**
  - **共有プールの不足**
    - ✓ メモリが不足すると、一度読み込んだデータ・ディクショナリ情報や一度立てた実行計画がフラッシュ（消去）される
    - ✓ 何度もデータ・ディクショナリの情報をディスクに読みに行く必要
    - ✓ キャッシュした実行計画を使いまわせず、再解析する必要
  - **DBバッファ・キャッシュの不足**
    - ✓ メモリが不足すると、一度キャッシュしたブロックをファイルに書き出し
    - ✓ 次回アクセス時、再度ディスクからブロックを読み込む必要
- **PGAのメモリ不足**
  - ✓ メモリ内でソート処理ができず、ディスク・ソートが発生
  - ✓ ソート、結合などの処理速度が大きく低下

メモリが不足しないように、メモリの使用率を監視し、
不足している場合にはメモリの配分を変更する必要がある

メモリ使用状況の調査

- **メモリ使用状況の調査**
  - メモリ上に必要な情報があった割合＝キャッシュ・ヒット
  - メモリ上に必要な情報がなかった割合＝キャッシュ・ミス
  - キャッシュ・ミス率が高い場合にはメモリの割り当てを増やす

ライブラリ・キャッシュのキャッシュ・ミス率

```
SELECT
(sum(reloads)/sum(pins))*10
FROM v$librarycache;
```

キャッシュミス率が1%を上回る場合、
共有プールの値を大きくする

ディクショナリ・キャッシュのキャッシュ・ミス率

```
SELECT
(sum(getmisses)/sum(gets))* 100
FROM v$rowcache;
```

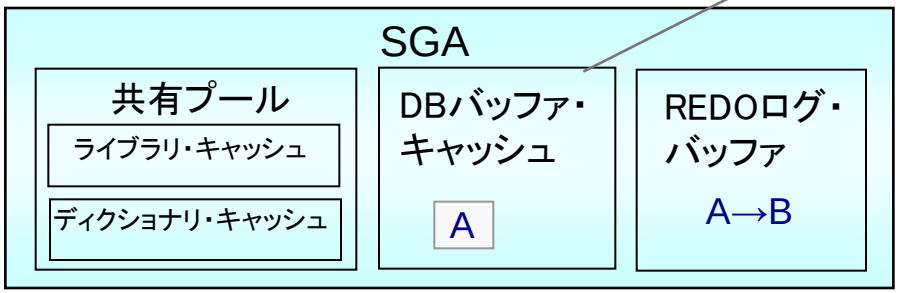
キャッシュミス率が15%を上回る場合、
共有プールの値を大きくする

データベース・バッファ・キャッシュのキャッシュ・ヒット率

$$\frac{\text{physical reads}}{\text{db block gets + consistent gets}}$$

※`SELECT value FROM v$sysstat WHERE name='physical reads';`

キャッシュ・ヒット率が90%を下回る場合、
データベース・バッファ・キャッシュの値を大きくする



メモリの設定方法
Oracle 8iまでの手動管理

- **従来は、管理者がメモリ・コンポーネントごとにサイズを指定**
  - ✓ 各コンポーネントの特性を理解して割り振る必要
  - ✓ PGAはソート、結合など、処理ごとに使用可能メモリを細かく設定
  - ✓ 処理によって必要なメモリ量が変わる場合には設定が難しい
    例）日中はOLTP、夜はバッチ

初期化パラメータ・ファイル

.....

SHARED_POOL_SIZE=500M

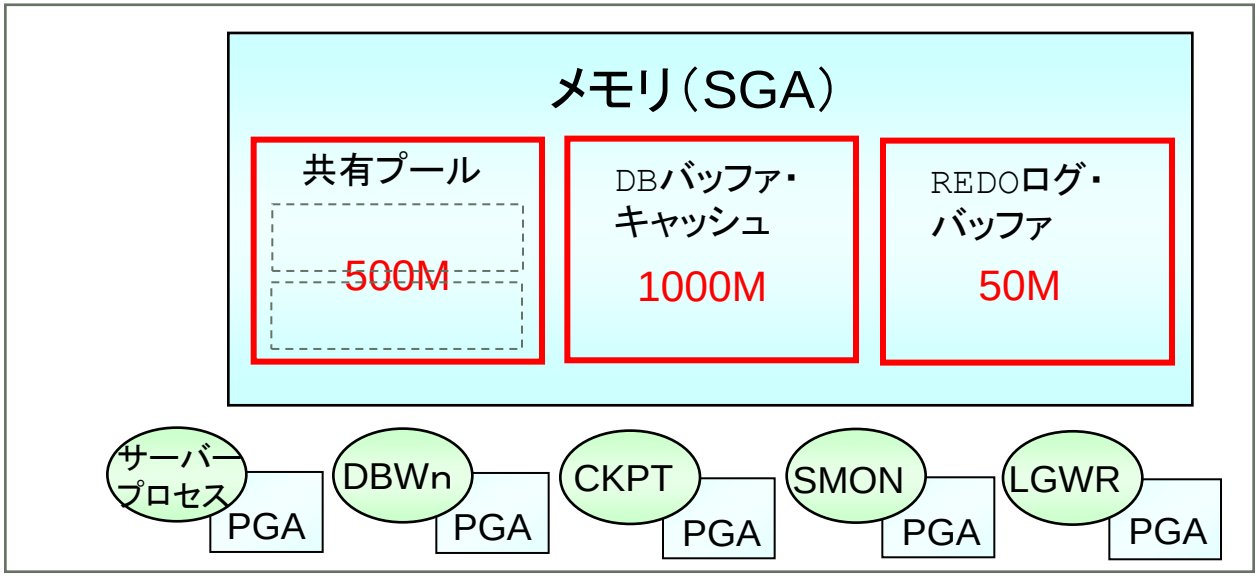
DB_CACHE_SIZE=1000M

LOG_BUFFER=50M

.....

SORT_AREA_SIZE=5M

HASH_AREA_SIZE=10M



メモリの設定方法
Oracle Database9iおよび10gからの自動管理

- **Oracle9i以降　PGAの自動管理**
  - ✓ PGAの総サイズを「PGA_AGGREGATE_TARGET」指定
- **Oracle Database10g以降　SGAの自動管理**
  - ✓ SGAの総サイズを「SGA_TARGET」で指定
  - ✓ 指定されたサイズの中で、必要に応じて各コンポーネントのメモリを割り振り
  - ✓ 負荷に応じて、運用中も自動的にサイズが調整され、最適化される

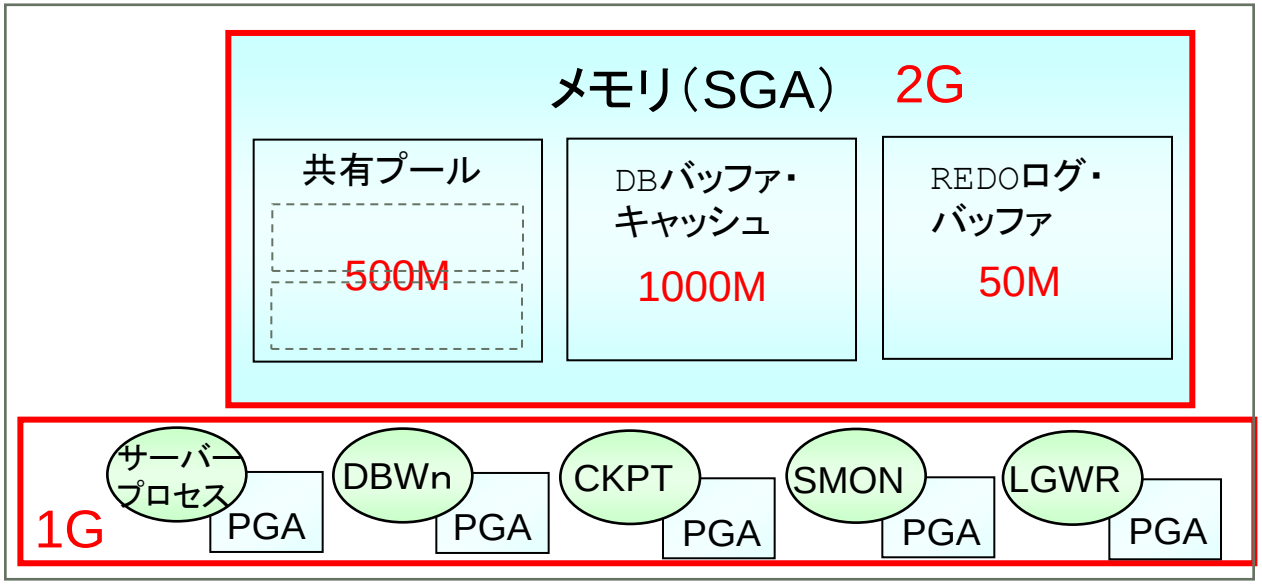
初期化パラメータ・ファイル

.....

