Oracle Database Innovation 革新し続ける、
クラウドに最適化されたデータベース
～ データベースクラウドを支える最新テクノロジの全貌 ～

ORACLE
CLOUD PLATFORM

POCO
The Power of Cloud by Oracle

ビジネスの変化に適応する次世代データベース基盤
～ Oracle Database In-Memoryの詳細とその可能性 ～

2015年10月16日
日本オラクル株式会社
丹羽　勝久

ORACLE

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。

本日お伝えしたい内容

1. Oracle Database In-Memory技術概要
2. 事例紹介
3. Oracle Database In-Memoryの応用構成

本日お伝えしたい内容

1. Oracle Database In-Memory技術概要
2. 事例紹介
3. Oracle Database In-Memoryの応用構成

爆発的なデジタル情報量の増加に向けた次世代プラットフォーム

情報量

90%
のデータは2年以内に作成

50倍
2020年までの情報量拡大

インターネット接続端末
2012年
90億台
2020年
500億台

洗練された顧客

体験を差別化

“Engage me Everywhere”
“Wow Me”
“Know Me”
“Meet My Expectations”
“Understand and Reward Me”

ソーシャル

26%
嫌な対応を受けた経験をつぶやく

86%
取引を中止する

94%
より良いサービスには対価を払う

モバイル端末

60億人
モバイル端末契約者数

世界人口の
87%

モバイルデータCAGR成長率
78%

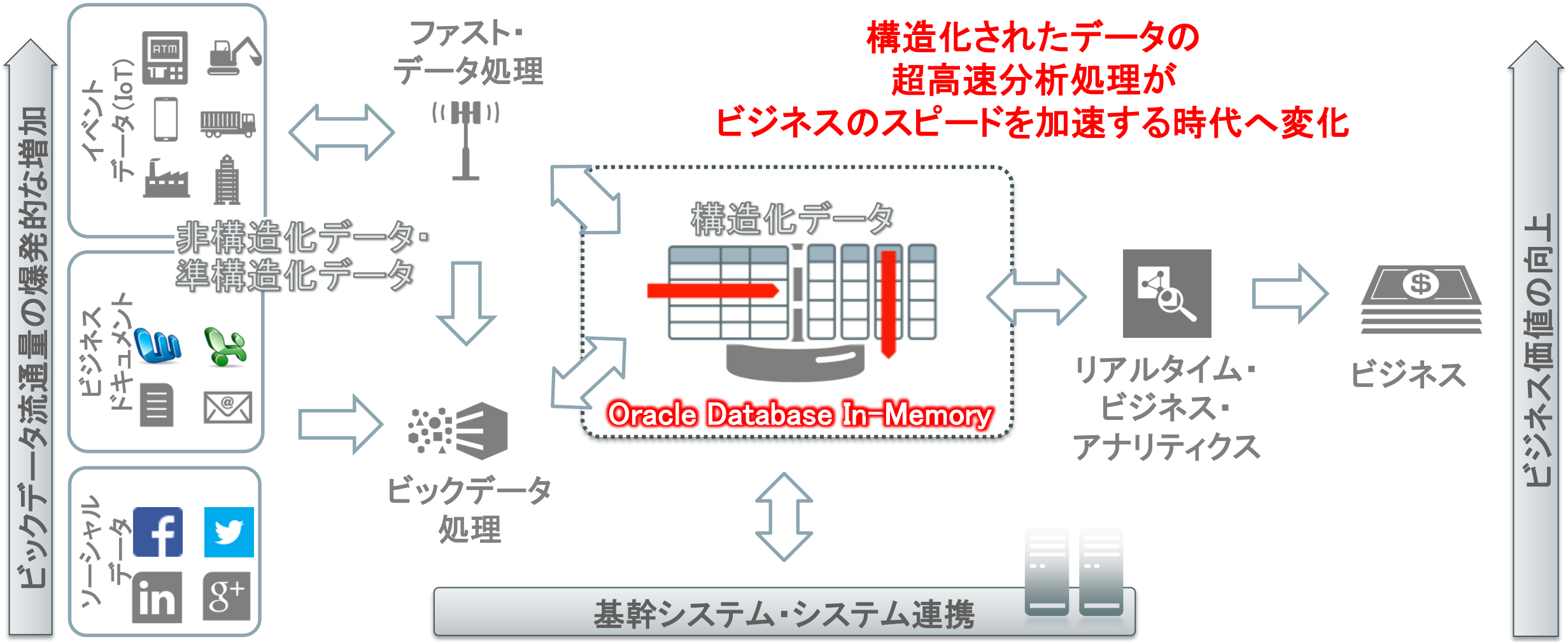
