



ORACLE®

BI&DWH Day 新世代の分析を実現！情報分析基盤の全貌 DWHの性能向上を実現するデータベース・テクノロジー解説

日本オラクル株式会社 テクノロジー製品事業統括本部
データベースエバンジェリスト 柴田 長
2010年11月25日

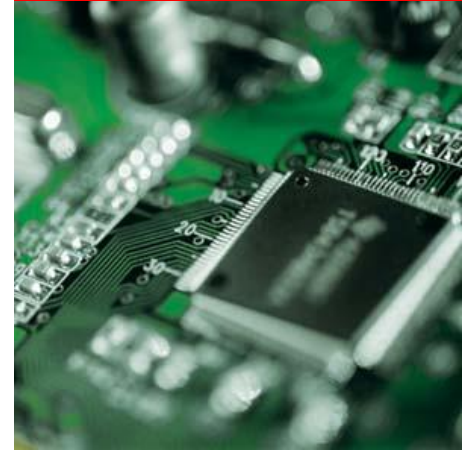
以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

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

- マルチコア化とデータベース性能の現状
- DWH向けのデータベース・テクノロジー
 - SQLの平行実行
 - データ圧縮
 - パーティショニング
 - マテリアライズド・ビュー
- まとめ

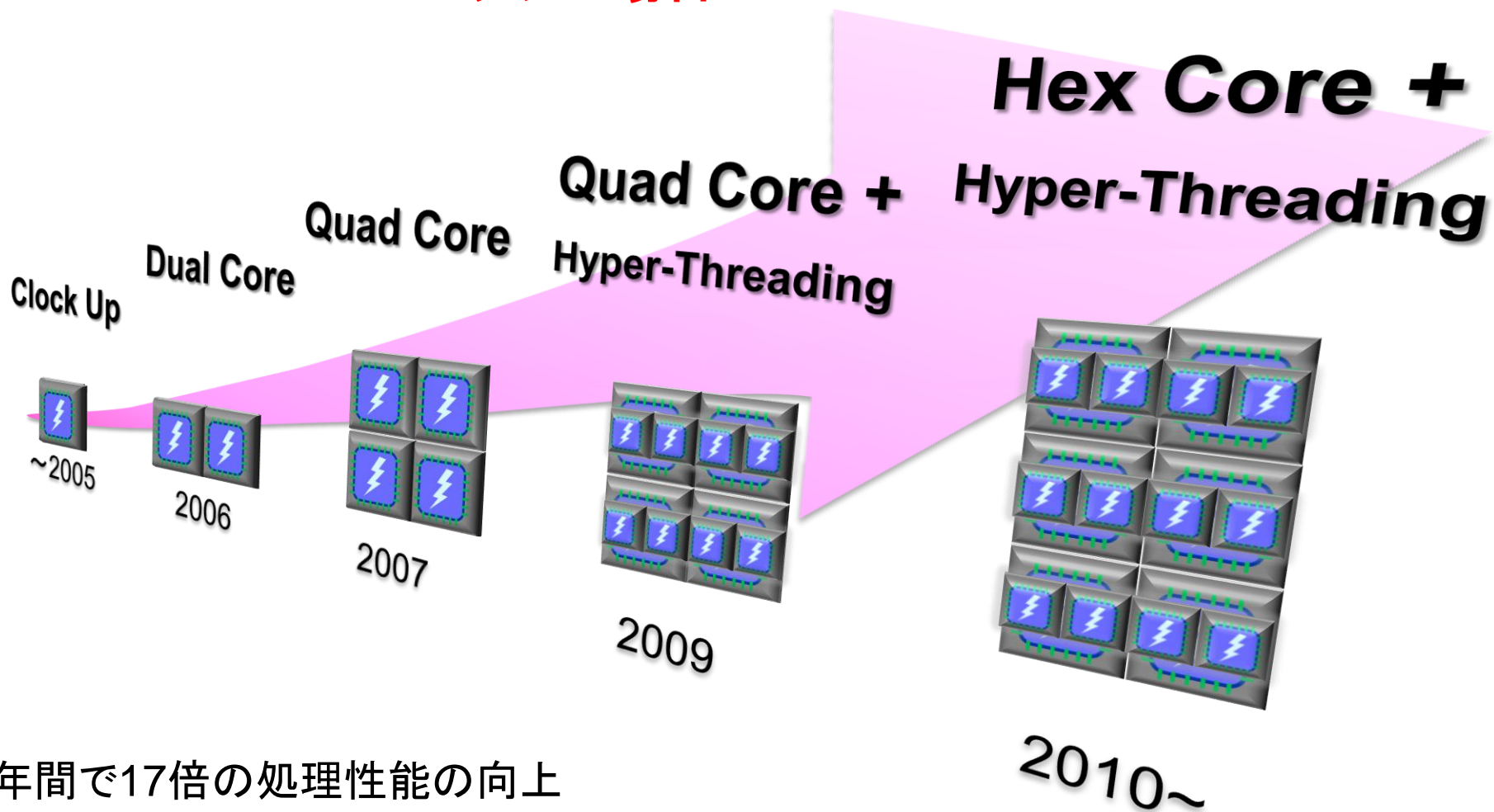
Oracle
GRID
Center



マルチコア化とデータベース性能の現状

マルチコア化

Intel Xeonプロセッサの場合



5年間で17倍の処理性能の向上

【参考】 http://www.intel.com/performance/server/xeon/server_table.htm

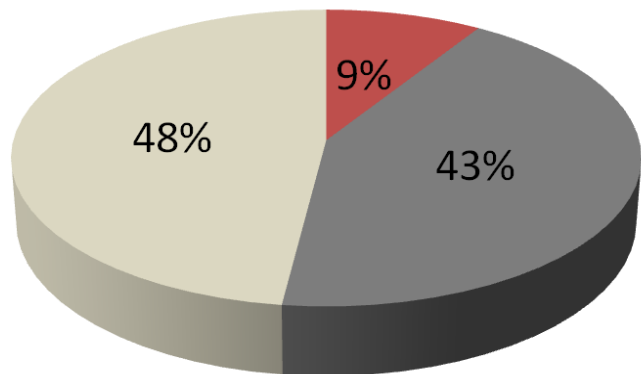
マルチコア化とデータベースの性能

Oracle Directのパフォーマンスクリニクの現状

CPUを追加すれば、性能問題は解決？

CPUがボトルネックだったケースは、わずか**9%**(*弊社統計)
→ **マルチコアを使いこなすことができない**

性能ボトルネックの原因の傾向



- CPU
- Storage I/O
- Complex
- CPU: 9%
- ストレージI/O: 43%
- 非効率なSQL文、索引の設計等 : 48%

*データ: Oracle Directが直近で実施したパフォーマンスクリニク

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

ある日の出来事

Slerの先輩から1本の電話が...

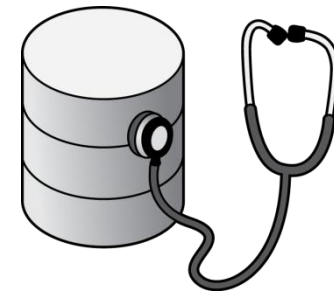
XXシステムからデータを抽出する
バッチ処理のSQLが遅いんだ。
チューニング方法を教えて!!



久々の電話が仕事
の電話でした...

ある日の出来事

Slerの先輩から1本の電話が...



XXシステムからデータを抽出する**バッチ処理のSQLが遅い**んだ。
チューニング方法を教えて!!

SQLの平行実行って知ってます？ **高速化**するか試してみたらどうですか？

ダメだった、**CPUは余っているけど**、全然変わらない。。。 (※注意)

ストレージI/O性能がボトルネックだと予想されるから、
データ圧縮を使用したら**高速化**するかも？

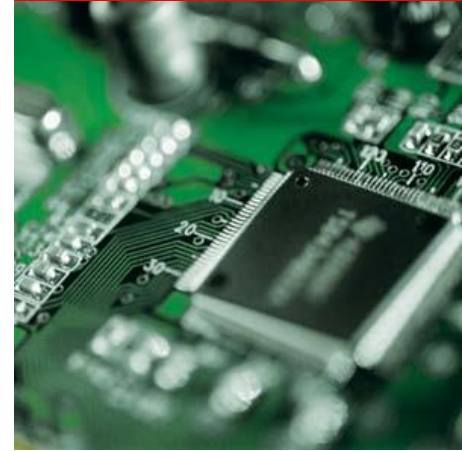
なんか、圧縮のコマンドを実行するとエラーがでるんだけど？

本当ですか？もしかして...
Oracle Databaseのエディションって、**Enterprise Edition**ではないのですか？

えっ何？エディションって。。。

※注意： 本当の理由は、Standard Editionは平行実行が使えないからです。

ORACLE



【DWH向けのデータベース・テクノロジー】

SQLの平行実行

Oracle GRID Center – テクニカル・ホワイト・ペーパー

http://www.oracle.co.jp/solutions/grid_center/nssol/index.html

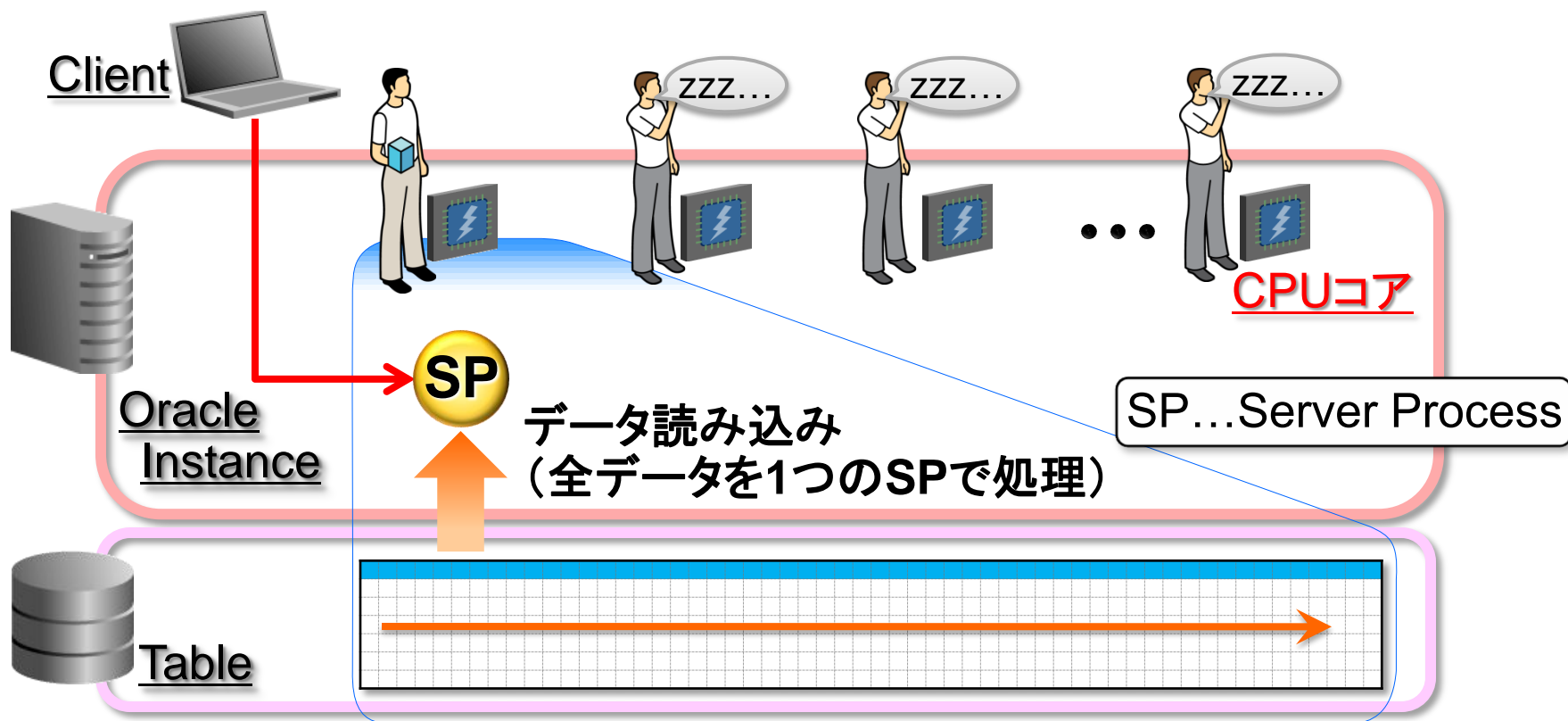
http://www.oracle.co.jp/solutions/grid_center/nec/index.html