PGA_AGGREGATE_TARGET =1G

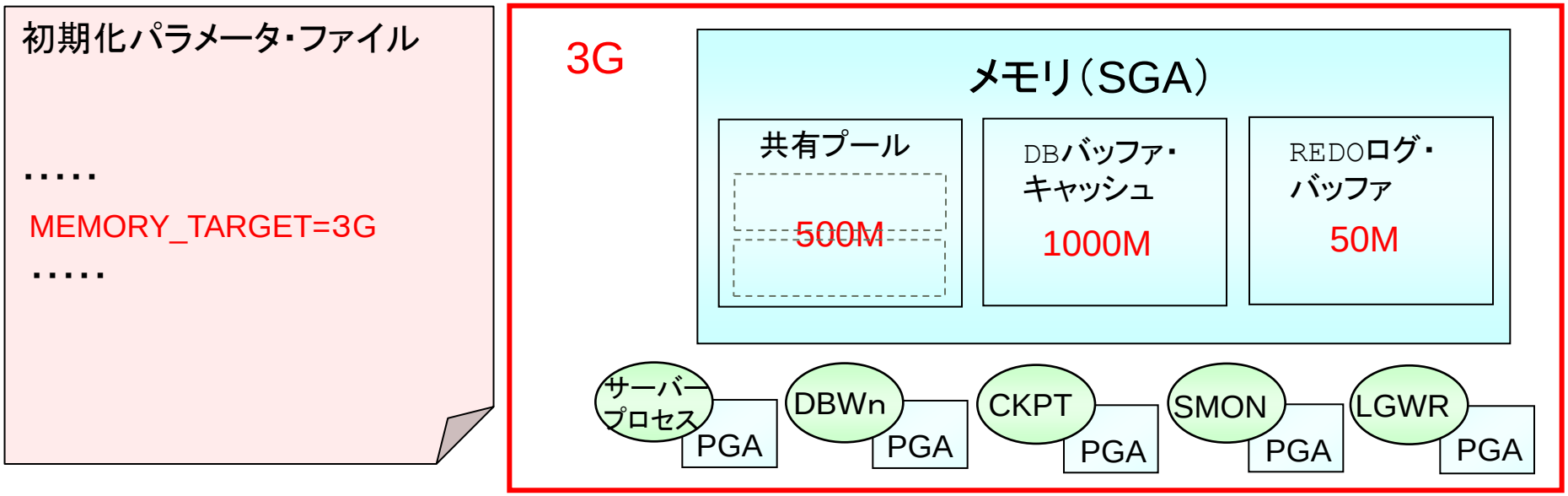
SGA_TARGET=2G

.....



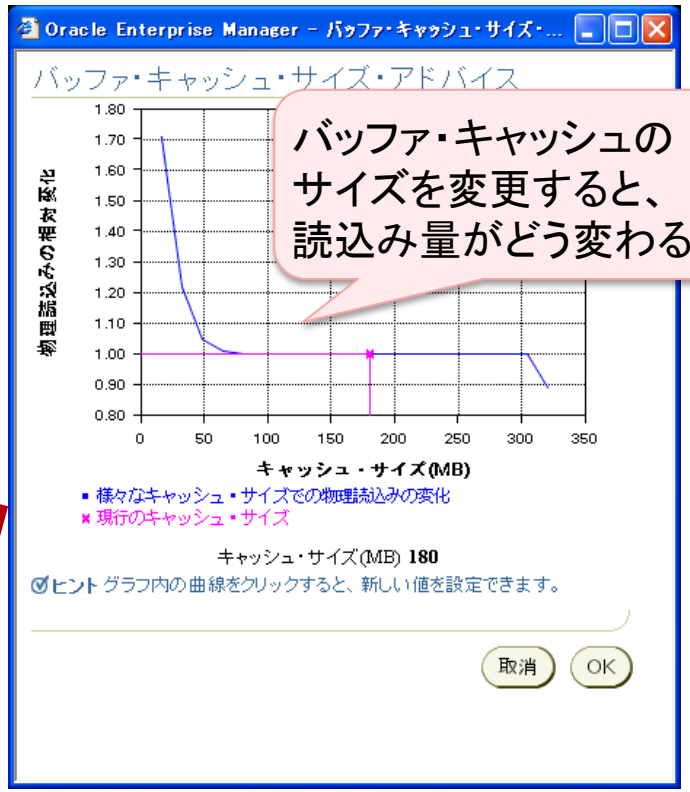
メモリの設定方法
Oracle Database 11gからの自動管理

- **Oracle11g以降　メモリの自動管理**
  - ✓ Oracleが使用するメモリの総サイズを「MEMORY_TARGET」で指定
  - ✓ 割り当てられたサイズの中で、必要に応じてPGAや各コンポーネントのメモリを自動的に割り振り
  - ✓ 負荷に応じて、運用中も自動的にサイズが調整され、最適化される



メモリの設定方法
メモリ・アドバイザ

- メモリ・アドバイザ
  - Oracle Enterprise Managerの画面から、メモリの割り当て状況や最適値を確認することが可能



バッファ・キャッシュのサイズを変更すると、読込み量がどう変わるか

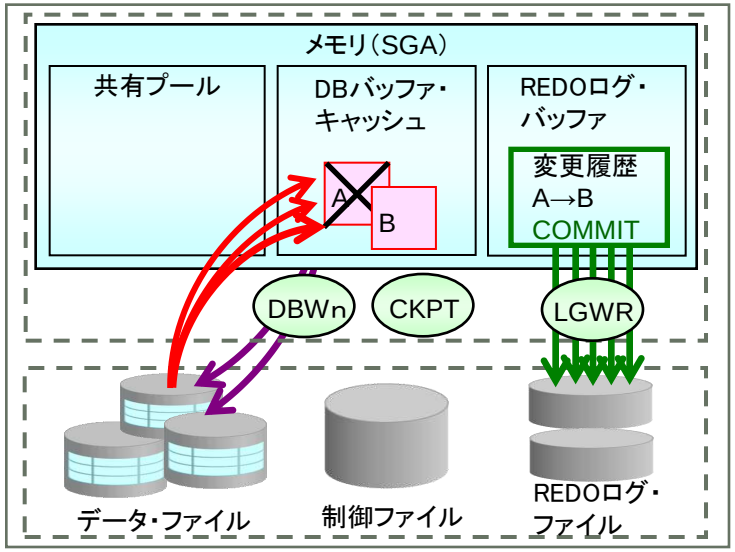
Agenda

- パフォーマンス・チューニングとは
- ボトルネック箇所の特定
- 代表的なチューニング項目
  - メモリ割り当てのチューニング
  - ディスクI/Oのチューニング
  - SQL文のチューニング

ディスクI/Oのチューニング

- ディスクI/Oのチューニング・ポイント
  - ディスクI/Oの速度は、ディスクの性能に依存
  - Oracle Databaseで可能なチューニングは、読み込みを効率化すること