企業情報システム

膨大な
20年前の
レガシーシステム

新たな
クラウドシステム

Sources: United Nations / International Telecommunications Union, internetworldstats.com, IDC/EMC 2011 Digital Universe Study, 2010 Digital Universe Decade Study, Data rEvolution Sept 2011, CSC's Leading Edge Forum Portio Research Mobile Factbook 2012, Facebook Director of Global SMB Markets Dan Levy, BIA/Kelsey’s Interactive Local Media West Conference, IDC: “Time for Change: Optimizing Datacenter Infrastructure with Technology Refresh”

大量データ分析の中核となるインメモリ処理
増加するデータ流通量をビジネス価値の向上へリアルタイムに反映



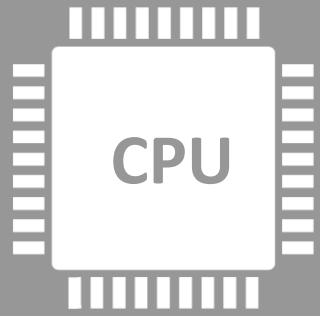
Oracle Database In-Memoryの開発ゴール

リアルタイム
アナリティクス

OLTPの
高速化

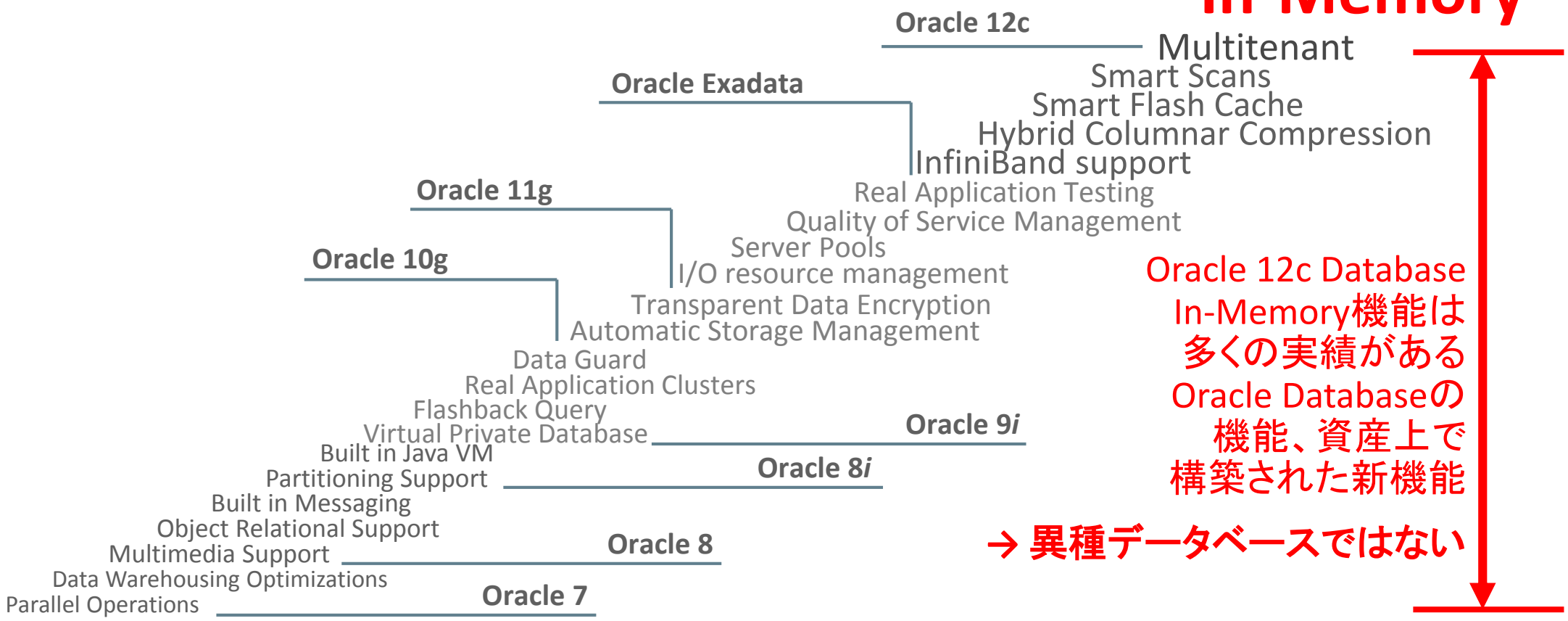
アプリケーション
の変更なし

最新世代の
H/Wを有効活用



Oracle Databaseを利用して頂いている多くのアプリケーション資産を最大限活かしたうえで、新たな技術革新であるインメモリ・データベースのメリットを享受して頂くことを可能とします。