ORACLE

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

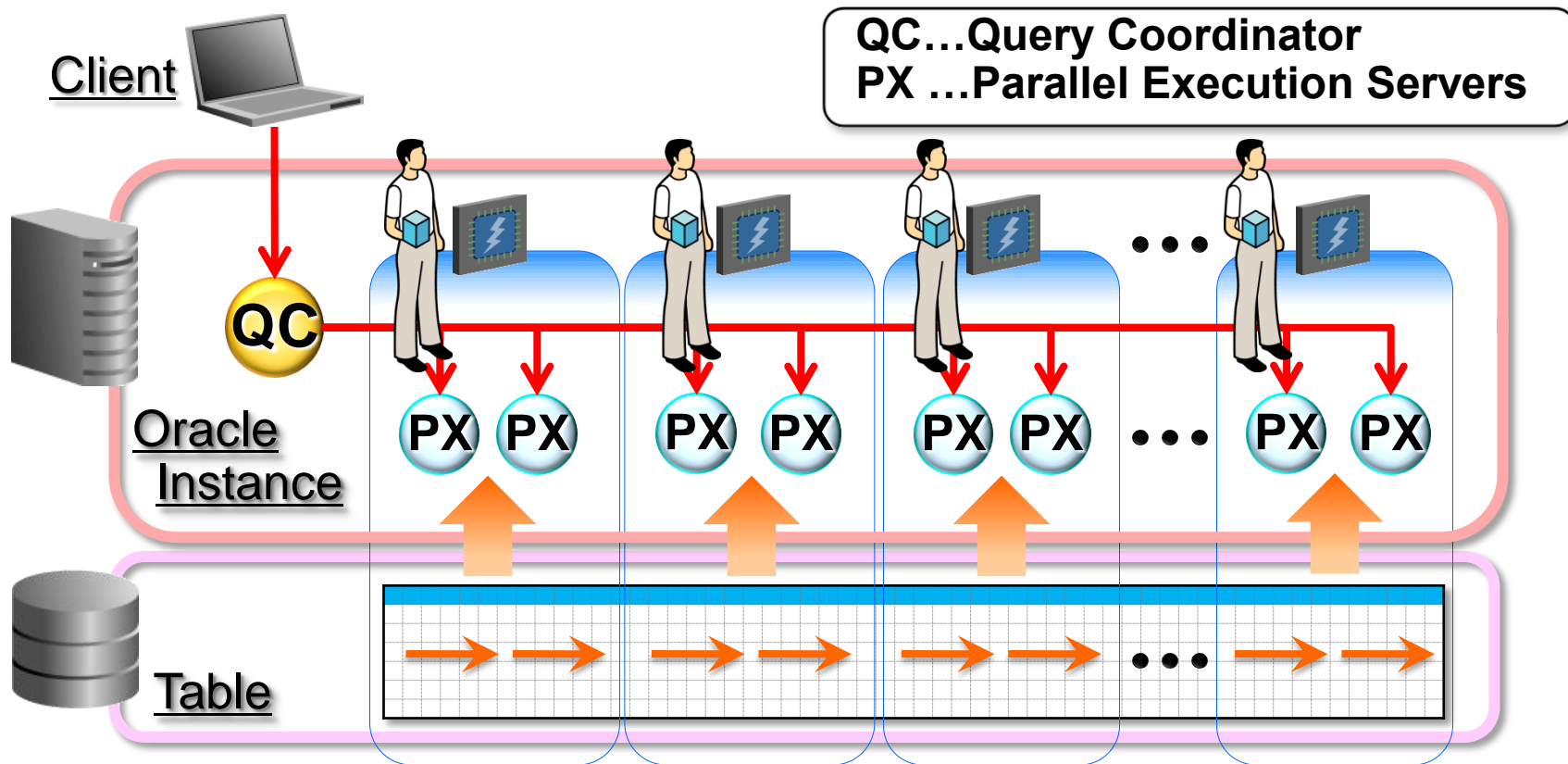
- **Standard Edition**ではSQLをシリアルで実行するため、1つのCPUコアしか使用しない。その為、CPUコアを追加しても性能向上は期待できない



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

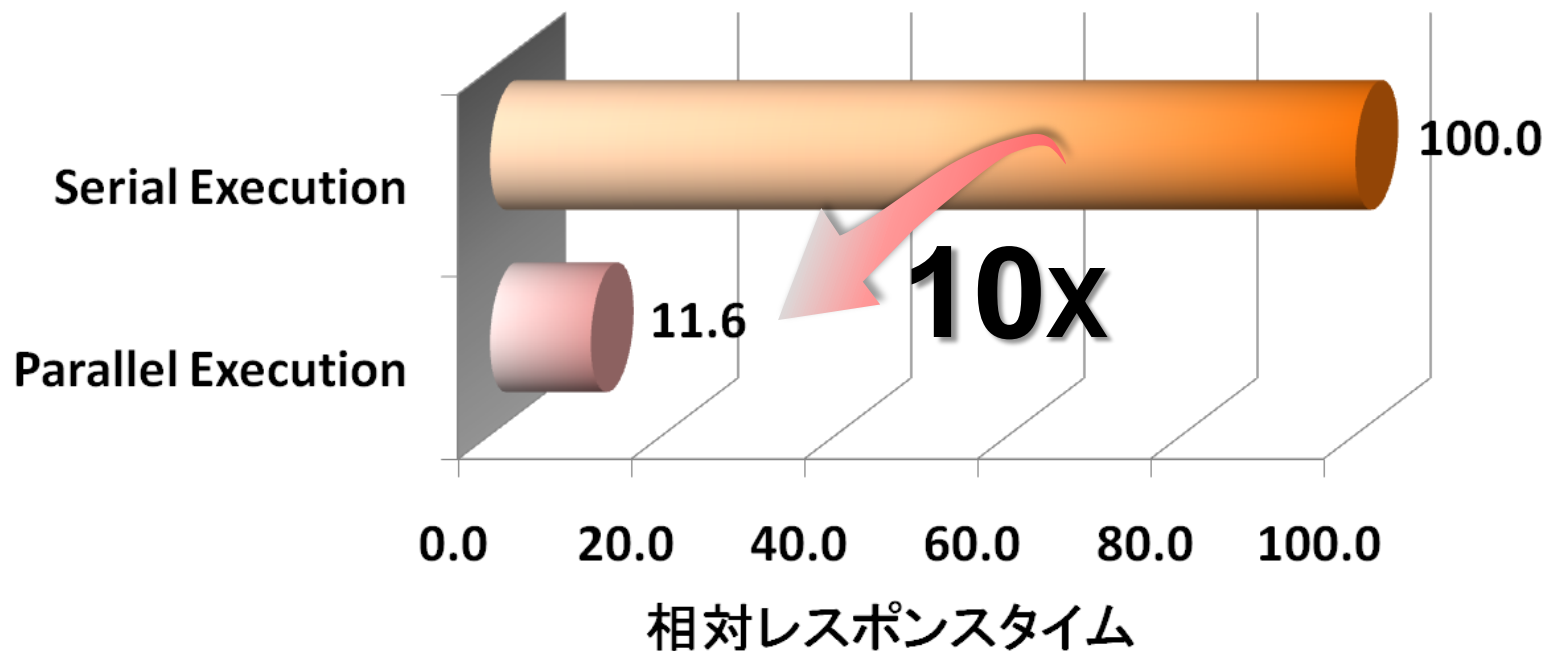


Parallel実行によるSQLの高速化

検証結果(レスポンスタイム)

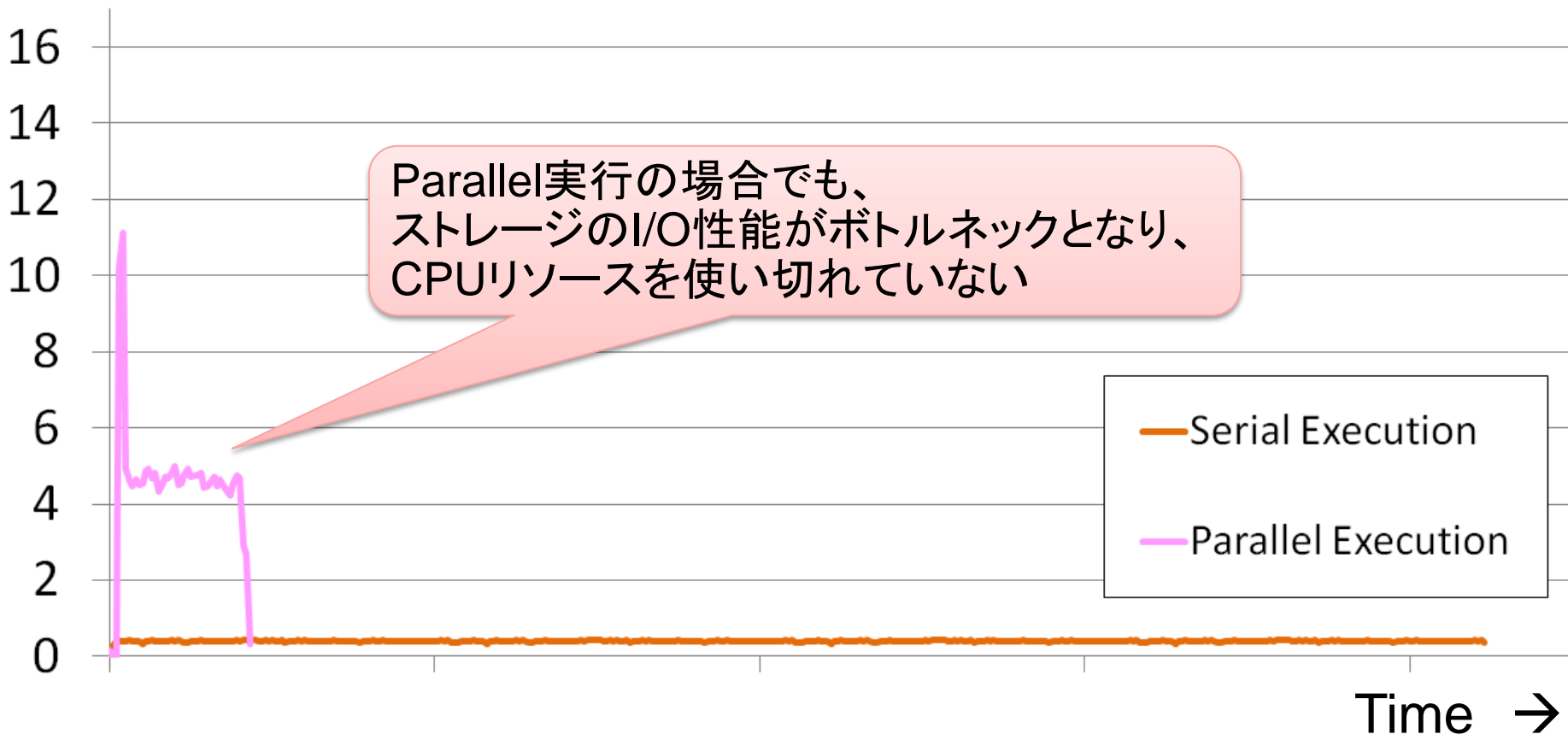
6000万件のデータの集計処理を

➤ Parallel Executionを使用することで、約**10倍高速**



Parallel実行によるSQLの高速化

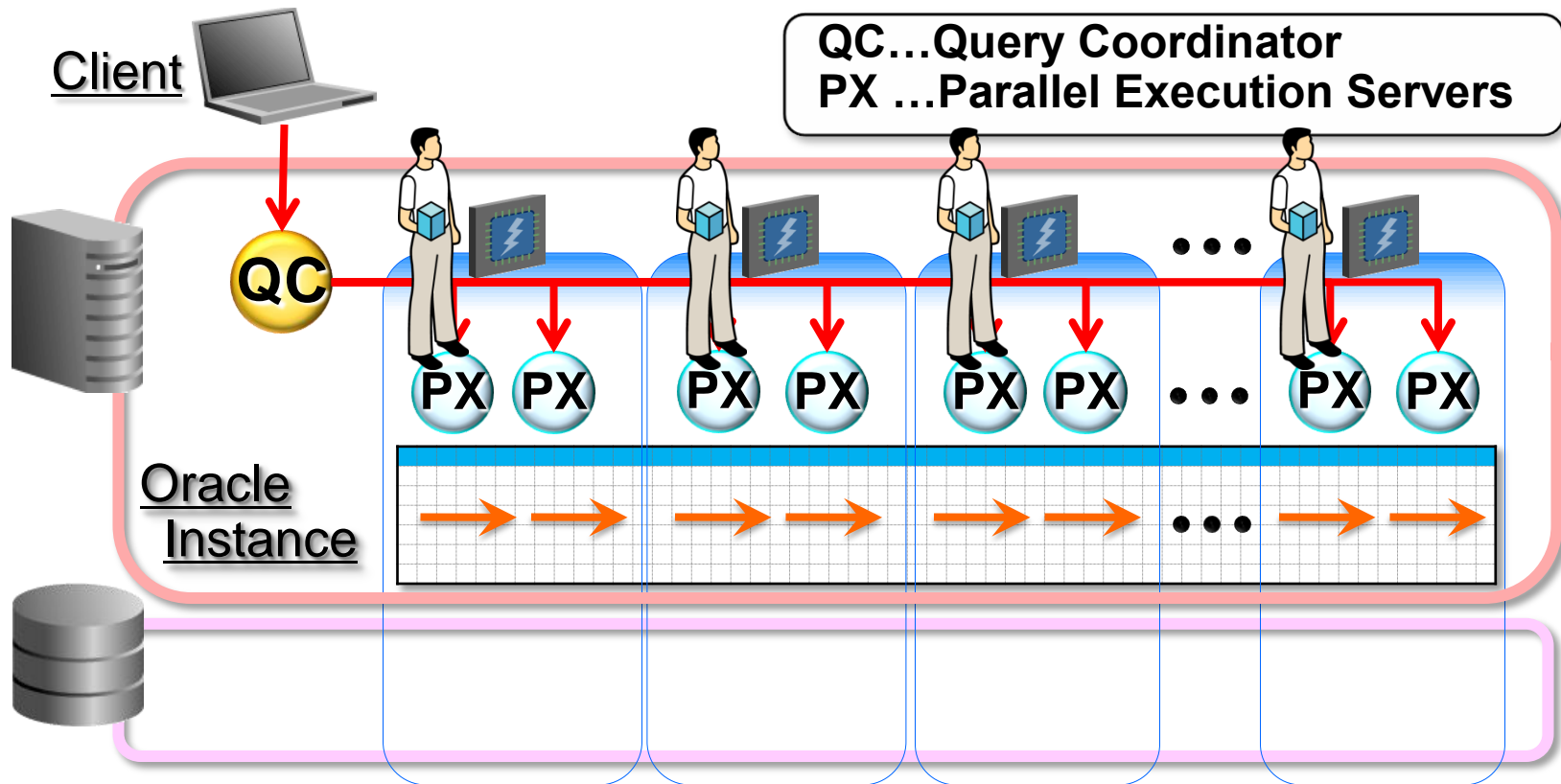
検証結果(CPU使用率)