ディスクI/Oのチューニング・ポイント
- データ読み込みの効率化
  - 索引
  - パーティショニング
- 効率的な領域の使用
  - 断片化の解消



データ読み込みの効率化

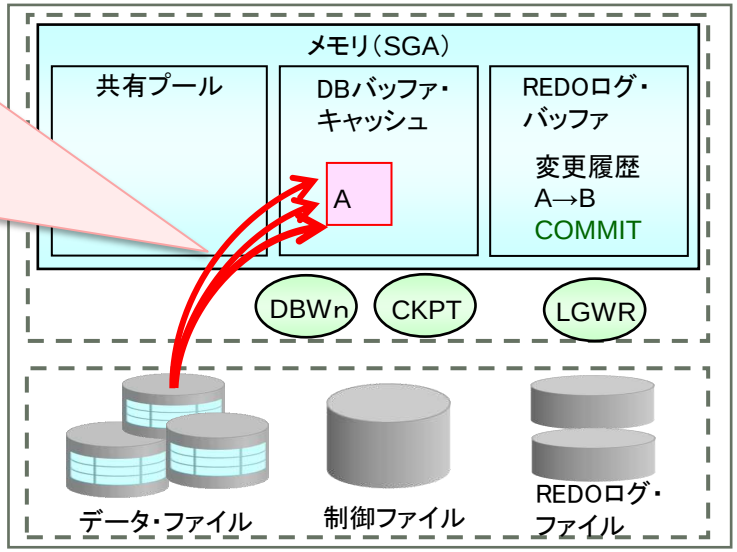
効率的なデータ読み込みの必要性

- 大量のデータをメモリ上に読み込む処理による悪影響
  - メモリの圧迫
  - ディスクI/Oの増加
  - 大量のデータを処理することによるCPU使用

効率的なデータ読み込みが必要

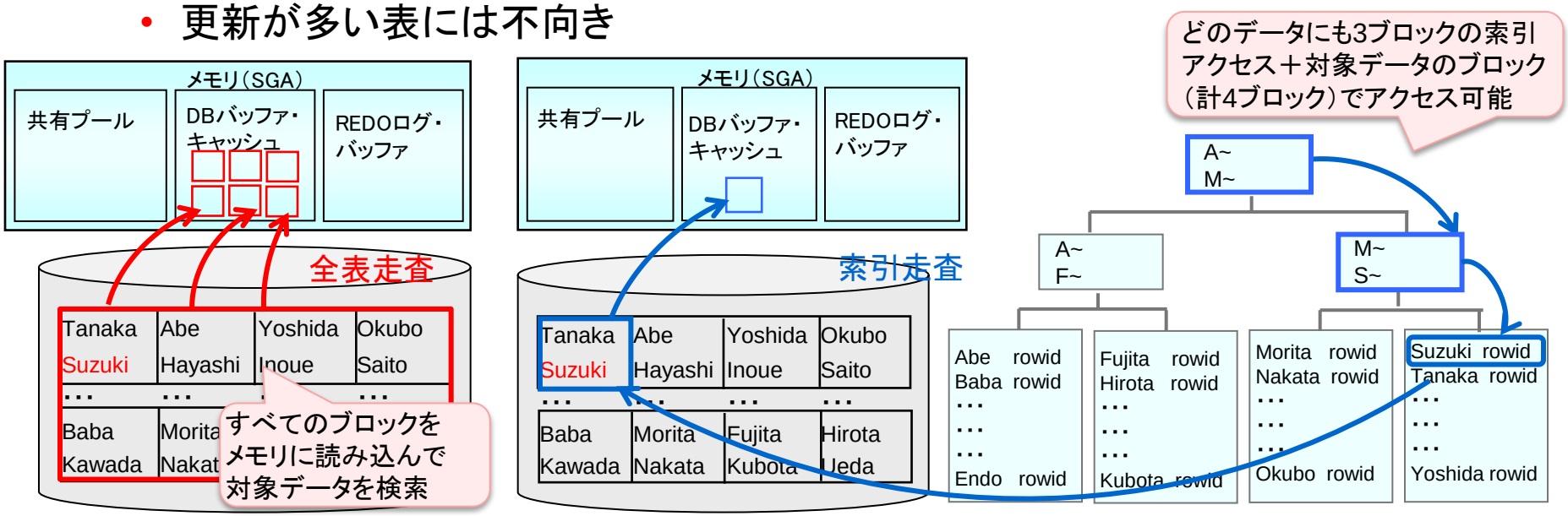
データの読み込み方法

- **全表走査（フル・スキャン）**
  - 表の全てのブロックを読み込んで検索対象のデータを探す方法
- **索引操作（インデックス・スキャン）**
  - 索引で検索対象の行データが格納されたブロックを特定し、該当ブロックのみを読み込んで検索する方法



データ読み込みの方法
全表走査と索引走査

- 全表走査
  - 表のすべてのブロックをメモリに読み込み、該当データを検索
  - 表が小さい場合、検索対象の割合が多い場合に効果的
- 索引走査
  - 索引を使って検索対象のデータが格納されたブロックを特定して読み込む
  - 表が大きい場合、検索対象の割合が少ない場合に効果的
  - 更新が多い表には不向き



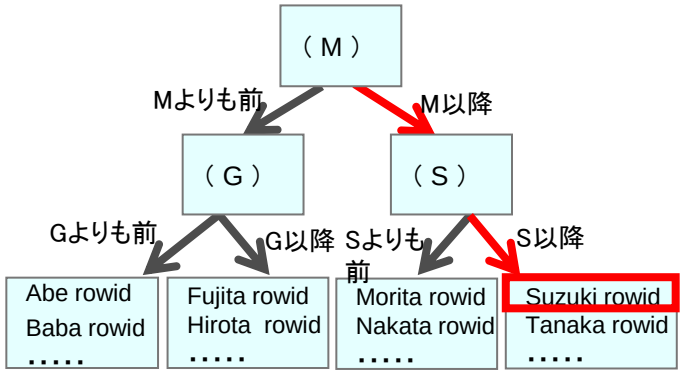
データ読み込みの効率化
索引を使ったチューニング

索引を使ったチューニングの詳細は
以下の資料で説明しています

実践!! パフォーマンス・チューニング
- 索引チューニング編 -

- 索引に関するチューニング
  - 最適な索引タイプの選択
    - B-tree索引
    - ビットマップ索引
    - 逆キー索引
    - 複合索引
    - 索引構成表
    - ファンクション索引
  - データの偏りの解消
    - ヒストグラムの作成
  - 索引のメンテナンス
    - 断片化の解消
    - 索引の再構築