Oracle Database In-Memoryに至る歴史



業界標準データベースとして歴史を重ねたOracle Databaseの一機能としてインメモリ機能を実装しました。
過去から積み上げられた各種Oracle Databaseの機能や資産を活かしながら利用可能な利便性の高い機能です。

2種類の主要なデータベース・フォーマット

データベースの主要なデータフォーマットとして、ロー型とカラム型が存在します。

ロー型

OLTP処理の高速化に適しています。

例：注文データの挿入と検索
少数の行(ロー)と多数の列(カラム)を高速処理

カラム型

集計、分析処理の高速化に適しています。

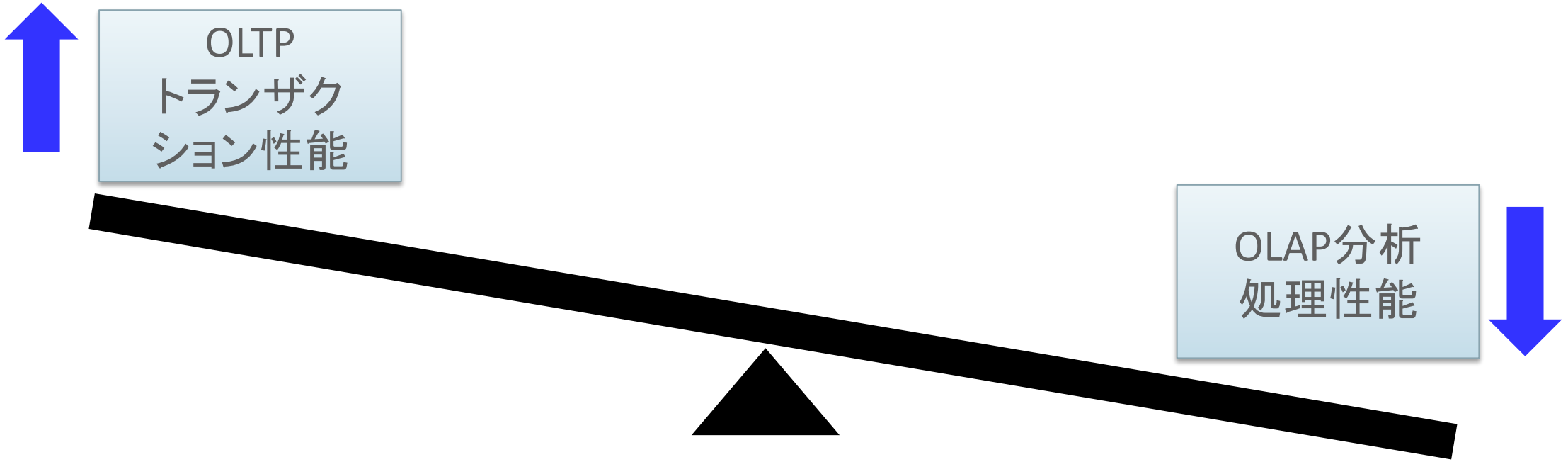
例：都道府県毎の売上合計のレポート
少数の列(カラム)と多数の行(ロー)を高速処理

一般的なデータベース論として、どちらかが優れたフォーマット、という訳ではなく、各々の特徴を活かし利用用途に応じた使い分けが必要です。Oracle Database In-Memoryは2種類のフォーマットをメモリ上で実現します。