In-Memory Parallel Execution

マルチコア性能のフル活用による更なるSQLの高速化

- 物理メモリ上にキャッシュされたデータに対するParallel実行により、ストレージの性能限界を排除した高速処理を実現

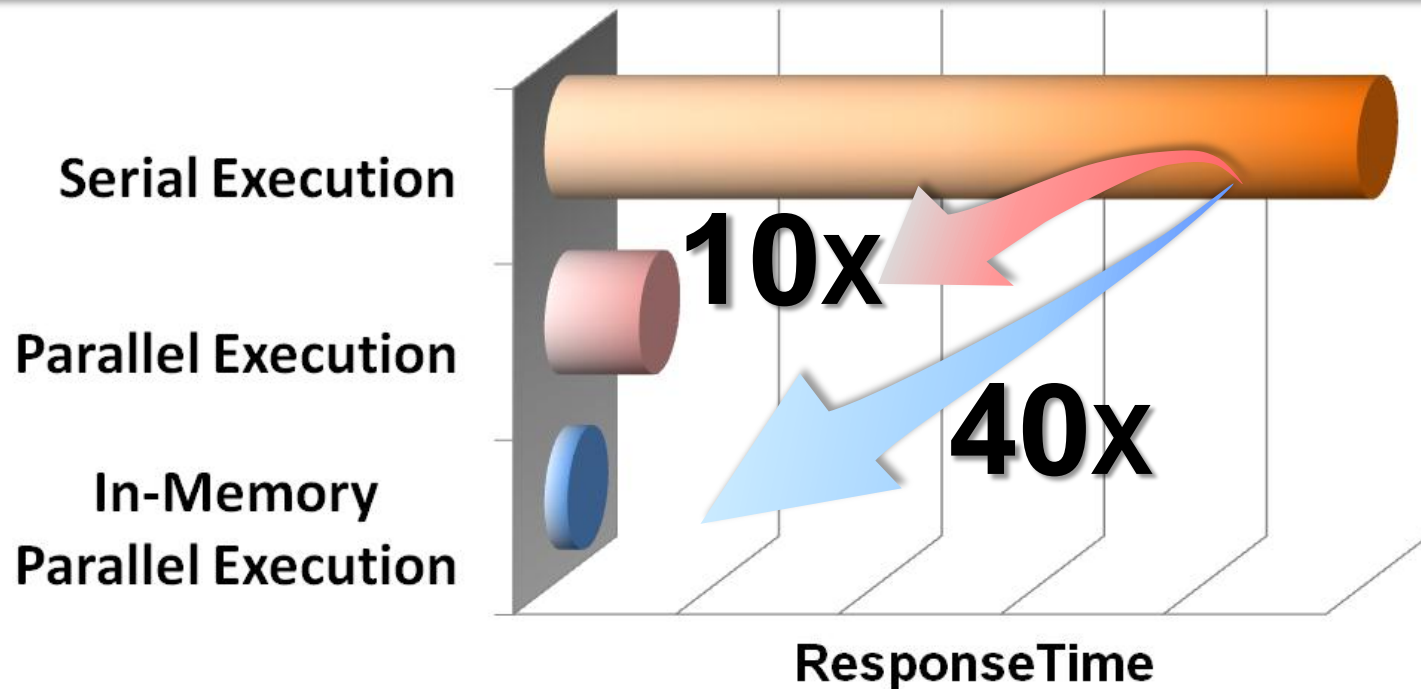


In-Memory Parallel Executionの効果

検証結果(レスポンスタイム)

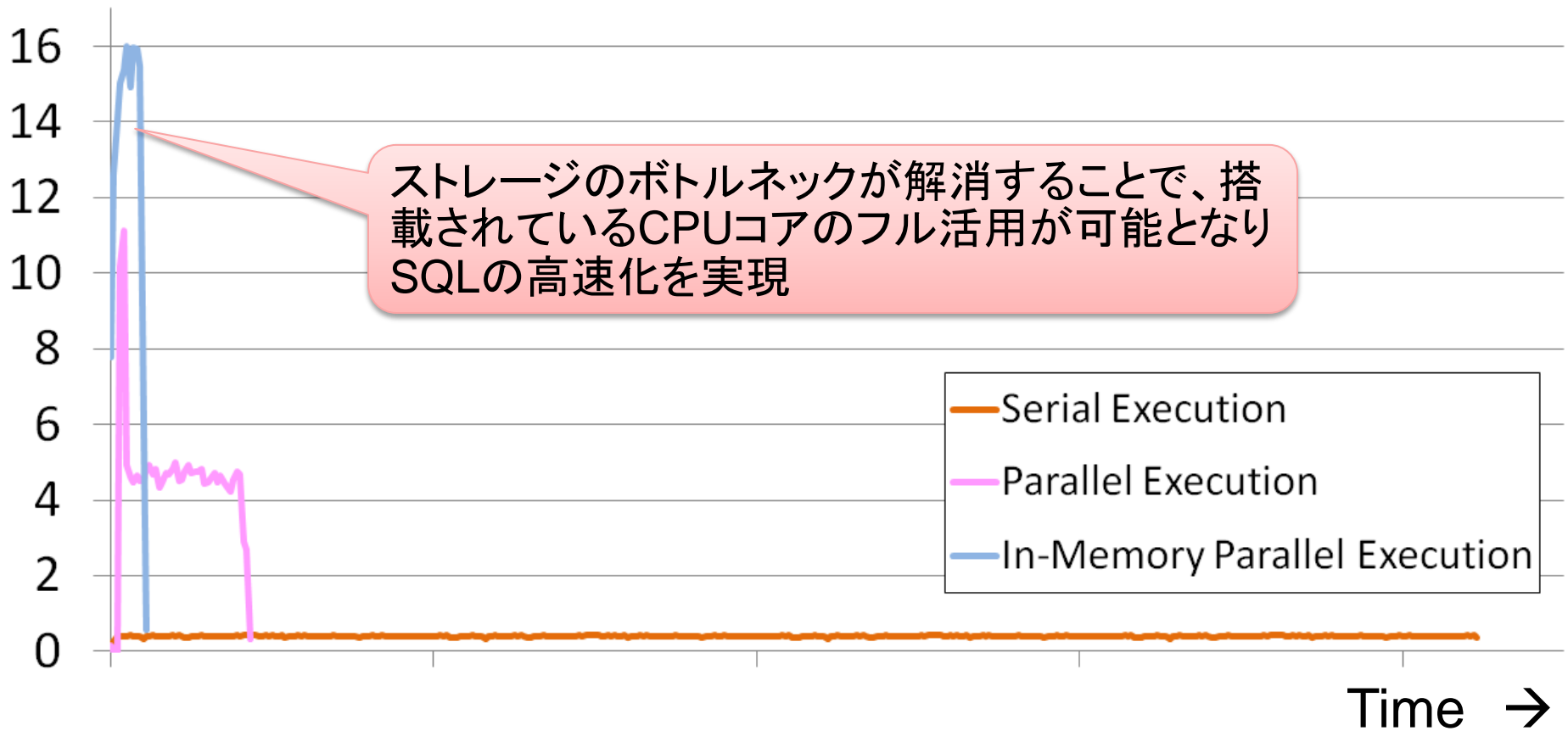
6000万件のデータの集計処理を

- Parallel Executionを使用することで、約**10倍高速**
- In-Memory Parallel Executionを使用することで約**40倍高速**



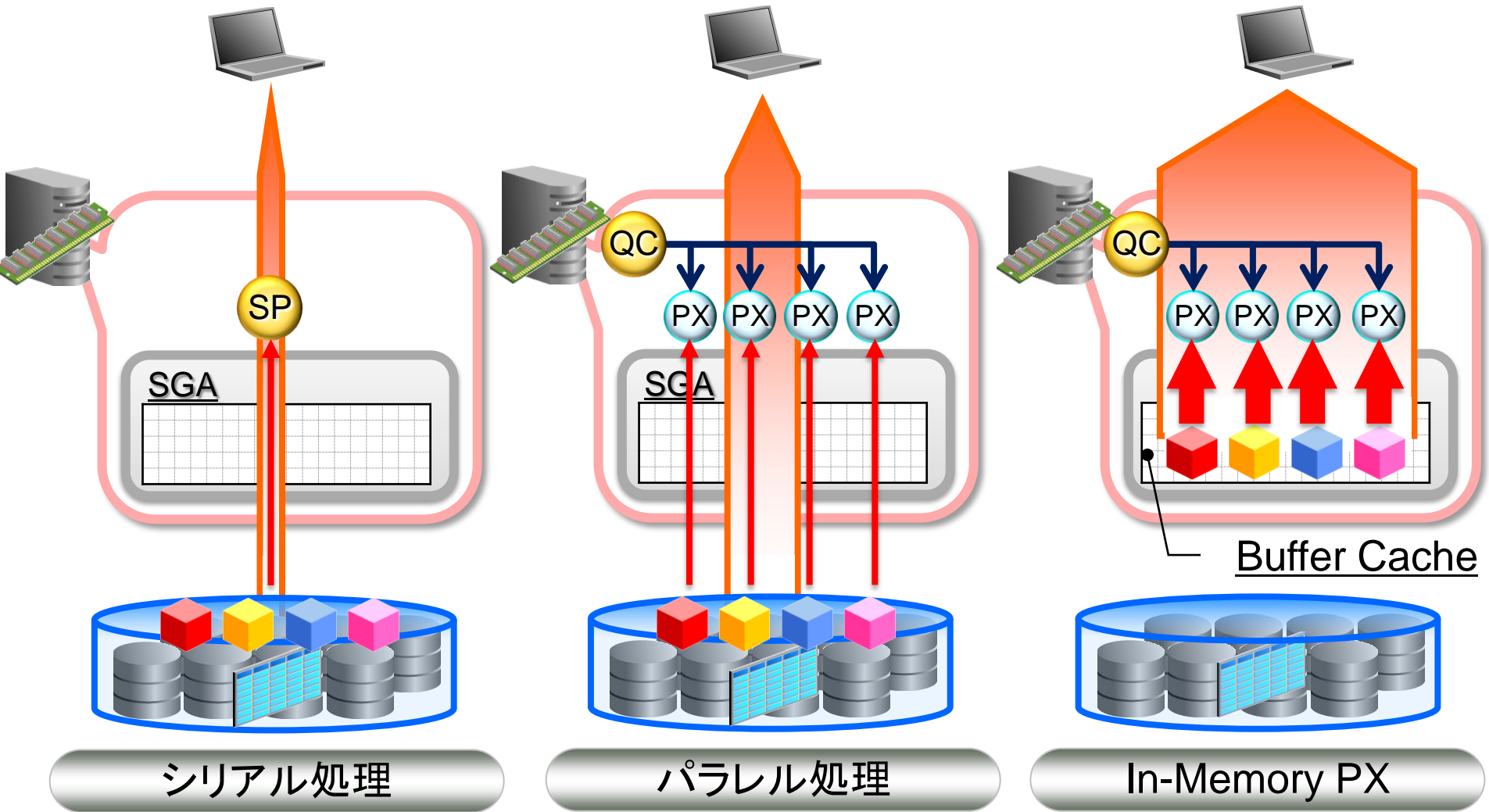
In-Memory Parallel Executionの効果

検証結果(CPU使用率)



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



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;
```

自動パラレル度設定

パラレル実行となるSQLの基準

- PARALLEL_MIN_TIME_THRESHOLD初期化パラメータ
 - このパラメータで設定されている値(単位:秒)以上の時間がシリアル実行で必要と判断された場合、自動パラレル実行の対象
 - デフォルト値:AUTO (10秒)
 - alter system文、alter session文での動的変更が可能