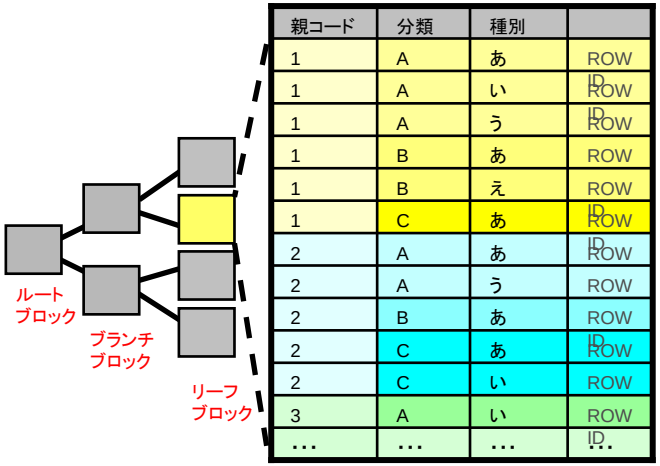
Bツリー索引



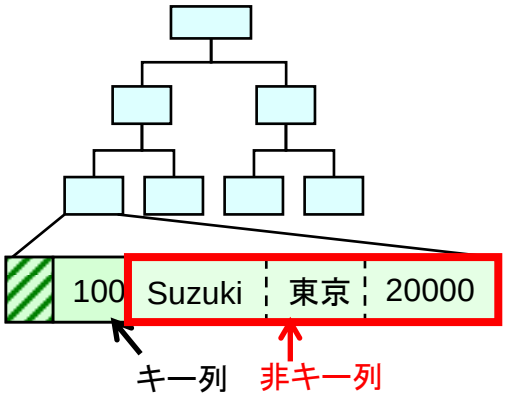
ビットマップ索引

ROWID	男	女
ROWID1	1	0
ROWID2	0	1
ROWID3	1	0
ROWID4	0	1
ROWID5	1	0

複合索引

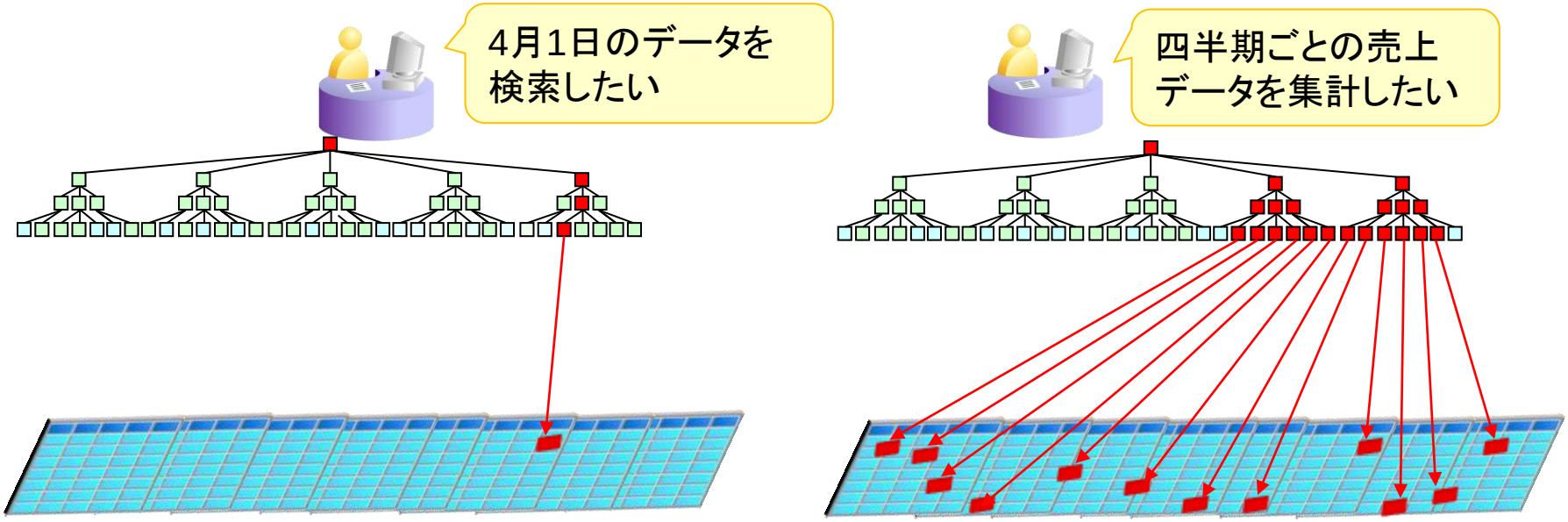


複合構成表



データ読み込みの効率化
索引の問題点

- **索引検索でもパフォーマンスが上がらないケース**
  - 表サイズが大きく、検索対象データも多い場合、索引と表に大量のアクセスが発生するため、パフォーマンスがあがらない



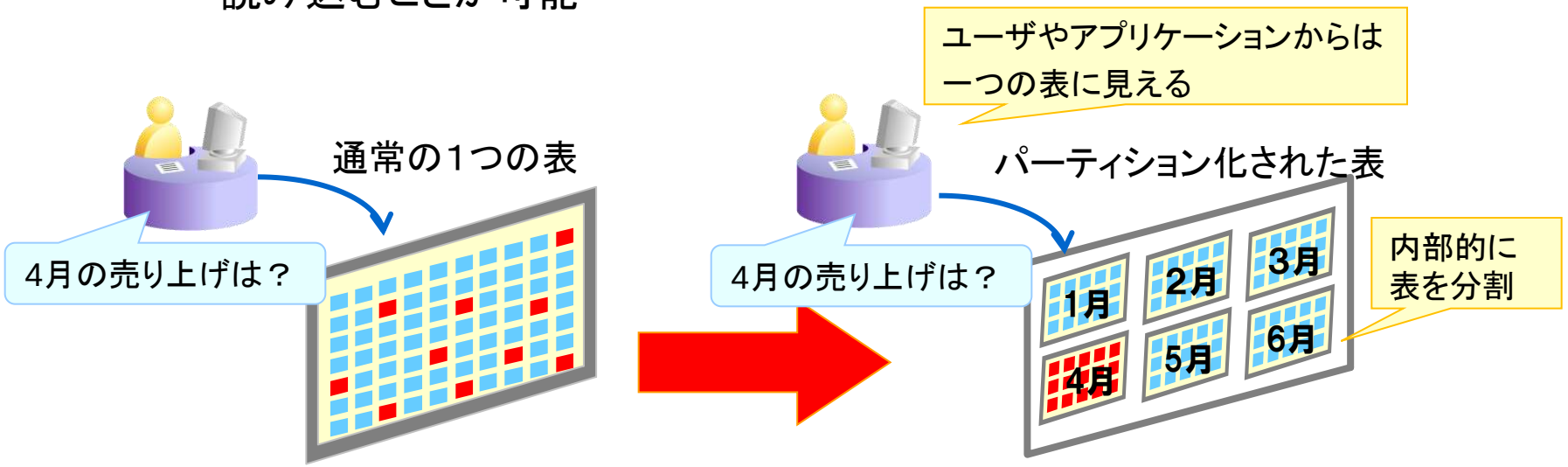
大量の索引読み込みが行われる検索では、DISK I/Oがボトルネックに・・・

索引は万能ではない

データ読み込みの効率化
パーテショニングを使った効率化

パーティショニングの詳細は
以下の資料で説明しています
実践!! 大規模データベース管理
ーパーティション基本編ー

- パーティショニング機能とは
  - 大きな表や索引をデータベース内部で複数の領域に分割して管理するしくみ
  - 検索範囲をパーティション単位に絞ることができるため、効率的にデータを読み込むことが可能



検索したいデータが一部でも、
表全体または索引を利用してアクセス

各種メンテナンス作業も
表全体に影響が及んでしまう

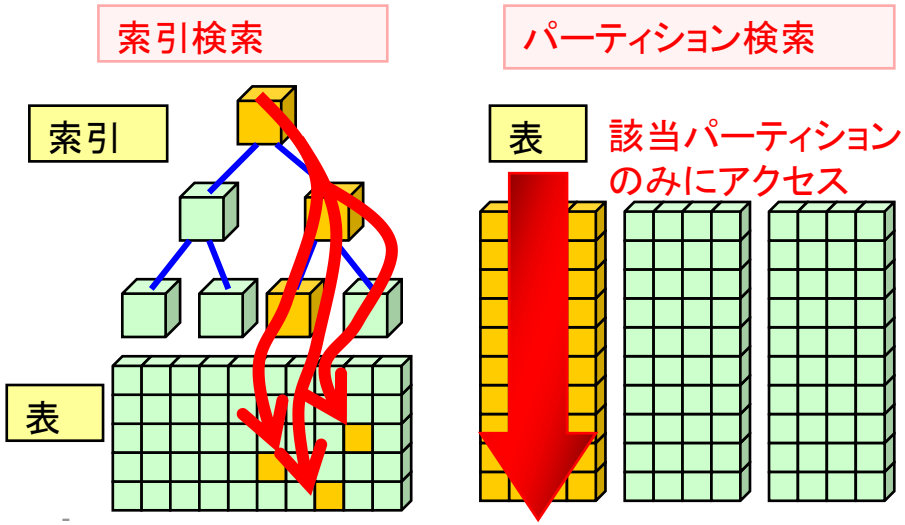
検索したいデータが格納されている
パーティションにのみアクセス

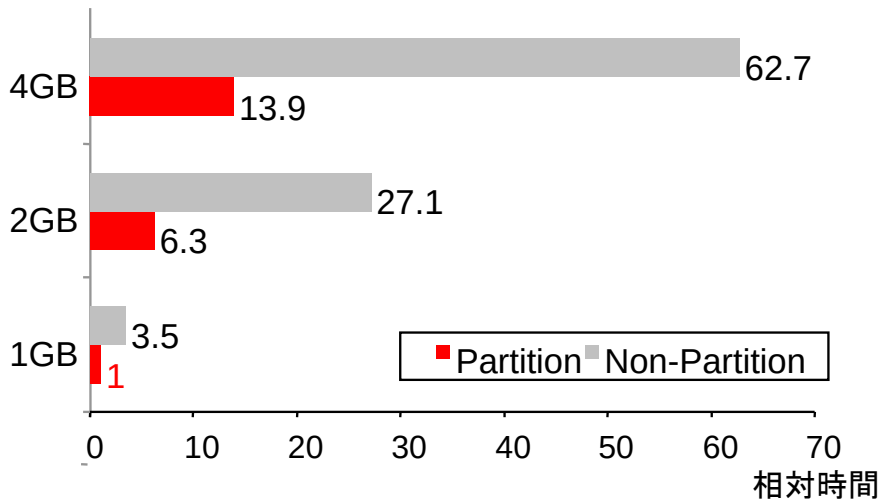
各種メンテナンス作業も
パーティション単位で実施可能

データ読み込みの効率化
索引検索とパーティション検索の比較

- 有効なケース
  - 索引が有効なケース
    - 検索対象が少ない場合
  - パーティショニングが有効なケース
    - 表が大きい場合
    - 検索対象が多い場合