OLTPとOLAPの性能向上はトレードオフ

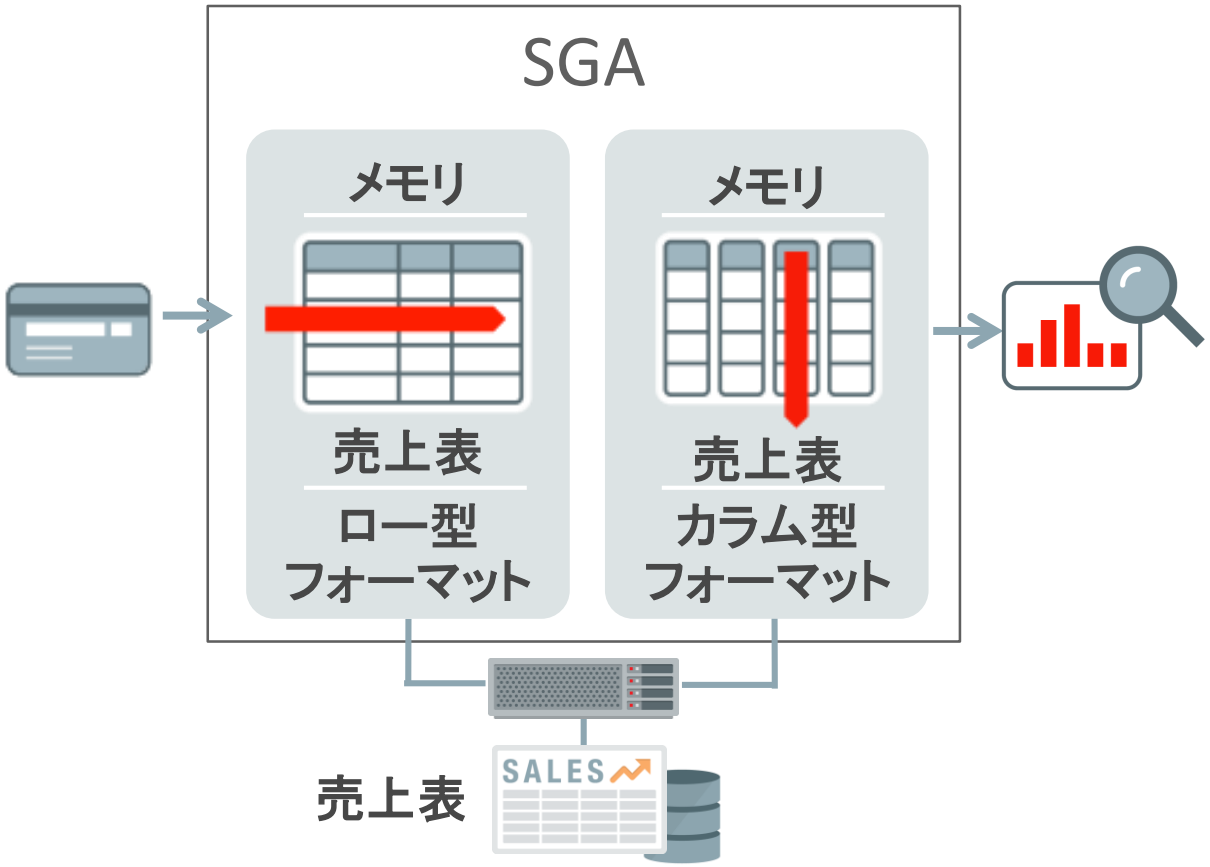
どちらかを性能向上するとどちらかにオーバーヘッドが発生

OLTPとOLAPを1つのデータベースで
共存することは難しい



インメモリ・デュアル・フォーマット

同一データをメモリ上で2種類のフォーマットを保持し、全ての処理を高速化させます。



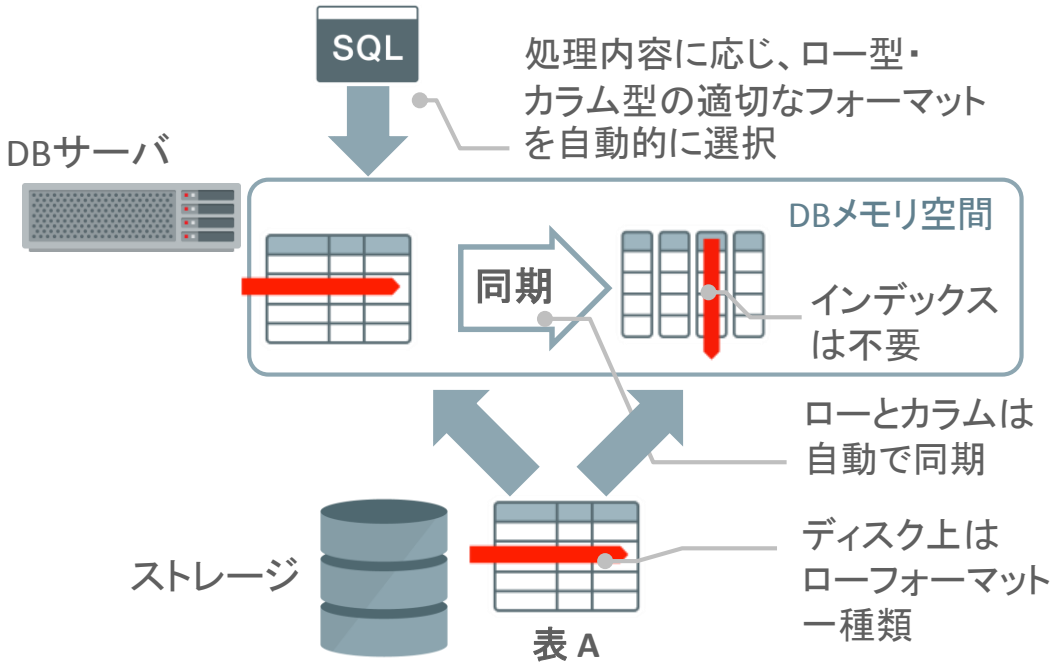
- データベースのメモリ空間内で、同一データをロー型、カラム型双方のデータ・フォーマットで保有します。
- 双方のデータ・フォーマットを使い分けることで、全てのデータベース処理の高速化を実現します。
- 処理の特性をSQL文からOracle Databaseが判断し、適切なメモリ上のデータフォーマットに対して処理を実行します。
  - 分析、集計処理はカラム型フォーマットに対して実行されます。
  - OLTP処理はロー型フォーマットに対して実行されます。
- トランザクションの一貫性は保証されます。