```
alter system set parallel_min_time_threshold=20 scope=both;
```

```
alter session set parallel_min_time_threshold=20;
```

といっても、実際にSQLがパラレル実行されているかわからない。。。 そんな時は、EMのSQL Monitoring

- 「パフォーマンス」タブ → 「SQL監視」

Overview

SQL ID: Scwn33jytkwtk
Parallel: 32
Execution Started: Mon Mar 15, 2010 10:09:54 PM
Last Refresh Time: Mon Mar 15, 2010 10:10:42 PM
Execution ID: 33554432
User: BT
Fetch Calls: 17

Time & Wait Statistics

Duration: 48.0s
Database Time: 21.3m
PL/SQL & Java: 0.0s
Wait Activity %: 100

IO Statistics

Buffer Gets: 22K
IO Requests: 22K
IO Bytes: 22K

Details

Plan Hash Value: 3684764808

Operation	Name	Estima...	Cost	Timeline(48s)	Exe...	Actu...	Mem...	Tem...	IO Bytes	CPU Activity %	Wait Activity %
SELECT STATEMENT					65	240					
PX COORDINATOR					65	240					2.13
PX SEND QC (RANDOM)	:TQ10002	102	12K		32	240					
HASH GROUP BY		102	12K		32	240	362MB				
PX RECEIVE		102	12K		32	7,680					
PX SEND HASH	:TQ10001	102	12K		32	7,680					
HASH GROUP BY		102	12K		32	7,680	465MB			35	
HASH JOIN		46M	12K		32	60M	3GB			28	
PX RECEIVE		399K	67		32	13M				1.42	
PX SEND BROADCAST	:TQ10000	399K	67		32	13M				2.37	
PX BLOCK ITERATOR		399K	67		32	398K					
TABLE ACCESS FULL	DIM3	399K	67		384	398K					
PX BLOCK ITERATOR		46M	12K		32	60M					
TABLE ACCESS FULL	FACT	46M	12K		645	60M			13GB	33	98

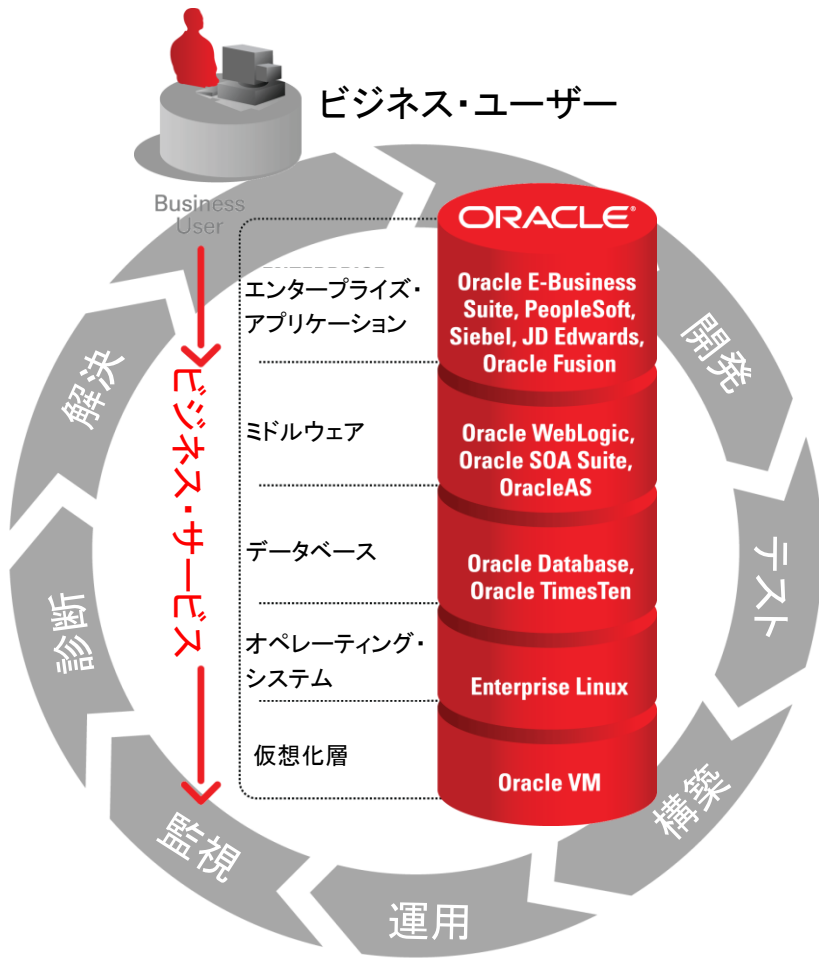
User I/O: direct path read - 1,057 samples (98%)

ORACLE Enterprise Manager 11g
Database Control
Oracle Database 11g
監視されたSQL実行

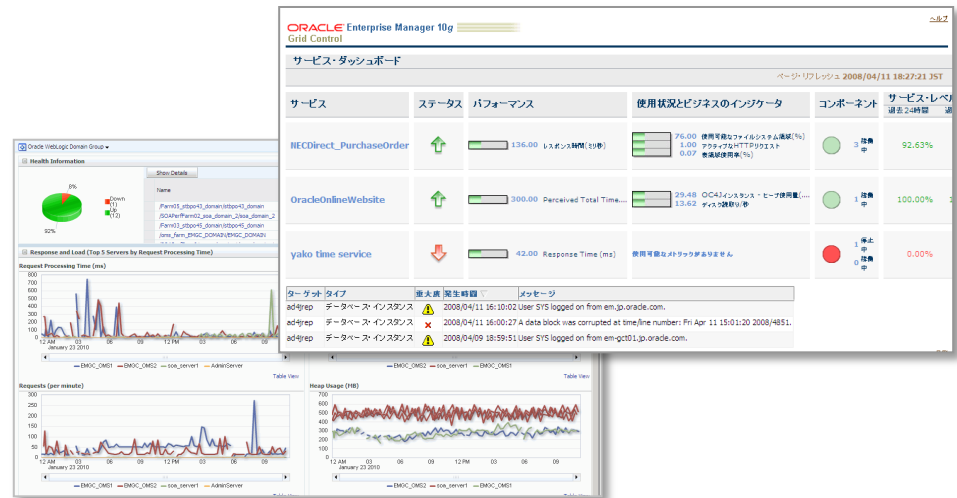
ステータス	期間	インスタンス	SQL ID	ユーザー	パラレル	データベース時間	IOリクエスト
🟢	12.0s	1	1j0nkdxn5lpp	TEST	10/3	11.9s	837
🟢	51.0s	1	1j0nkdxn5lpp	TEST	10/3	5.6m	951
🟢	1.0m	1	1j0nkdxn5lpp	TEST	10/3	6.3m	1309
🟢	1.1m	1	1j0nkdxn5lpp	TEST	10/3	8.4m	1478
🟢	1.8m	1	1j0nkdxn5lpp	TEST	10/3	9.9m	1828
🟢	1.9m	1	1j0nkdxn5lpp	TEST	10/3	11.8m	2402
🟢	2.3m	1	1j0nkdxn5lpp	TEST	10/2	19.8m	2000
🟢	2.2m	1	1j0nkdxn5lpp	TEST	10/3	20.5m	2000

オラクルのシステム管理ソリューション

Oracle Enterprise Manager



- システムライフサイクル全体の効率化
 - プロアクティブな監視
 - 深い診断と問題解決へのアドバイス
 - オペレーションの自動化による効率化
 - コンフィギュレーションの一元管理
 - サービスレベル管理
 - 仮想化環境への対応

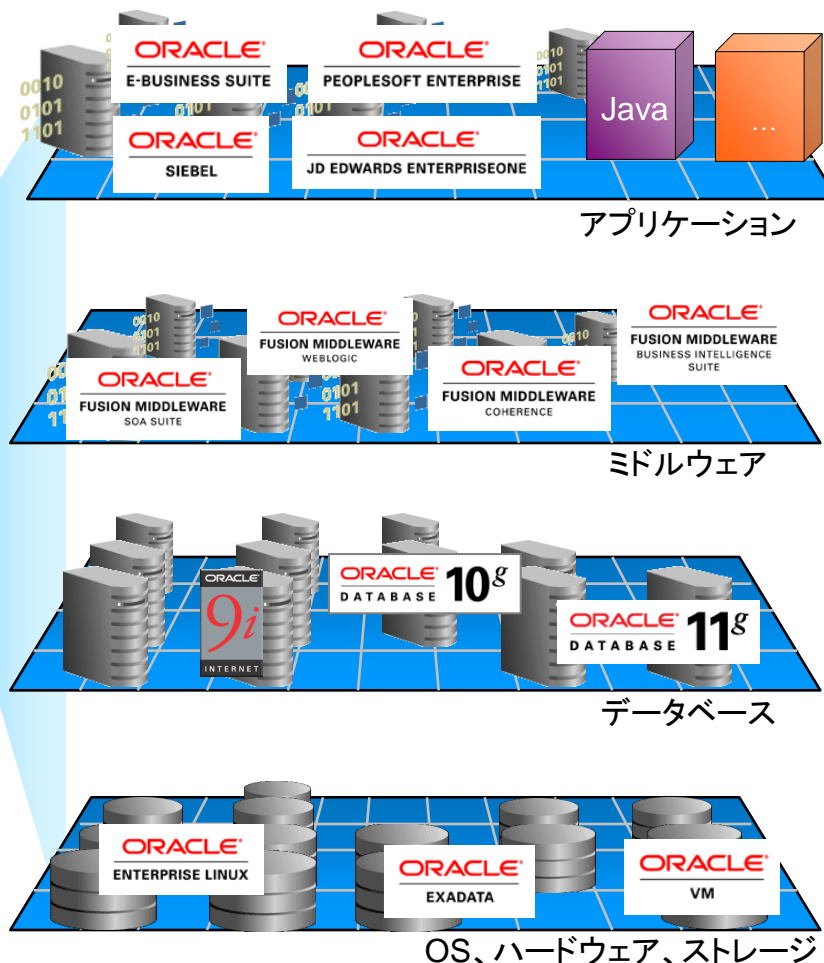
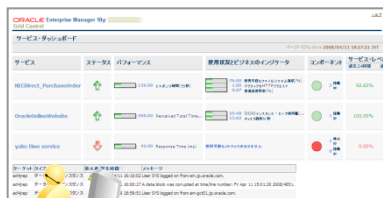


Oracle Enterprise Managerによる統合管理

Oracle Enterprise Manager

統合管理

- ・単一のUI
- ・監視ダッシュボード
- ・レポート(性能、セキュリティ等)
- ・ジョブ



アプリケーションの管理

- ・Javaアプリケーション詳細分析
- ・EBS パッチ管理
- ・Siebel性能ボトルネック分析
- ・構成管理
- ・ユーザー視点での性能監視

ミドルウェアの管理

- ・監視(可用性、リソース、性能)
- ・JVMの詳細分析
- ・SOA環境の可視化
- ・構成管理

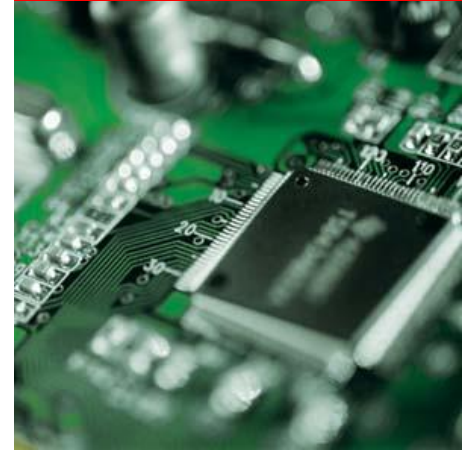
データベースの管理

- ・管理(バックアップ、領域追加など)
- ・監視(可用性、性能、リソース)
- ・分析、アドバイス、チューニング
- ・パッチ適用
- ・構成管理、オブジェクト管理
- ・機密情報のマスキング

その他の管理

- ・OS(Unix/Windows/Linux)の監視
- ・ストレージ(NetApp/EMC)の監視
- ・構成管理

ORACLE



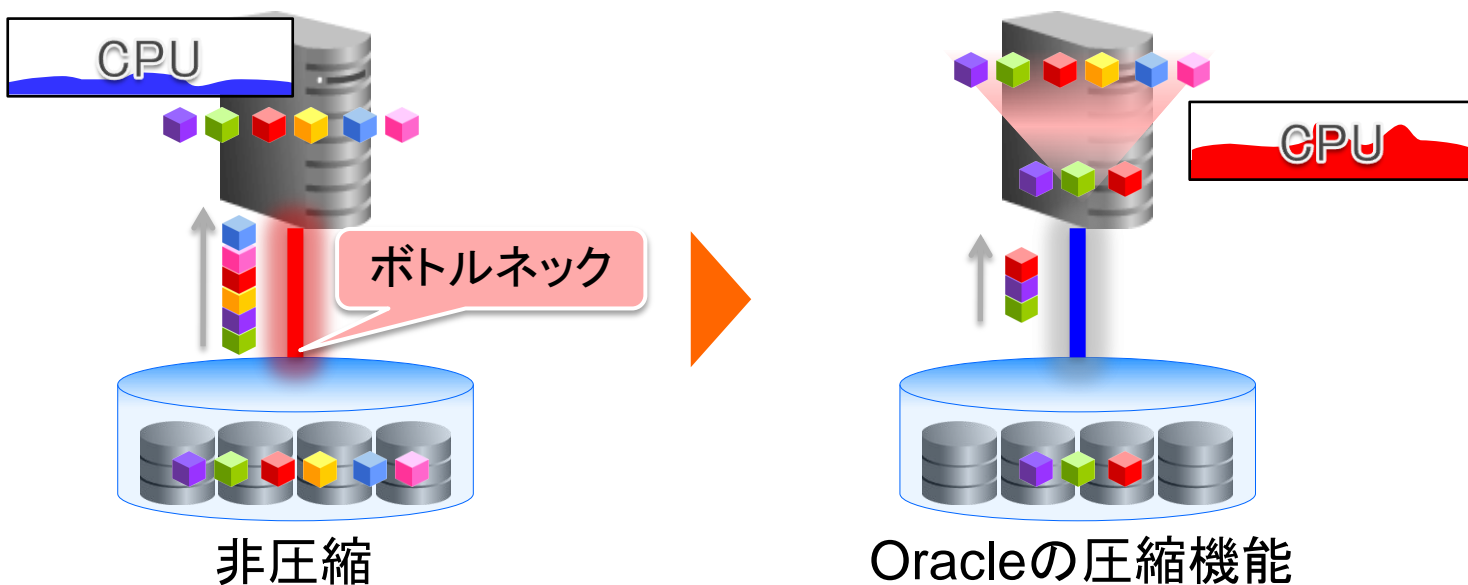
【DWH向けのデータベース・テクノロジー】

データ圧縮

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

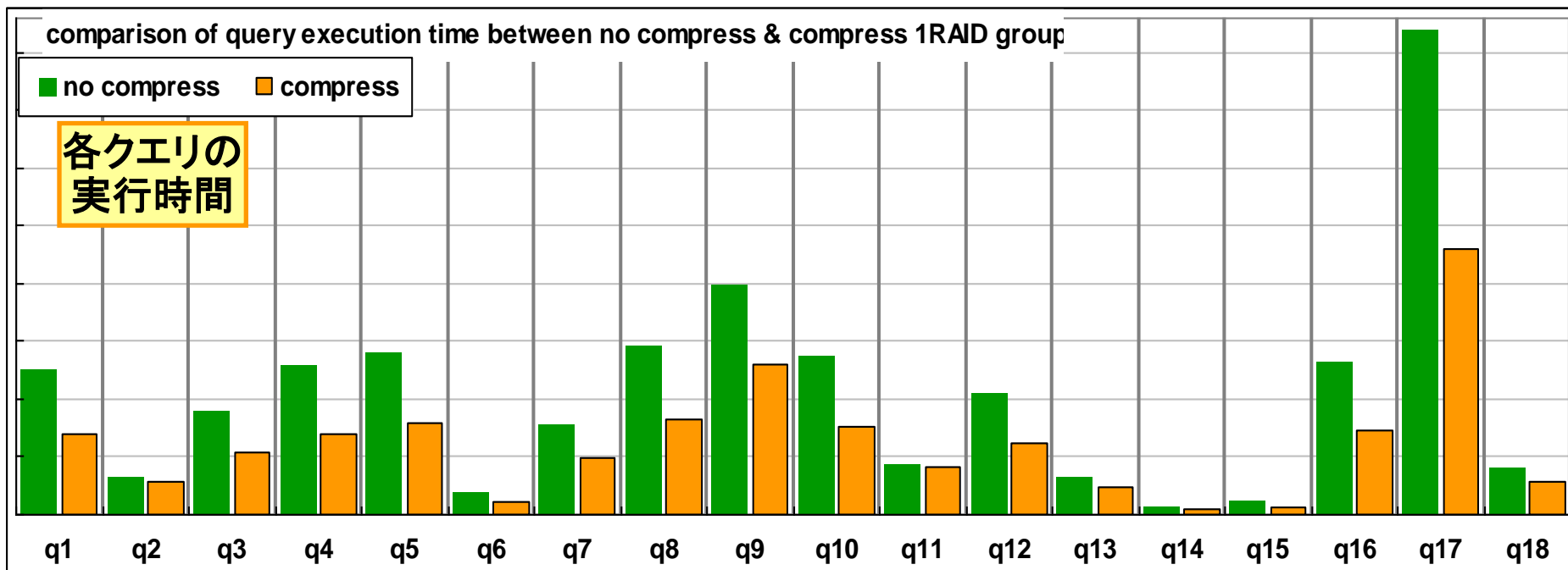
- Oracle Databaseの圧縮機能は、H/Wリソースを有効活用
 - データ量を大幅に縮小可能
 - サーバー側でデータを展開する仕組み
- ストレージI/O性能のボトルネックを解消し性能向上



データ圧縮の効果

検証結果(レスポンスタイム)

- データ圧縮による検索性能の向上を確認
 - 圧縮率の高い表に対する処理の性能向上幅が高い傾向



【参考】 <http://primeserver.fujitsu.com/sparcenterprise/documents/data/pdf/fj-gc-spe-dwh-1.2.pdf>

Advanced Compression Option

圧縮機能一覧

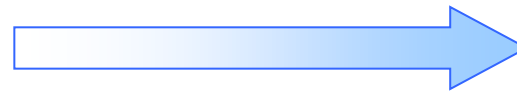
- 最大限のリソース活用とコスト削減を支援する包括的な圧縮機能 (Oracle Database 11g~)

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



Advanced Compression Option

OLTP圧縮と従来の表圧縮

- データブロックの単位で重複データを除外するOracle独自のアルゴリズム
- アプリケーション層へは、非圧縮時と同じ元のデータを提供
 - アプリケーション側の設定変更は不要
- 従来の表圧縮
 - Oracle Database 9i R2~ (Enterprise Editionの標準機能)
 - データ挿入時、ダイレクト・ロード時のみにデータ圧縮を行う
 - DWHへのETLバッチ処理などで利用
- Advanced CompressionのOLTP圧縮
 - Oracle Database 11g R1~ (EE+Advanced Compression Option)
 - どのようなデータ・ロードでも圧縮対象
 - リアルタイムDWHを実現する際にも圧縮が可能

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)

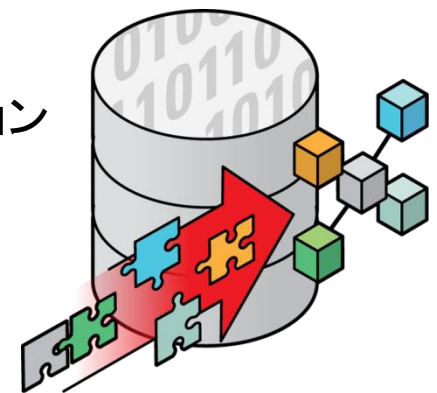
BEGIN
  DBMS_REDEFINITION.FINISH_REDEF_TABLE('SH','SALES', 'SALES2');
END;
/

drop table SALES_TMP;
```