- 検索時間の比較
  - 各サイズの表に対して
    150,000 レコード検索
  - 表サイズが大きくなればなるほど
    パーティションが有効





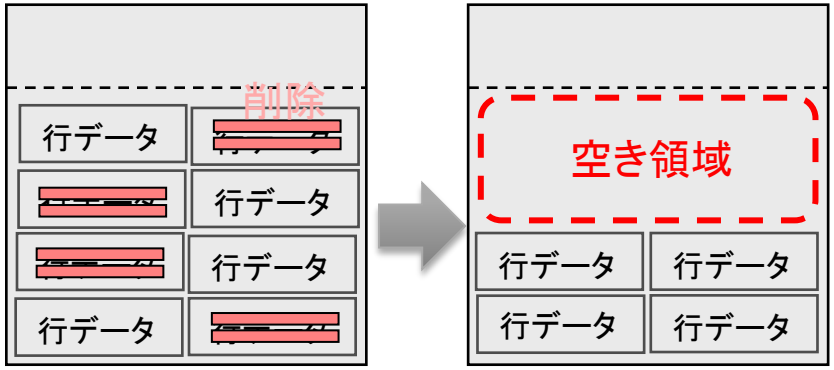
※（実際の処理時間に任意の数を掛けています）

効率的な領域の使用
断片化により起こる問題

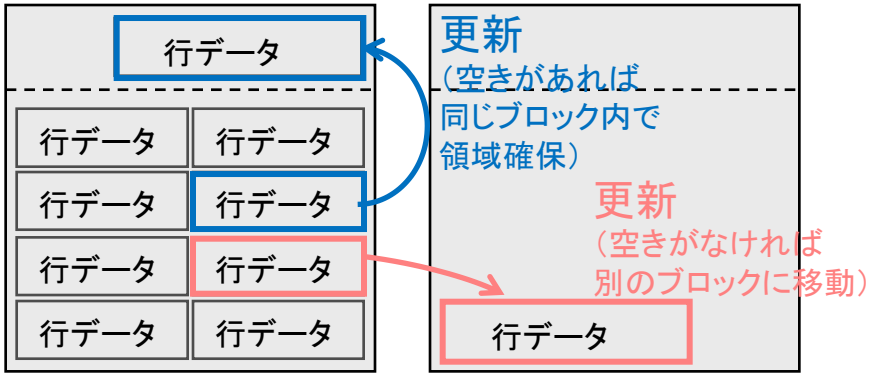
- 運用中のデータ追加、削除、更新により、断片化が発生
  - 行データの削除によるブロック内の空き領域
  - 行データの更新に伴うデータの増加による行移行
  - データの追加、削除による、索引の断片化

使用領域の増加、読み込むブロックの増加によるパフォーマンス低下

断片化
行データの削除により、ブロック内に空きができる
ブロックごと読み込むので、効率が悪い

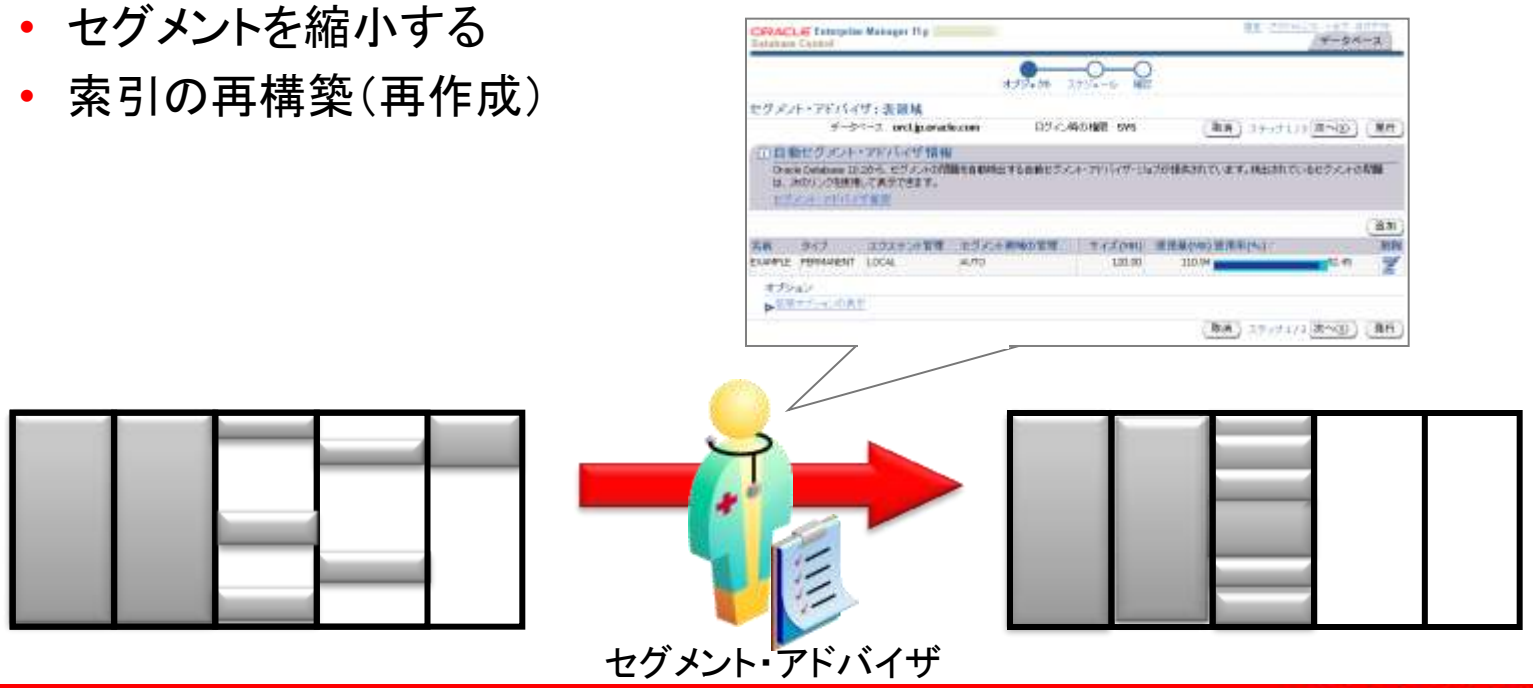


行移行
更新時に行データのサイズが増え、他のブロックに
データが移ること
検索時に移行前のブロックも読むので効率が悪い



効率的な領域の使用
断片化の診断と解決

- **従来の断片化の診断と解消**
  - ANALYZEコマンドで分析
  - Export/Import等でデータを入れなおす
- **Enterprise Managerによる断片化の診断と解消**
  - セグメント・アドバイザ機能により、断片化している領域を特定
  - セグメントを縮小する
  - 索引の再構築（再作成）

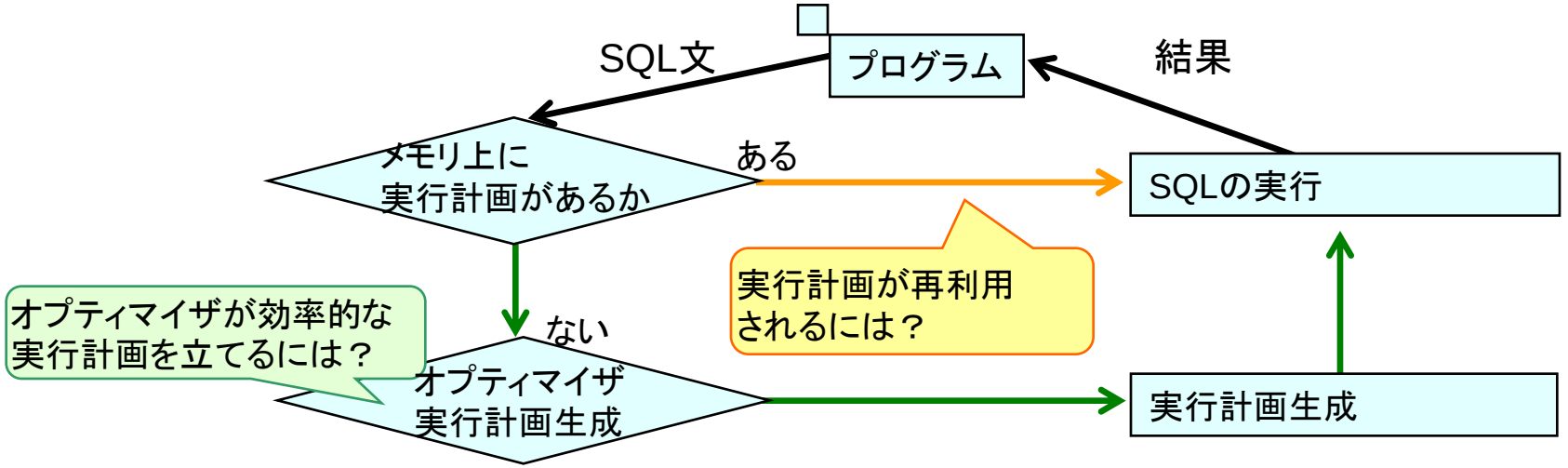


Agenda

- パフォーマンス・チューニングとは
- ボトルネック箇所の特定
- 代表的なチューニング項目
  - メモリ割り当てのチューニング
  - ディスクI/Oのチューニング
  - SQL文のチューニング