Oracle Database In-Memoryのメモリ利用方法

OLTPと分析処理をリアルタイムに融合した唯一のインメモリ・データベース

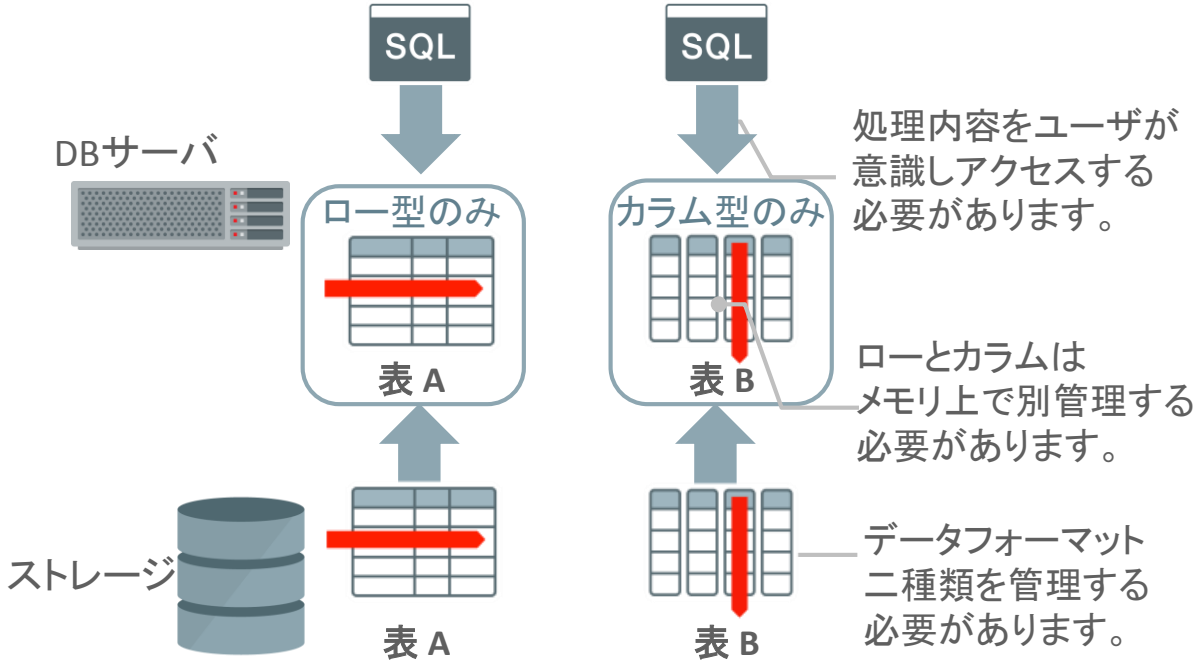
Oracle Database In-Memory

ロー(行)型とカラム型を**両方同時に**実現



一般的なインメモリ・データベースの仕組み

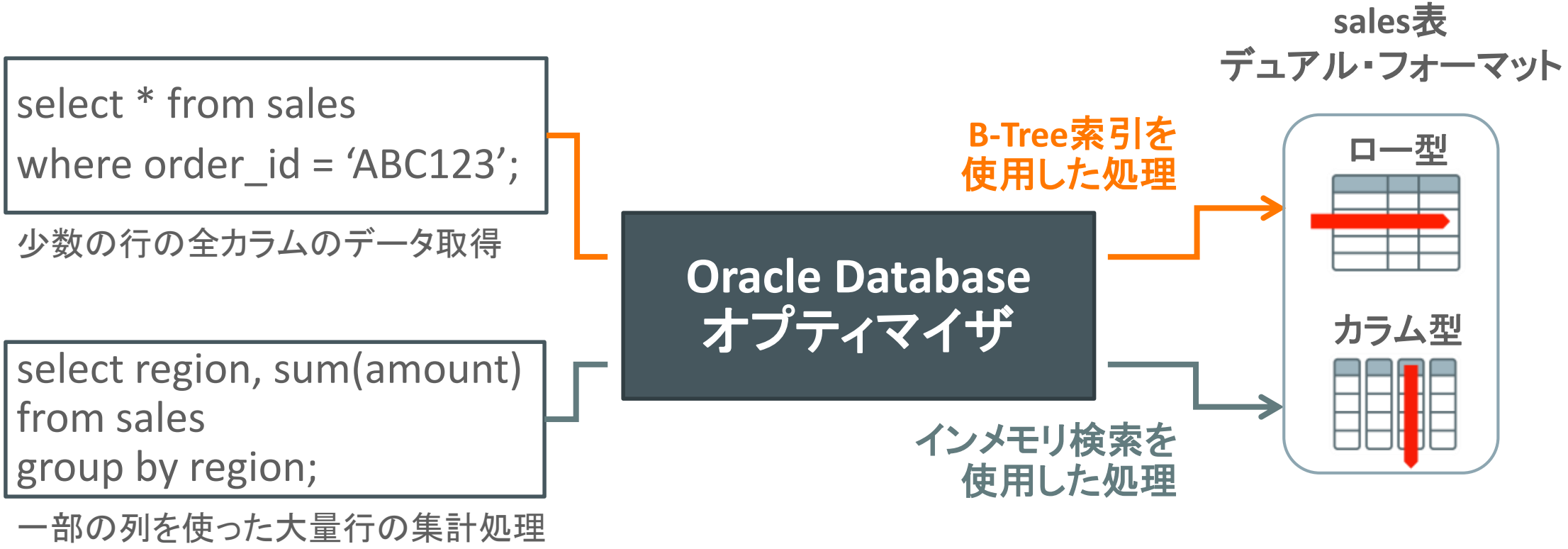
ロー(行)型とカラム型の**どちらか一方を**選択



アプリケーション設計の段階で利用するフォーマットを決める必要がある他社DBと異なり、両方を同時に保持できるOracle Databaseでは、**OLTPで発生したデータに対してリアルタイムにDWH検索を行うことも可能となります**。