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

パーティションを限定する場合は、partnameにパーティション名を設定

見積もる圧縮タイプを設定

- 非圧縮 → 1
- OLTP表圧縮 → 2

```
DBMS_COMPRESSION.GET_COMPRESSION_RATIO (  
  SCRATCHTBSNAME      VARCHAR2      IN  
  OWNNAME              VARCHAR2      IN  
  TABNAME              VARCHAR2      IN  
  PARTNAME             VARCHAR2      IN  
  COMPTYPE             NUMBER        IN  
  BLKCNT_CMP          BINARY_INTEGER OUT  
  BLKCNT_UNCMP        BINARY_INTEGER OUT  
  ROW_CMP              BINARY_INTEGER OUT  
  ROW_UNCMP           BINARY_INTEGER OUT  
  CMP_RATIO           NUMBER        OUT  
  COMPTYPE_STR        VARCHAR2      OUT);
```

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

OLTP表圧縮

Oracle Advanced Compression Advisor (9i R2 ~11g R1)

- アップグレード前に、DBMS_COMPRESSIONパッケージを使用して、どの程度圧縮されるのかを見積もることが可能
 - 以下のサイトから、Create package文をダウンロード+適用
 - 注意: Enterprise Editionのみに対応

Oracle Advanced Compression Advisor

Overview

Oracle Advanced Compression Advisor provides an estimate of potential storage savings that can be realized through the use of OLTP Table Compression feature (part of Advanced Compression option). This estimate is based on analysis of a sample of data and provides a good estimate of the actual results you may obtain once you implement OLTP Table compression feature in your environment.

Using Compression Advisor

<http://www.oracle.com/technology/products/database/compression/compression-advisor.html>

ダウンロード

ORACLE

OLTP表圧縮

Oracle Advanced Compression Advisor (9i R2 ~11g R1)

- 実行例

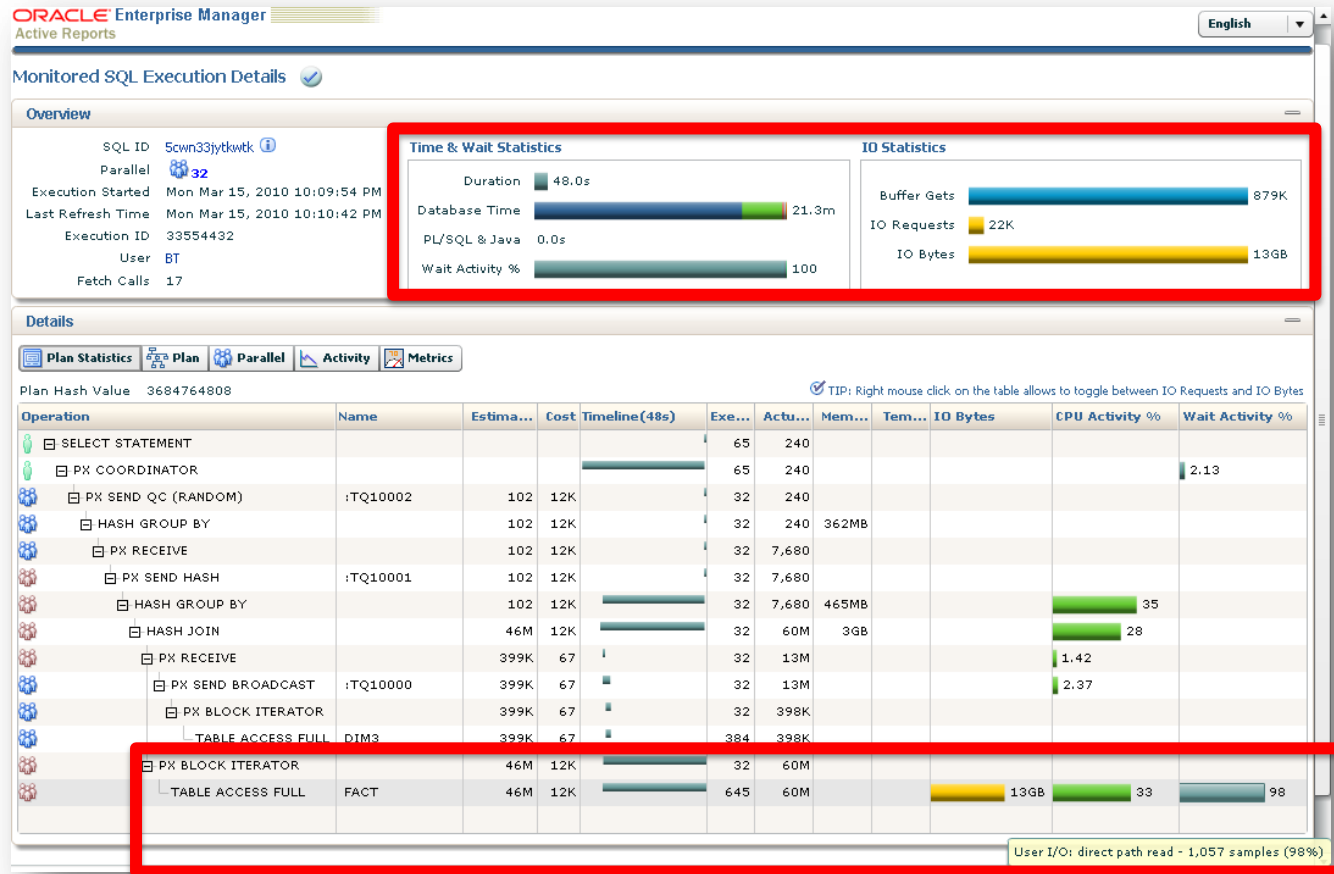
```
SQL> set serveroutput on
SQL> exec DBMS_COMPRESSION.getratio('SH','SALES','OLTP',10);

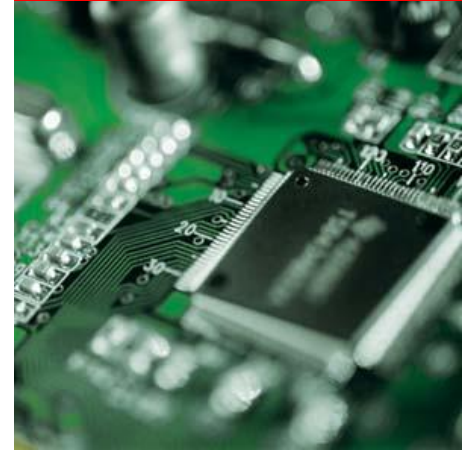
Sampling table: SH.SALES
Sampling percentage: 10%
Compression Type: OLTP
Expected Compression ratio with Advanced Compression Option: 2.96
```

- プロシージャ内部で、実際に圧縮表と非圧縮表を作成して各セグメントのサイズを比較する為
 - Enterprise Editionでしか使用できない
 - 既に圧縮済みの表に対して実行することで、圧縮の効果の確認も可能