SQL文の処理ステップ

- **SQL文の処理ステップ**
  - 発行されたSQL文の解析（パース）
    - 同一SQL文がメモリ上に存在すれば、その結果を使ってすぐに実行
    - 同一SQL文がメモリ上に存在しなければ、解析処理を行う
  - 解析処理：オプティマイザが最適な実行計画を検討
    - 索引を利用するか、全表走査するか
    - 複数表を結合する場合にどの順番で、どの結合方法を使うか



【参考】SQL文の実行計画

実行計画例

オプティマイザがSQL文ごとに、最適な実行計画を立てる

```
SELECT last_name,department_name
FROM   employees JOIN departments USING  (department_id);

-------------------------------------------------------------------------------
| Id  | Operation                    | Name        | Rows  | Bytes | Cost (%CPU)|···
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |             |   106 |  2862 |     6  (17)|
|   1 |  MERGE JOIN                  |             |   106 |  2862 |     6  (17)|
|   2 |   TABLE ACCESS BY INDEX ROWID| DEPARTMENTS |    27 |   432 |     2   (0)|
|   3 |    INDEX FULL SCAN           | DEPT_ID_PK  |    27 |       |     1   (0)|
|*  4 |   SORT JOIN                  |             |   107 |  1177 |     4  (25)|
|   5 |    TABLE ACCESS FULL         | EMPLOYEES   |   107 |  1177 |     3   (0)|
```

```
SELECT last_name,department_name
FROM   employees JOIN departments USING  (department_id)
WHERE  department_id=10;

-------------------------------------------------------------------------------
| Id  | Operation                    | Name              | Rows  | Bytes | Cost (%CPU)|···
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                   |     1 |    27 |     2   (0)|
|   1 |  NESTED LOOPS                |                   |     1 |    27 |     2   (0)|
|   2 |   TABLE ACCESS BY INDEX ROWID| DEPARTMENTS       |     1 |    16 |     1   (0)|
|*  3 |    INDEX UNIQUE SCAN         | DEPT_ID_PK        |     1 |       |     0   (0)|
|   4 |   TABLE ACCESS BY INDEX ROWID| EMPLOYEES         |     1 |    11 |     1   (0)|
|*  5 |    INDEX RANGE SCAN          | EMP_DEPARTMENT_IX |     1 |       |     0   (0)|
```

【参考】SQL文の実行計画
実行計画の確認方法

- 実行計画を確認する方法
  - SQL*PLUSのAUTOTRACEコマンド
  - Explain plan for <SQL>
  - SQLトレース
  - V$SQL及びV$SQL_PLAN(9i～)
  - Enterprise Manager（10g～）

実行計画の調べ方（SQL*PlusのAUTOTRACE機能）

1. SYSユーザでPLUSTRACEロールを作成し、SQLを実行するユーザに付与する。

```
SQL> @%ORACLE_HOME%¥sqlplus¥admin¥plustrce.sql
SQL> GRANT plustrace TO scott;
```

2. SQLを実行するユーザで実行計画を保存するための表(PLAN_TABLE)を作成する。

```
SQL> connect scott/tiger
SQL> @%ORACLE_HOME%¥rdbms¥admin¥utlxplan.sql
```

3. AUTOTRACE 機能を ON にし、SQL文を実行する。

```
SQL> SET AUTOTRACE ON
SQL> SELECT ...
```

SQL文の実行計画とパフォーマンス
効率的な実行計画を立てるためのポイント

- コーディング・ルールの統一

```
SELECT name FROM emp;
```

```
SELECT name FROM EMP;
```
← 大文字/小文字の違い

```
SELECT name  FROM  emp;
```
← スペース/改行の違い

- バインド変数の利用

```
SELECT name FROM emp WHERE id = 1023
```

```
SELECT name FROM emp WHERE id = 3074
```
← 値が異なる

```
variable b1 number
begin
:b1 := 300;
end;
/
SELECT name FROM emp where id = :b1;
```

バインド変数：SQLの条件値を変数化したもの

SQL文の実行計画とパフォーマンス
実行計画の生成のしくみ

- **問合せの結果を生成する最も効率的な方法（物理的なアクセス手順）を決定し、実行計画を作成する機能＝オプティマイザ**
  - 索引を利用するか
  - 全表スキャンを利用するか
  - 複数の表を結合するときに、結合順序/結合方法はどうするか　など
- **オプティマイザの種類**
  - ルールベース・オプティマイザ（RBO）
  - コストベース・オプティマイザ（CBO）

ルールベース・オプティマイザ
- 従来のオプティマイザ（Oracle10g 以降サポートされない）
- 事前定義されたアクセスパスの中から、優先順位に従って実行計画を選択

考慮点
- Oracle7.3以降の新機能には対応していない
- SQL文の構文によって実行計画が決まるため、柔軟性に乏しい
- データの中身、検索条件により、より高速なアクセスパスが存在する可能性がある

SQL文の実行計画とパフォーマンス
コストベース・オプティマイザの特徴

- **コストベース・オプティマイザ**
  - 統計情報に基づいてアクセスコストを見積もり、最もコストの低い実行計画を作成する
    - コスト：DISK I/O、CPU使用量、メモリー使用量から算出される『使用リソース』

オプティマイザ統計情報
- 表統計（行数、ブロック長、平均行長）
- 列統計（列内のデータ種類数、列内のNULL数）
- 索引統計（リーフブロック数、ツリーの高さ）
- システム統計（I/O、CPUパフォーマンス）

- **コストベース・オプティマイザを活用するための考慮点**
  - オプティマイザ統計情報を取得する必要
    - 9iまで：手動取得
    - 10g以降：自動取得（ただし状況によっては手動で取得したほうがよい場合も）
  - データ量の変化、データの偏りなどにより、最適な実行計画が立てられない可能性もある
    - ヒント句などを使用し、特定の実行計画を指定

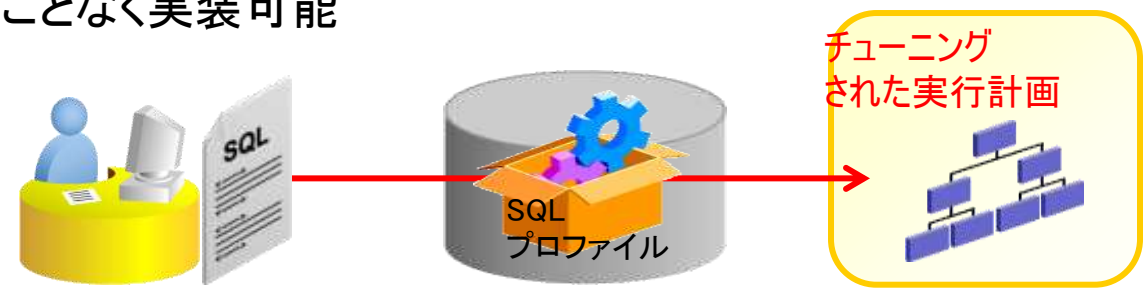
SQL文の実行計画とパフォーマンス
実行計画の改善

- **オプティマイザ・ヒント：特定のアクセス・パスを使用させるための指定**
  - 適切な索引の使用を指定
  - 適切な表結合方法や結合順序を指定

ヒントの使用例（ /*+ と */ の間でヒントを指定し、SQLに直接埋め込む ）
＜例＞sales表のcustomer_id列のcust_id_indxという索引を使用

```
SELECT /*+ INDEX(sales cust_id_indx) */ sales_date
FROM sales  WHERE customer_id=100;
```