デュアル・フォーマットの透過的/効果的な活用

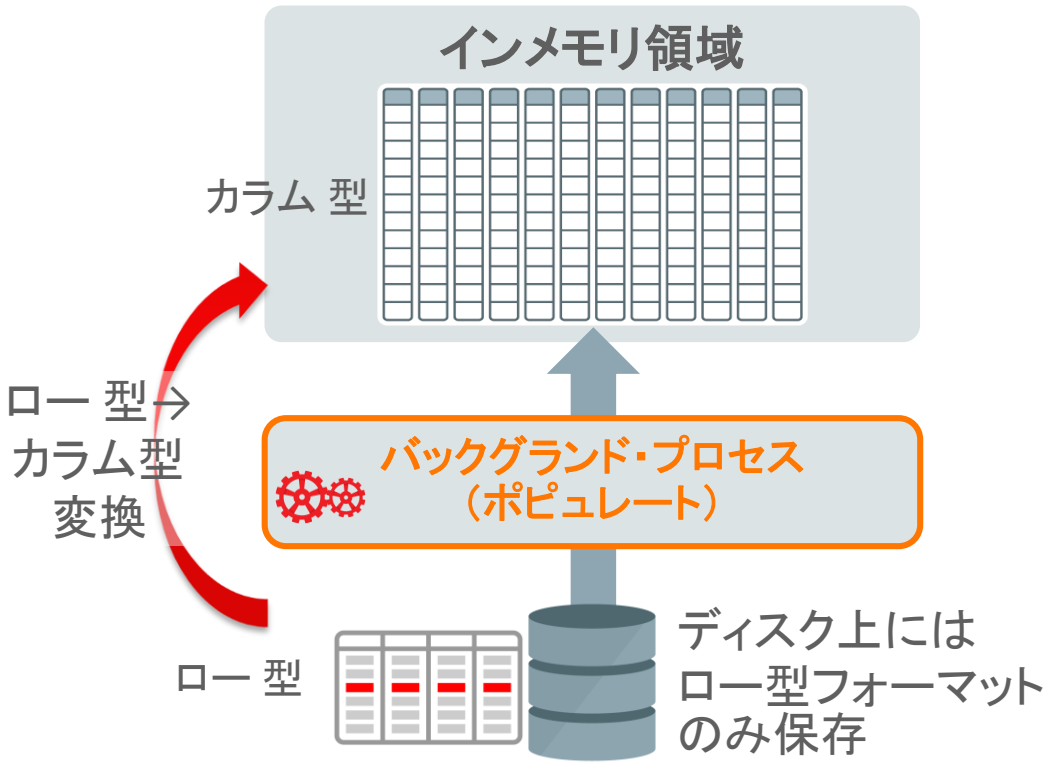
データベースのオプティマイザがSQLにあわせて最適なフォーマットを選択



テーブル毎に処理するデータフォーマットが決まるのではなく、SQL毎に処理するデータフォーマットを変えることが可能であるため、同一の表をOLTP処理にも、DWH処理にも利用することが可能です。