といっても、実際にSQLがストレージI/Oボトルネックかわからない。。。 そんな時は、EMのSQL Monitoring

- 「スキーマ」タブ → 「表」



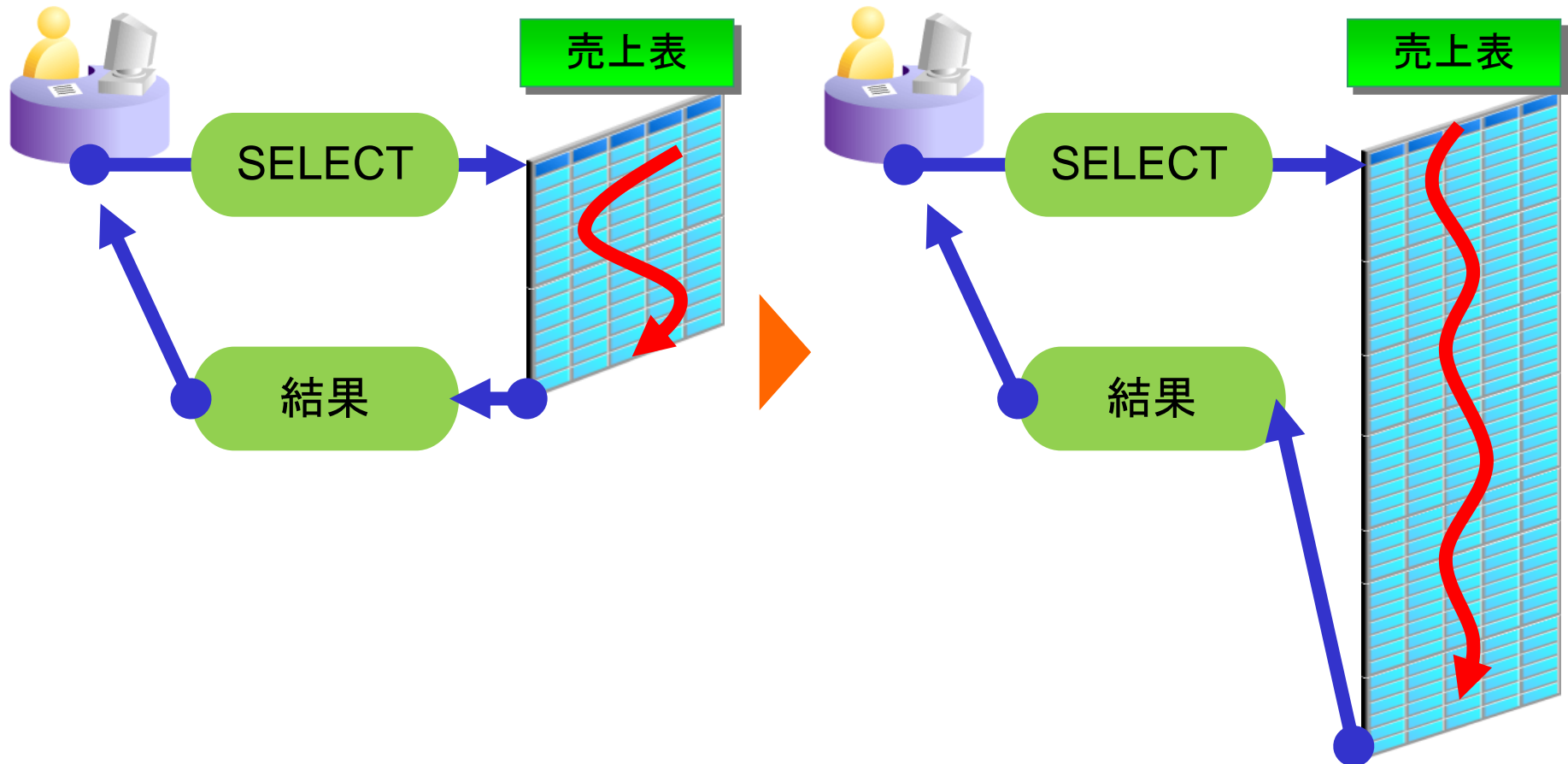


【DWH向けのデータベース・テクノロジー】 パーティショニング

データ量増大により発生する課題

検索パフォーマンスの劣化

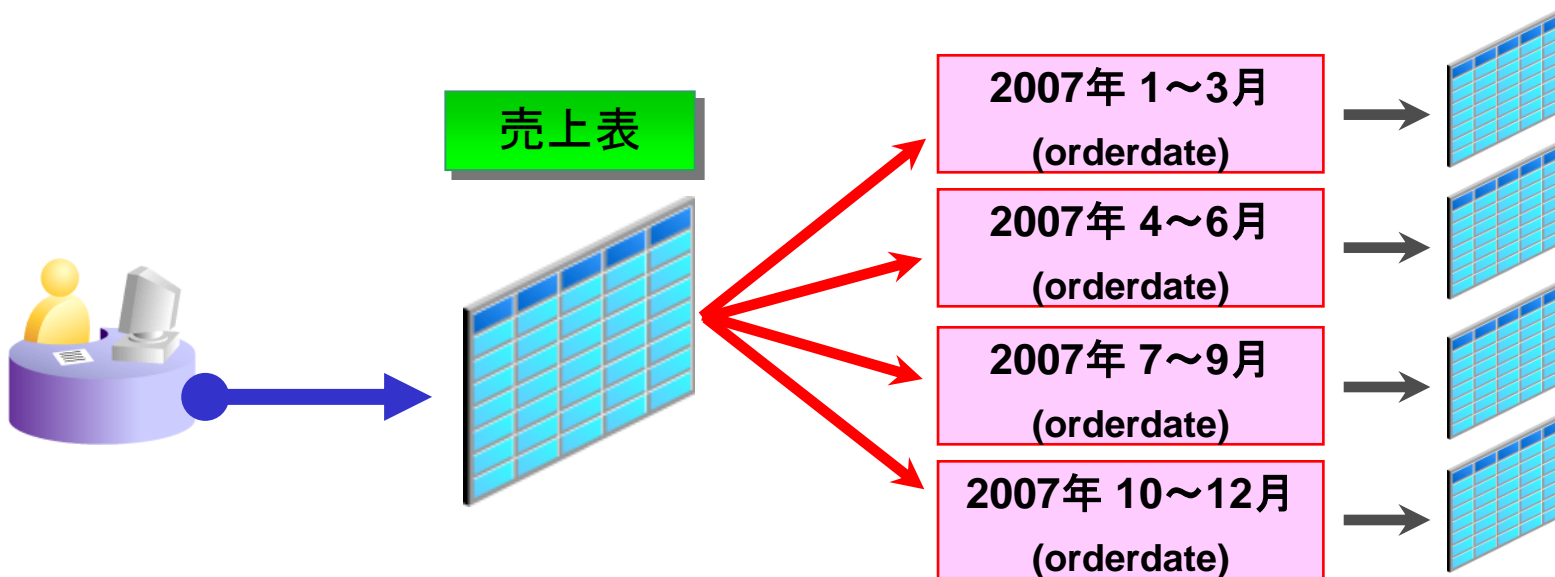
- データ量が増大することにより、パフォーマンス問題が顕著になる傾向



パーティショニング

表を分割して管理

- 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に使用可能
- 例えば、売上表を期間に応じてパーティション化した場合



パーティショニング

パーティション・プルーニング (読み込み対象データの限定)

今期(Q4)の売上の
平均値を見たい

SELECT

オプティマイザ

結果

売上表

Q1

(sales date)

Q2

(sales date)

Q3

(sales date)

Q4

(sales date)

```
SELECT area, period, avg(sales_rev) ...  
FROM sales_history  
WHERE sales date between 10 and 12  
GROUP BY area, period ...
```

ORACLE

パーティション・プルーニングの効果

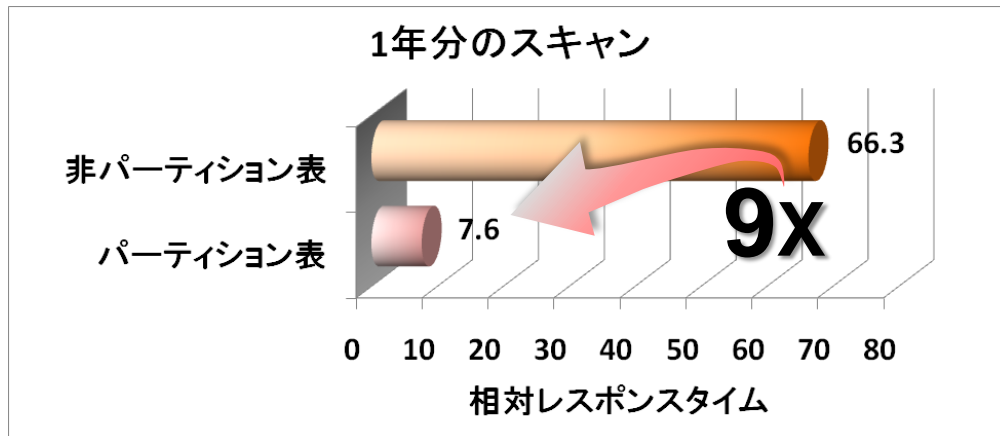
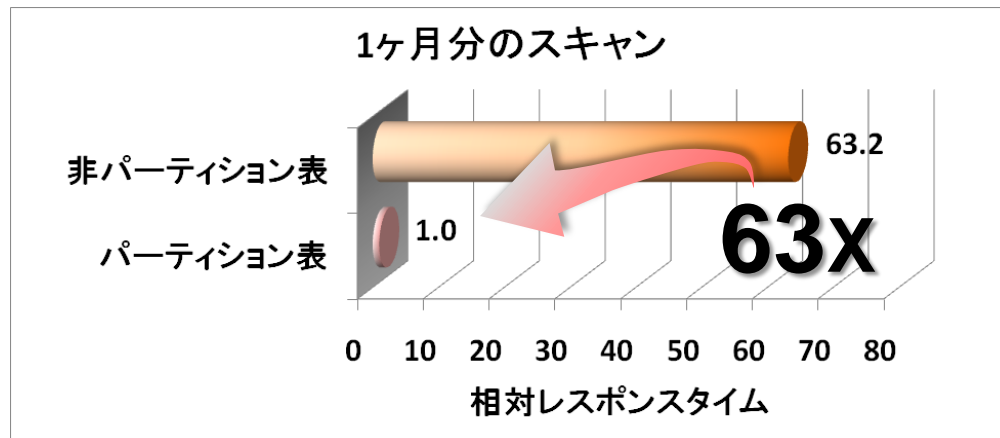
検証結果(レスポンスタイム)

- 7年間分のデータを保持する表を1ヶ月単位でパーティション化



- 読み込み対象データが限定されることで、大幅な性能向上を確認

- 1ヶ月分の検索: **63倍** 高速化
- 1年分の検索: **9倍** 高速化



【参考】 http://www.oracle.com/technology/global/jp/tech/grid/doc/Oracle_DB_ILM_WhitePaperV1.51_ja.pdf

パーティションの種類

- 基本パーティション
 - ✓ レンジ・パーティション: **値の範囲**でデータを分割
 - ✓ リスト・パーティション: **値の集合(種類)**でデータを分割
 - ✓ ハッシュ・パーティション: **値のHASH値**でデータを均等に分割
- 組み合わせパーティション(コンポジット・パーティション)

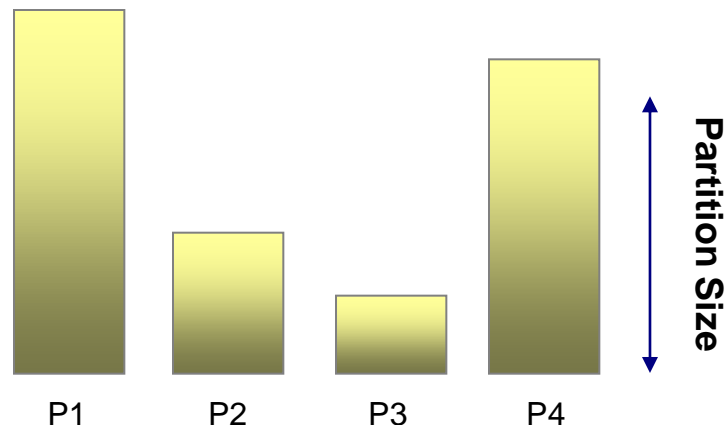
		サブ		
		レンジ	リスト	ハッシュ
メイン	レンジ	11g R1	9i R2	8i
	リスト	11g R1	11g R1	11g R1