- **Enterprise Manager「SQLプロファイル」**
  - 最適な実行計画を立てるための追加情報をデータベース内部に収集
  - SQL文を変更することなく実装可能



SQL文のチューニング
SQLチューニング・アドバイザによる自動チューニング

- **SQLチューニング・アドバイザ**
  - Oracle Database10gから実装されたアドバイス機能
  - 高負荷で問題となるSQL文や実行計画を診断し、アドバイスを提示
    - 統計の再取得
    - SQL文の問題点を探し、SQL文の修正方法
    - 必要な索引の作成をアドバイス
    - SQLプロファイルの作成

高負荷の
SQL文

SQL
チューニング
・アドバイザ

チューニング・アドバイザが
負荷を軽減する最適な
対処方法を提示

失効・欠落している
統計の収集

Index の作成

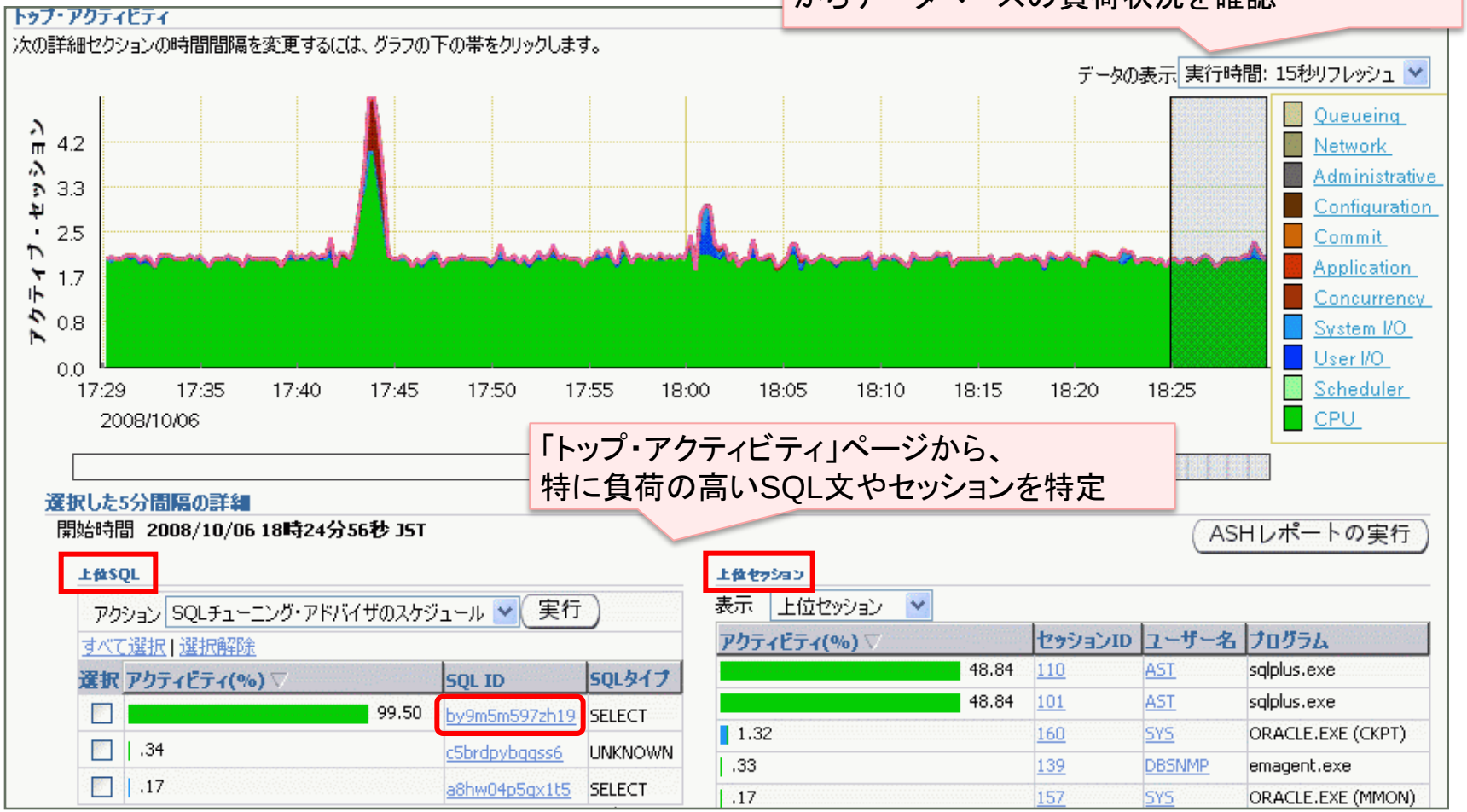
SQL文の
再構成

SQLプロファイル
の作成

SQL文のチューニング

SQLチューニング・アドバイザの実行例（1）

Enterprise Managerの「パフォーマンス・ページ」からデータベースの負荷状況を確認



SQL文のチューニング
SQLチューニング・アドバイザの実行例（2）

SQLの詳細: by9m5m597zh19

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

テキスト

```
select /*+ USE_NL(s c) FULL(s) FULL(c) AST */ c.cust_id, sum(s.quantity_sold)
from sh.sales s, sh.customers c
where s.cust_id = c.cust_id and c.cust_id < 2 group by c.cust_id
```

「上位SQL」から、負荷の高いSQL文を特定
→このSQL文の実行計画を確認

チューニング対象のSQL文を選び、
「SQLチューニング・アドバイザのスケジュール」から実行

詳細

次の詳細を参照するには計画ハッシュ値を選択してください。　計画ハッシュ値 4005616876

統計　アクティビティ　プラン　計画管理　チューニング履歴

データソース カーソル・キャッシュ　取得時間 2008/10/06 18:32:22 (UTC+09:00)　解析スキーマ AST　オプティマイザ・モード ALL_ROWS

追加情報

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 (%)	時間	パーティションの開始	パーティションの停止	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		6			9,398	100						
HASH GROUP BY		5	1	13	9,398	1	0:1:53			SEL$1		"C"."CUST_ID"[NUMBER,22], SUM (...
NESTED LOOPS		4	1	13	9,397	1	0:1:53					"C"."CUST_ID"[NUMBER,22], "S"....
TABLE ACCESS FULL	SH.CUSTOMERS	1	1	5	405	0	0:0:5			SEL$1 / C@SEL$1	"C"."CUST_ID"<2	"C"."CUST_ID"[NUMBER,22]
PARTITION RANGE ALL		3	1	8	8,992	1	0:1:48	1	28			"S"."QUANTITY_SOLD"[NUMBER,22]
TABLE ACCESS FULL	SH.SALES	2	1	8	8,992	1	0:1:48	1	28	SEL$1 / S@SEL$1	("S"."CUST_ID"<2 AND "S"."CUST...	"S"."QUANTITY_SOLD"[NUMBER,22]