インメモリデータ配置中も処理を継続可能

新たなバックグランド・プロセスがメモリー上に配置(ポピュレート)します。

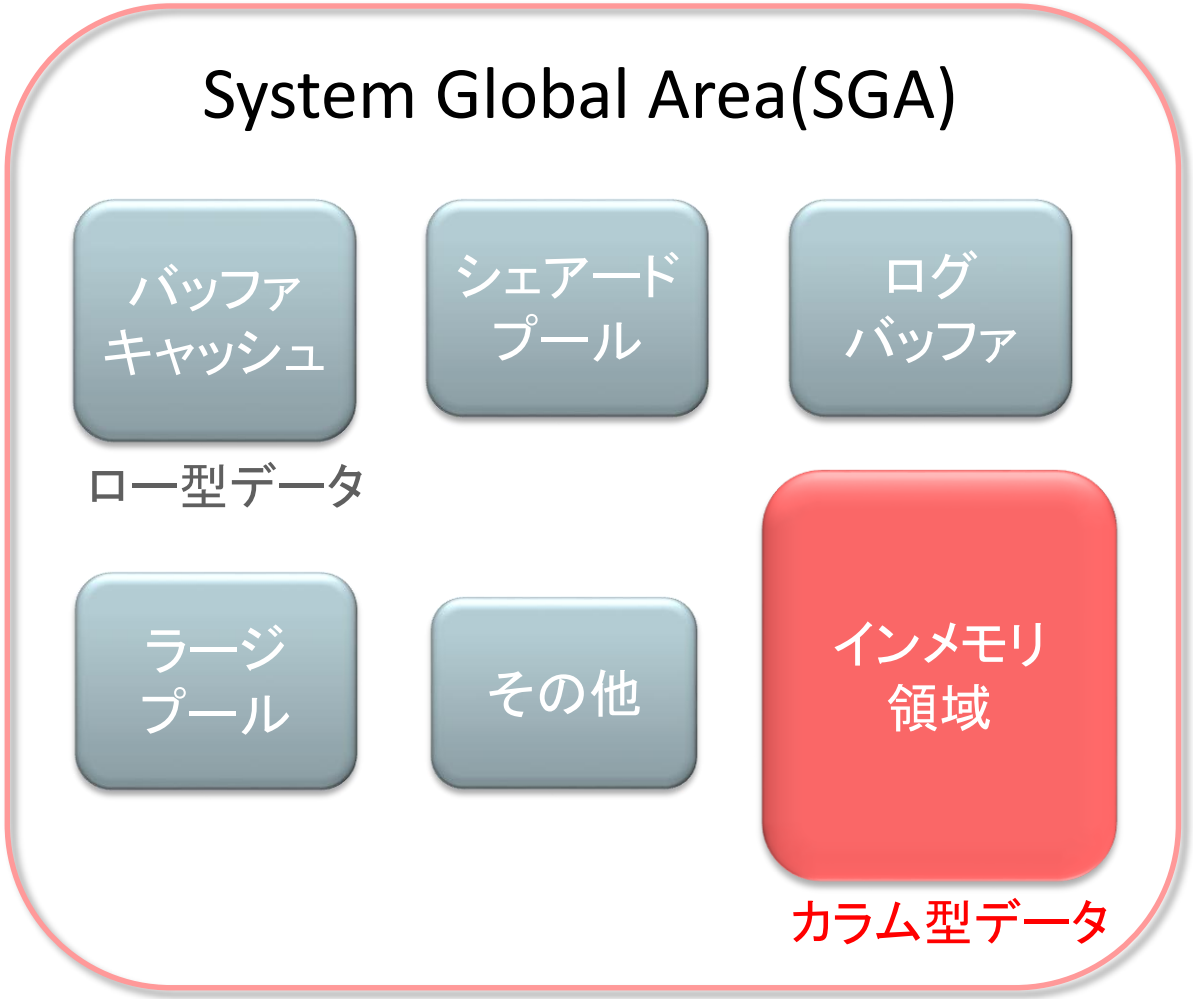


- パラメータでプロセス数を設定可能です。
- ポピュレート中もアクセス可能です。
- テーブル毎に優先度を設定することが可能です。

優先レベル	ロードタイミング	Note
CRITICAL	データベース起動後	一番優先度が高い
HIGH	データベース起動後	CRITICALの次の優先度