パーティションと平行実行

ハッシュ・パーティションのメリット

レンジ、リストパーティション

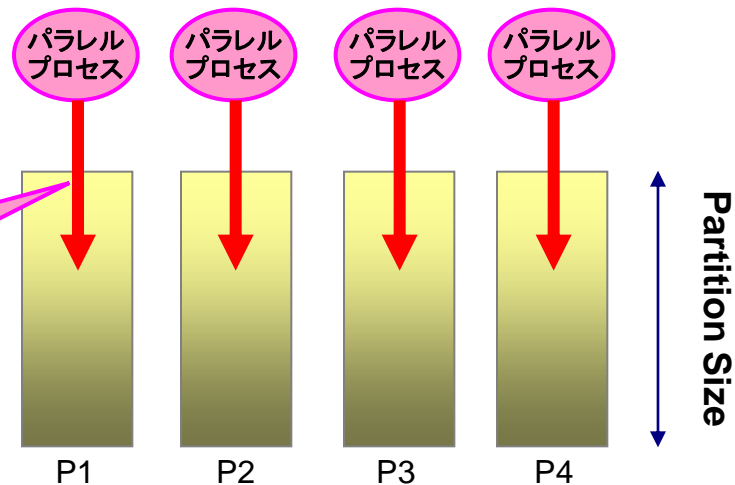
データによっては、
各パーティションサイズが均等ではない



ハッシュパーティション

各パーティションサイズがほぼ均等になり、
パーティション毎のI/Oバランスの調整が可能

各平行プロセスの処理量が均一化され、
最適化された平行実行が可能



といっても、実際にどの表をどの種類でパーティション化すれば。。。 そんな時は、EMのパーティション・アドバイザー

- ワークロード(システムで実行されるSQL文)を解析し、適切なパーティション構成を推奨する機能

ORACLE Enterprise Manager 11g Database Control

データベース

タスクの結果: SQLACCESS8668493

SYSTEMとしてログイン

推奨: 10

SQLアクセス・アドバイザーは、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。読み取り専用として表示されている名前や依存性を編集すると、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

すべてのアクション用に表領域を設定

実行ステータス	アクション	オブジェクト名	オブジェクトの属性	索引付けされた列	ベース表	スキーマ	表領域	パーティションキー	SQLパーティション	推奨使用表領域(MB)
■	PARTITION TABLE	LINEITEM				TPCHTEST		(L_SHIPDATE)	PARTITION BY RANGE (L_SHIPDATE) INTER...	0.0
■	PARTITION TABLE	ORDERS				TPCHTEST		(O_ORDERDATE)	PARTITION BY RANGE (O_ORDERDATE) INTER...	0.0
■	PARTITION TABLE	CUSTOMER				TPCHTEST		(C_NATIONKEY)	PARTITION BY RANGE (C_NATIONKEY) INTER...	0.0
✓	RETAIN INDEX	LL_ORDERKEY	BTREE	L_ORDERKEY	TPCHTEST.LINEITEM	TPCHTEST				11,545.6
■	CREATE INDEX	CUSTOMER_IDX\$018B0006	BTREE	C_MKTSEGMENT LOCAL_CUSTKEY	TPCHTEST.CUSTOMER	TPCHTEST				628.3

推奨の影響を受けるSQL:10

文ID	文	元のコスト	新規コスト	コスト改善率	コスト改善率(%)	実行回数
20	select * from select_l_orderkey, sum(l_extendedprice * (1 - l_discount)) as revenue, o_orderdate, o_shippriority from customer, orders, lineitem where c_mktsegment = 'HOUSEHOLD' and c_custkey = o_custkey and l_orderkey = o_orderkey and o_orderdate < date '1995-03-06' and l_shipdate > da...	4059888	673426	3385562	83.41	1

性能向上のための、パーティション構成がアドバイスされる

ORACLE Enterprise Manager 11g Database Control

SQLアクセス・アドバイザー: ワークロード・ソース

データベース: orapar

ログイン時の権限: SYSTEM

分析に使用するワークロードの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

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

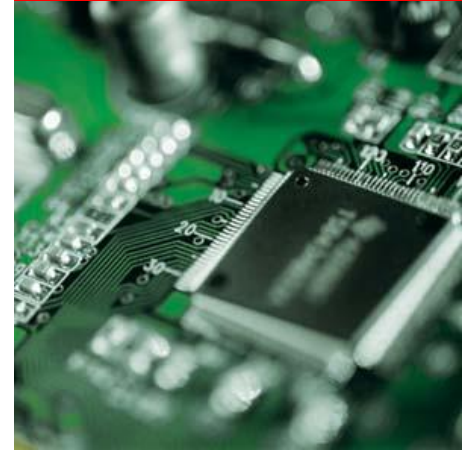
次のスキーマと表から仮想ワークロードを作成
表にディメンションまたは主キー/外部キー制約が含まれていれば、アドバイザーにより仮想のワークロードが作成されます。スキーマと表

追加

カンマ区切りリスト

ヒント そのスキーマに属するすべての表を指定する場合は、スキーマ名を入力します。

ワークロードを指して
アドバイザーを実行

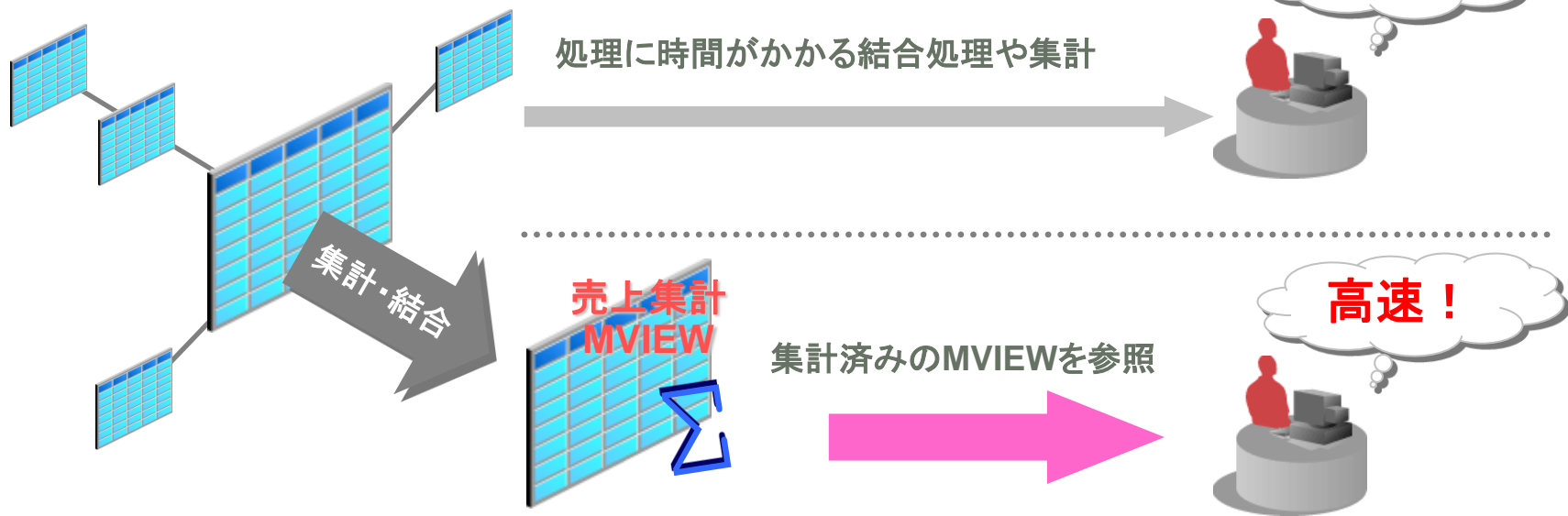


【DWH向けのデータベース・テクノロジー】 マテリアライズド・ビュー

マテリアライズド・ビュー

概要

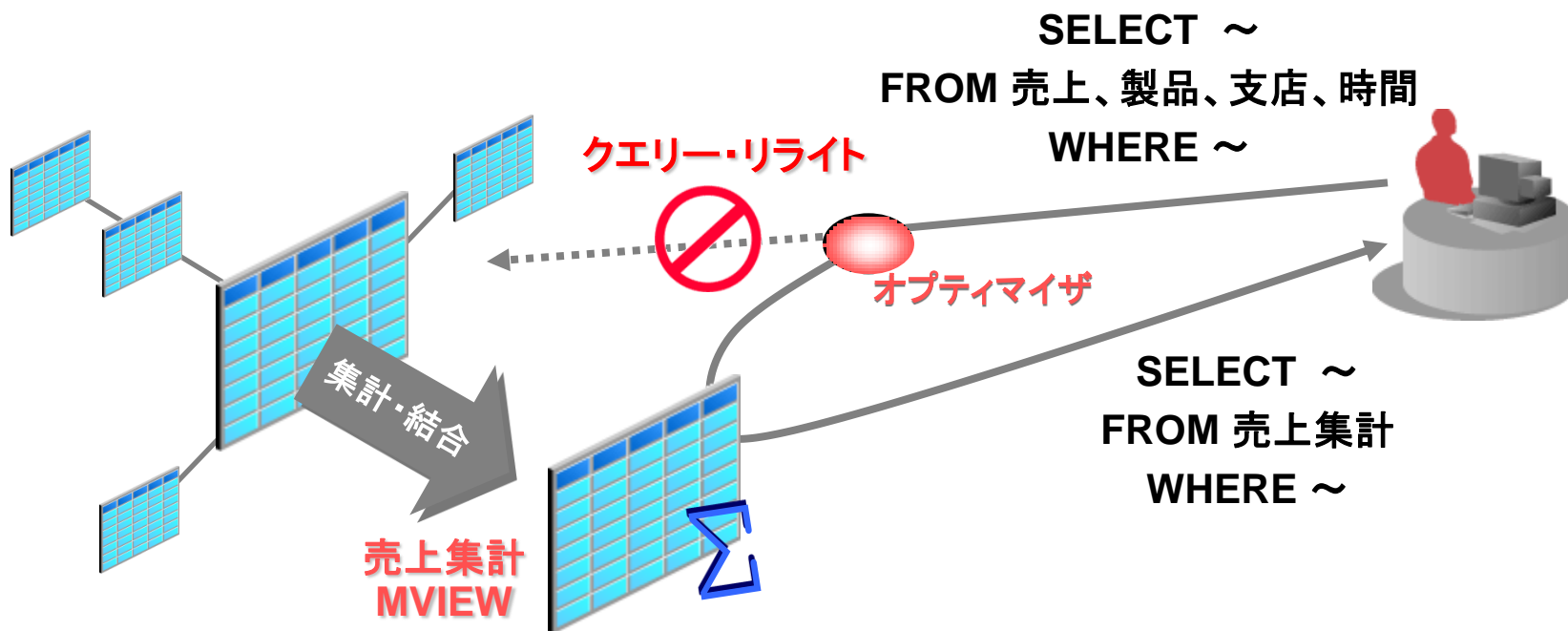
- SQL実行の度に、大量データの集計や結合を実施するのは効率が悪い
 - H/Wリソースの無駄な消費、検索性能の劣化、...
- マテリアライズド・ビューを使用して、集計や結合を事前に実行した結果サマリーを保持しておけば解決！！



マテリアライズド・ビュー

アプリケーションから透過的に使用可能

- クエリー・リライト
 - アプリケーション側で実装されている実表に対するSQLを書き直す必要なく、自動的にマテリアライズド・ビューを使用する機能



マテリアライズド・ビュー

リアルタイムDWHへの対応

- マテリアライズド・ビューは元表の変更(データロードなど)を反映させるためにリフレッシュを行う必要
- マテリアライズド・ビュー作成時に設定可能なリフレッシュ・モード
 - ON DEMAND: 手動
 - DBMS_MVIEWパッケージのプロシージャを用いた手動リフレッシュ
 - ON COMMIT: 自動
 - コミット時に自動リフレッシュ

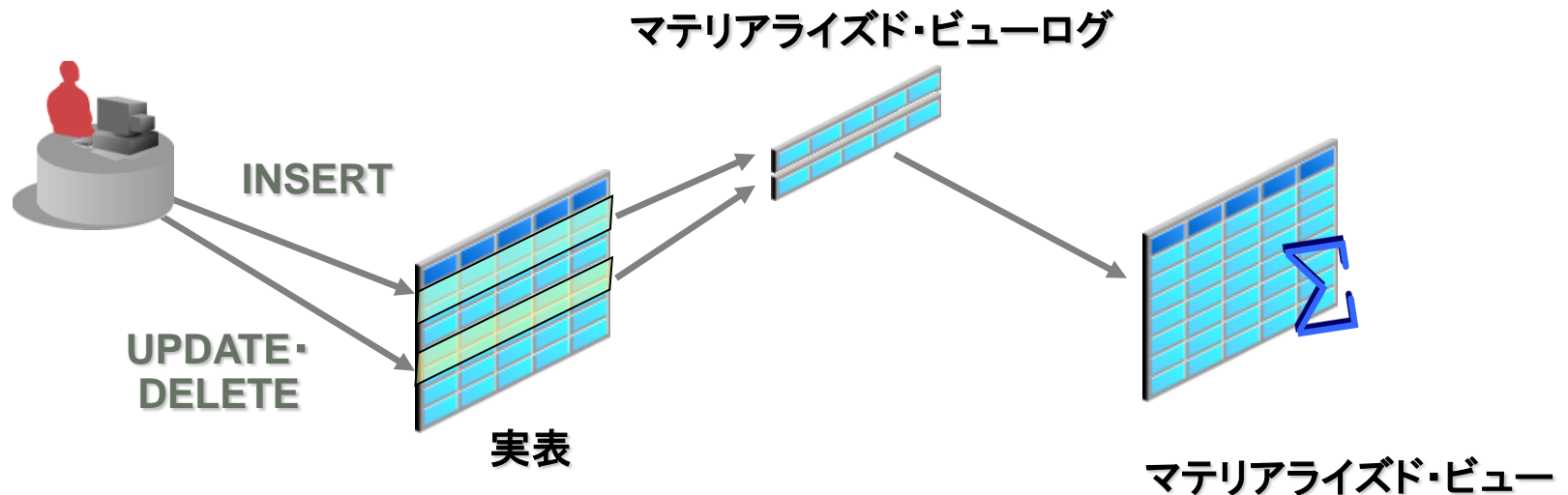
リフレッシュ・オプション	説明
Complete (Always)	MVデータを全て再計算
Fast	高速リフレッシュ、差分のみ適用
Force	高速リフレッシュ、できなければ再計算
Never	リフレッシュしない