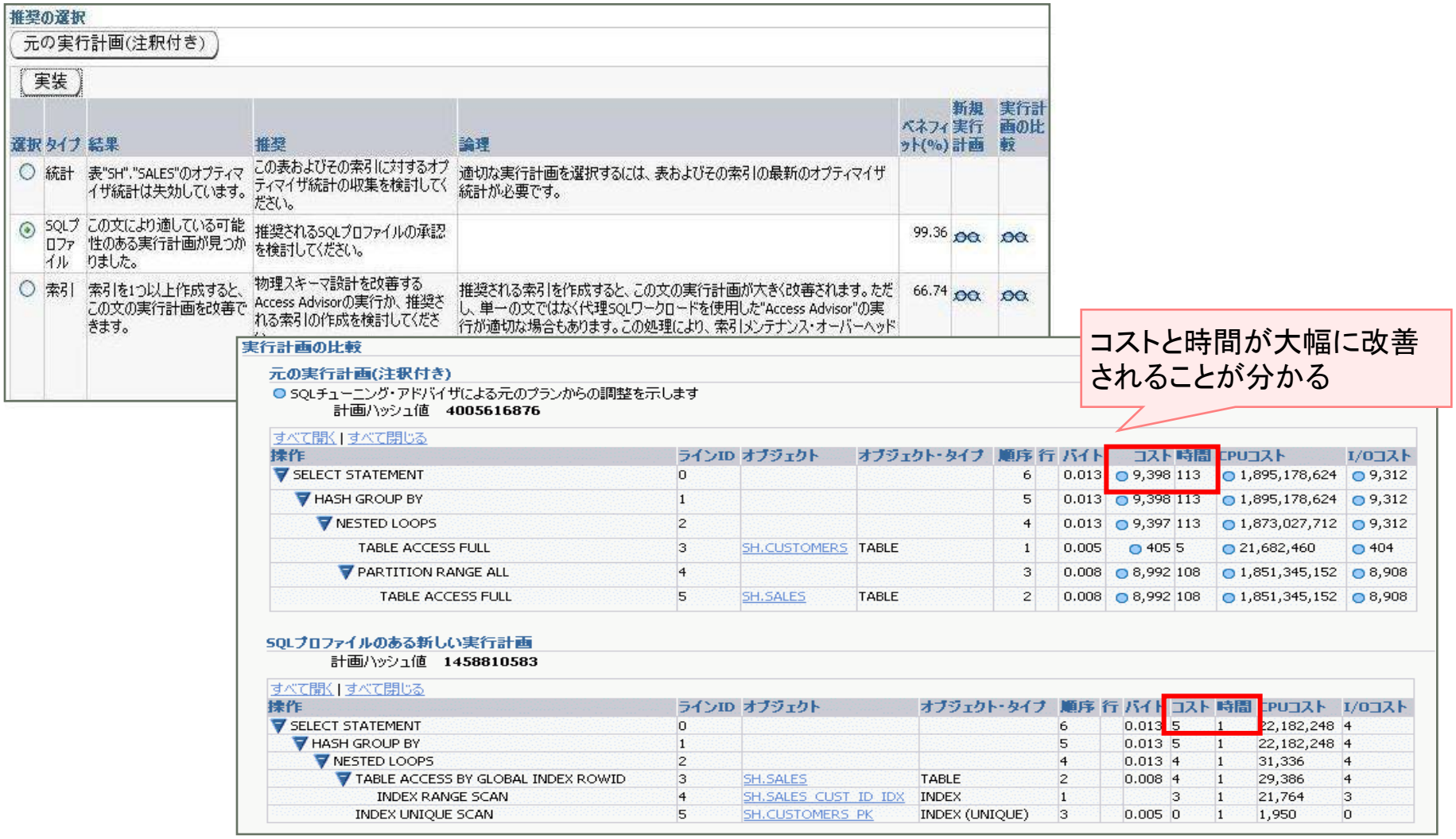
リライトの説明の表示

統計　アクティビティ　プラン　計画管理　チューニング履歴

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文のチューニング

SQLチューニング・アドバイザの実行例（３）



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(%)	新規実行計画	実行計画の比較
○	統計	表"SH"."SALES"のオプティマイザ統計は失効しています。	この表およびその索引に対するオプティマイザ統計の収集を検討してください。	適切な実行計画を選択するには、表およびその索引の最新のオプティマイザ統計が必要です。			
◉	SQLプロファイル	この文により適している可能性のある実行計画が見つかりました。	推奨されるSQLプロファイルの承認を検討してください。		99.36	oo	oo
○	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	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	66.74	oo	oo

実行計画の比較

元の実行計画(注釈付き)

SQLチューニング・アドバイザによる元のプランからの調整を示します

計画ハッシュ値　**4005616876**

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			6		0.013	9,398	113	1,895,178,624	9,312
HASH GROUP BY	1			5		0.013	9,398	113	1,895,178,624	9,312
NESTED LOOPS	2			4		0.013	9,397	113	1,873,027,712	9,312
TABLE ACCESS FULL	3	SH.CUSTOMERS	TABLE	1		0.005	405	5	21,682,460	404
PARTITION RANGE ALL	4			3		0.008	8,992	108	1,851,345,152	8,908
TABLE ACCESS FULL	5	SH.SALES	TABLE	2		0.008	8,992	108	1,851,345,152	8,908

SQLプロファイルのある新しい実行計画

計画ハッシュ値　**1458810583**

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			6		0.013	5	1	22,182,248	4
HASH GROUP BY	1			5		0.013	5	1	22,182,248	4
NESTED LOOPS	2			4		0.013	4	1	31,336	4
TABLE ACCESS BY GLOBAL INDEX ROWID	3	SH.SALES	TABLE	2		0.008	4	1	29,386	4
INDEX RANGE SCAN	4	SH.SALES_CUST_ID_IDX	INDEX	1			3	1	21,764	3
INDEX UNIQUE SCAN	5	SH.CUSTOMERS_PK	INDEX (UNIQUE)	3		0.005	0	1	1,950	0

コストと時間が大幅に改善されることが分かる

まとめ

- パフォーマンス・チューニングとは
- ボトルネック箇所の特定
- 代表的なチューニング項目
  - メモリ割り当てのチューニング
  - ディスクI/Oのチューニング
  - SQL文のチューニング

チューニングとは
ボトルネックを特定し、ボトルネックを解消するための
対策（設定変更、構成変更など）を行うこと

ボトルネックになりやすいポイントは
メモリやディスクI/O、SQL文。
環境に合わせてチューニングポイントを検討

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。

OTN×ダイセミ　でスキルアップ!!

・一般的な技術問題解決方法などを知りたい！
・セミナ資料など技術コンテンツがほしい！

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

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

一般的技術問題解決にはOTN掲示版の
「データベース一般」をご活用ください

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

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

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

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

ORACLE

OTNセミナー　オンデマンド　コンテンツ

ダイセミで実施された技術コンテンツを動画で配信中!!

ダイセミのライブ感はそのままに、お好きな時間で受講頂けます。

OTN　オンデマンド　検索

※掲載のコンテンツ内容は予告なく変更になる可能性があります。
　期間限定での配信コンテンツも含まれております。お早めにダウンロード頂くことをお勧めいたします。

Oracle エンジニアのための技術情報サイト

オラクルエンジニア通信

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

- 技術資料
  - ダイセミの過去資料や製品ホワイトペーパー、スキルアップ資料などを多様な方法で検索できます
  - キーワード検索、レベル別、カテゴリ別、製品・機能別
- コラム
  - オラクル製品に関する技術コラムを毎週お届けします
  - 決してニッチではなく、誰もが明日から使える技術の「あ、そうだったんだ！」をお届けします

オラクルエンジニア通信

先月はこんな資料が人気でした

- ✓ Oracle Database 11gR2 RAC インストレーション・ガイド ASM 版 Microsoft Windows x86-64
- ✓ Oracle Database 11gR2 旧バージョンからのアップグレード

オラクル クルクルキャンペーン

あのOracle Database Enterprise Editionが超おトク!!

おトクな買い方
オラクル5年分

- ライセンス使用期間　を5年間に設定
- 初期のライセンスコストがなんと67%OFF　！
- テクニカル・サポート価格も53%OFF　！

Oracle Databaseの
ライセンス価格を大幅に抑えて
ご導入いただけます

多くのお客様でサーバー使用期間とされる
5年間にライセンス期間を限定

- 期間途中で永久ライセンスへ差額移行
- 5年後に新規ライセンスを購入し継続利用
- 5年後に新システムへデータを移行

Enterprise Editionはここが違う!!

- 圧倒的なパフォーマンス!
- データベース管理がカンタン!
- データベースを止めなくていい!
- もちろん障害対策も万全!

この機能でこの価格
ライセンスパック

- Oracle Databaseの機能を存分に使える!
- 2ノードRAC構成も可能!
- サーバー構成によって計4種類のパックから選べる!

詳しくはコチラ
http://www.oracle.co.jp/campaign/kurukuru/index.html

Oracle Direct 0120-155-096

お問い合わせフォーム
http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28

ORACLE

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　検索

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

※フォームの入力には、Oracle Direct Seminar申込時と同じ
　ログインが必要となります。
※こちらから詳細確認のお電話を差し上げる場合がありますので、ご登録されている連絡先が最新のものになっているか、ご確認下さい。

フリーダイヤル

0120－155－096

※月曜~金曜　9:00~12:00、13:00~18:00
（祝日および年末年始除く）

ORACLE