マテリアライズド・ビュー

高速リフレッシュ

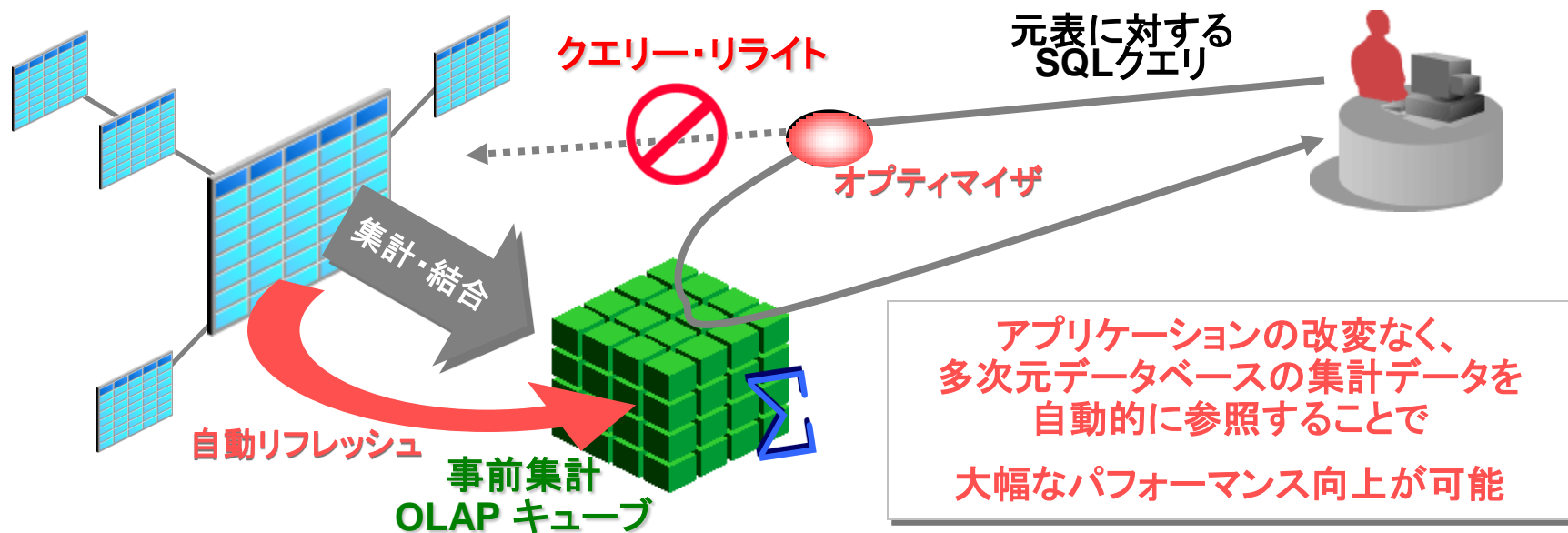
- 実表に発生した更新データのみ(差分)の適用で高速にリフレッシュ
- リフレッシュ(同期)の時間は元表のサイズではなく、変更データ量に依存
 - 変更情報を管理する為、マテリアライズド・ビュー・ログの作成が必要



キューブ・マテリアライズド・ビュー

Oracle Database 11g~

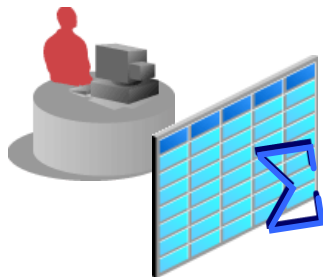
- Oracle OLAPキューブをマテリアライズド・ビューとして利用可能
 - 全ての組み合わせが事前集計済み
 - 通常のSQLアクセスがオプティマイザによりクエリー・リライト
 - 高速リフレッシュ可能
 - 管理対象削減、Disk容量削減 => 大幅なコスト削減



マテリアライズド・ビュー 作成例

- 高速リフレッシュ、クエリ・リライトを有効に設定した例

```
CREATE MATERIALIZED VIEW product_sales_mv
  BUILD IMMEDIATE
  REFRESH FAST
  ENABLE QUERY REWRITE
  AS SELECT p.prod_name, SUM(s.amount_sold) AS dollar_sales,
           COUNT(*) AS cnt, COUNT(s.amount_sold) AS cnt_amt
  FROM   sales s, products p
  WHERE  s.prod_id = p.prod_id
  GROUP BY p.prod_name;
```



• オプション

- 作成タイミングの指定
 - **BUILD DEFERRED**: Completeリフレッシュまでデータ投入されない
 - **BUILD IMMEDIATE** (デフォルト値): 作成と同時にデータ投入
- リフレッシュ・オプションの指定
 - **REFRESH FAST ON COMMIT** など
- クエリ・リライトの有効化
 - **ENABLE QUERY REWRITE** を指定

といっても、実際にどのようなMViewを作成すれば良いかわからない。。。
そんな時は、**EMのSQL Access Advisor**

ORACLE Enterprise Manager 11g Database Control

Cluster Database: orcl > Logged In As SYS

Advisor Central

Advisors Checkers View Data Real Time: 15 Second Refresh

Advisors

- ADDM
- MTR Advisor
- SQL Performance Analyzer
- Automatic Undo Management
- Segment Advisor
- Streams Performance Advisor
- SQL Advisors**

SQL Advisors

The SQL Advisors address several important use cases having to do with SQL: identify physical structures optimizing a SQL workload, tune individual statements with heavy execution plans, identify and correct result set divergence, build test cases for failed SQL.

SQL Access Advisor

SQL Access Advisor Evaluate an entire workload of SQL and recommend indexes, partitioning, materialized views that will improve the collective performance of the SQL workload.

SQL Access Advisor: Initial Options

Verify use of access structures (indexes, materialized views, partitioning, etc) only

Recommend new access structures

Inherit Options from a previously saved Task or Template

Overview

The SQL Access Advisor evaluates SQL statements in a workload Source, and can suggest indexes, partitioning, materialized views and materialized view logs that will

ORACLE Enterprise Manager 11g Database Control

Workload Source Recommendation Options Schedule Review

SQL Access Advisor: Recommendation Options

Database orcl_orcl_1 Logged In As SYS

Access Structures to Recommend

- Indexes
- Materialized Views
- Partitioning

Task Name: SQLACCESS6832566
 Status: COMPLETED
 Advisor Mode: LIMITED
 Scheduler Job: ADV_SQLACCESS6832566

Started: Nov 19, 2010 8:38:00 PM JST
 Ended: Nov 19, 2010 8:41:49 PM JST

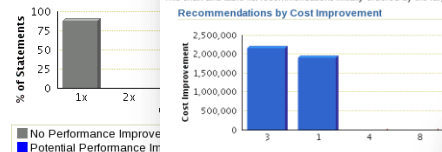
Summary Recommendations SQL Statements Details

Potential for Improvement

Workload I/O Cost



Query Execution Time



Recommendations
 Recommendations: 8
 Space Requirements (MB): 0.023
 User Specified Space Adjustment: Unlimited
 Hide Recommendation Action Counts

Indexes	: Create	0	Drop	0	Retain	5
Materialized Views	: Create	3	Drop	0	Retain	0
Materialized View Logs	: Create	3	Retain	0	Alter	0
Partitions	: Tables	0	Indexes	0	Materialized Views	0

SQL Statements
 Select Recommendations for Implementation
 Include Retain Actions
 Recommendation Details Schedule Implementation Show SQL

Select	Implementation	Object Types	Cost
<input checked="" type="checkbox"/>	3	3	
<input checked="" type="checkbox"/>	1	3	
<input checked="" type="checkbox"/>	4	3	

ORACLE Enterprise Manager 11g Database Control
 Setup Preferences Help Logout
 Cluster Database
 Advisor Central > Results for Task: SQLACCESS6832566
 Logged in As SYS

Task Name: SQLACCESS6832566
 Status: COMPLETED
 Advisor Mode: LIMITED
 Scheduler Job: ADV_SQLACCESS6832566
 Publish Point: 2

Started: Nov 19, 2010 8:38:00 PM JST
 Ended: Nov 19, 2010 8:41:49 PM JST
 Running Time (seconds): 229
 Total Time Limit (minutes): 10000

Summary Recommendations SQL Statements Details

Recommendations by Cost Improvement

ORACLE Enterprise Manager 11g Database Control
 Setup Preferences Help Logout
 Cluster Database
 Advisor Central > Results for Task: SQLACCESS6832566 >
 Logged in As SYS

Recommendation: 3
 SQL Access Advisor generates default object names and uses the default schemas and tablespaces specified during task creation, but you can change them. If you edit any name, dependent names, which are shown as readily, will be updated accordingly. If the Tablespace field is left blank the default tablespace of the schema will be used. When you click OK, the SQL script is modified, but it is not actually executed until you select 'Schedule Implementation' on the Recommendations or SQL Statements pages.

Actions
 Set Tablespace for All Actions: [] [Go]

Implementation Status	Action	Object Name	Object Attributes	Base Table	Schema	Tablespace
<input checked="" type="checkbox"/>	CREATE MATERIALIZED VIEW LOG			SH.SALES_DEMO32	SH	
<input checked="" type="checkbox"/>	CREATE MATERIALIZED VIEW	MV\$\$_01150001	General Match		SH	
<input checked="" type="checkbox"/>	GATHER TABLE STATISTICS	MV\$\$_01150001			SYS	

SQL Affected by Recommendation: 3

Statement ID	Statement	Original Cost	New Cost	Cost Improvement	Cost Improvement (%)	Execution Count
2716	select count(*) from sales_demo32	2148408	6	2148402	100.00	2

ORACLE Enterprise Manager 11g Database Control
 Setup Preferences Help Logout
 Cluster Database
 Advisor Central > Results for Task: SQLACCESS6832566 > Recommendation: 3 >
 Logged in As SYS

Show SQL

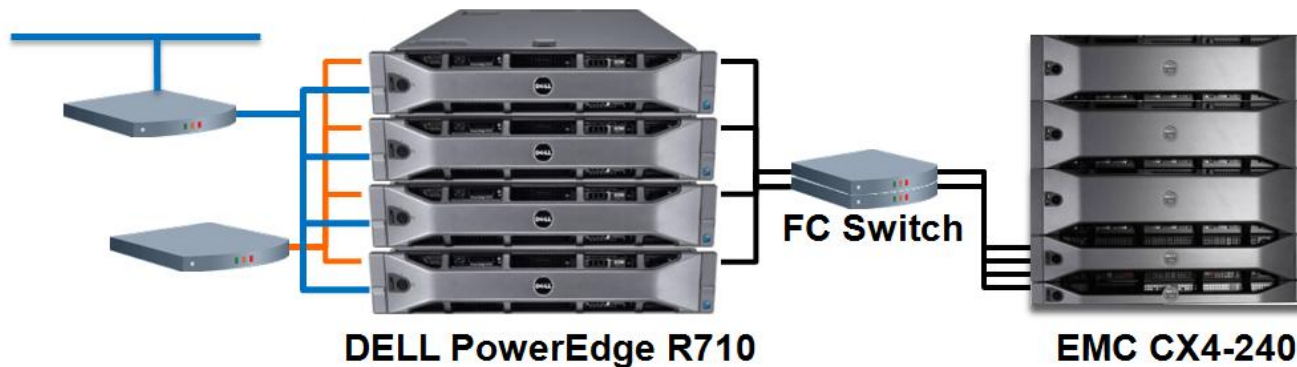
CREATE MATERIALIZED VIEW "SYS"."MV\$\$_01150001"
 REFRESH FAST WITH ROWID
 ENABLE QUERY REWRITE
 AS SELECT COUNT(*) M1 FROM SH.SALES_DEMO32;

Done

Oracle **GRID** Center - **DeMo**

Oracle GRID Center - DeMo

1. 自動パラレル度設定による検索性能の向上
2. Enterprise Managerによるマテリアライズド・ビューのアドバイス



DB Server (1 node spec)	DELL PowerEdge R710 CPU: Intel Xeon E5540(4 Core) x 2 Memory: 36GB
Storage	EMC CX4-240 HDD: FC 15krpm 146GB x 10



本セッションのまとめ

まとめ

DWHの性能向上を実現するデータベース・テクノロジー

上位アプリケーションに**透過的** Enterprise Managerで**容易**に最適設定

- SQLの自動パラレル度設定
 - H/Wリソースを効果的に活用した性能向上
 - SQL Monitoringによるパラレル実行の確認
- データ圧縮
 - ストレージI/Oボトルネックの改善による性能向上
 - Compression Advisorによる圧縮率の事前見積もり
- パーティショニング
 - アクセス・データを限定することによる性能向上
 - Partitioning Advisorによるワークロードに最適なパーティション化
- マテリアライズド・ビュー
 - 事前に集計・結合済みデータを保持することによる性能向上
 - SQL Access Advisorによるワークロードに最適なマテリアライズド・ビューの作成



Hardware and Software Engineered to Work Together

ORACLE

ORACLE®