MEDIUM	データベース起動後	HIGHの次の優先度
LOW	データベース起動後	MEDIUMの次の優先度
NONE	初回オブジェクトアクセス時	優先度指定がない場合のデフォルト値

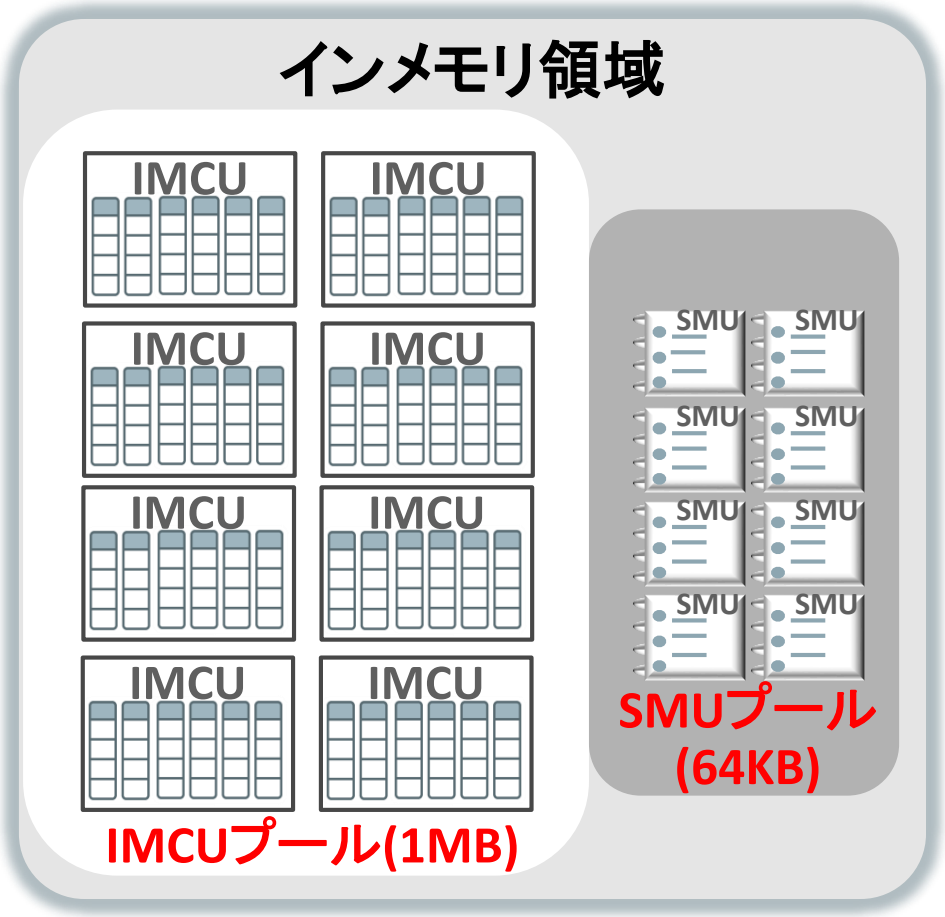
メモリ上にカラム型データのポピュレートが完了するまで、ロー型データを利用し処理が可能です。これによりメモリへの展開(配置)を待つことなく素早く業務処理を開始する事が可能です。結果としてダウンタイムを最小化します。

インメモリ領域: SGA内の新しい領域



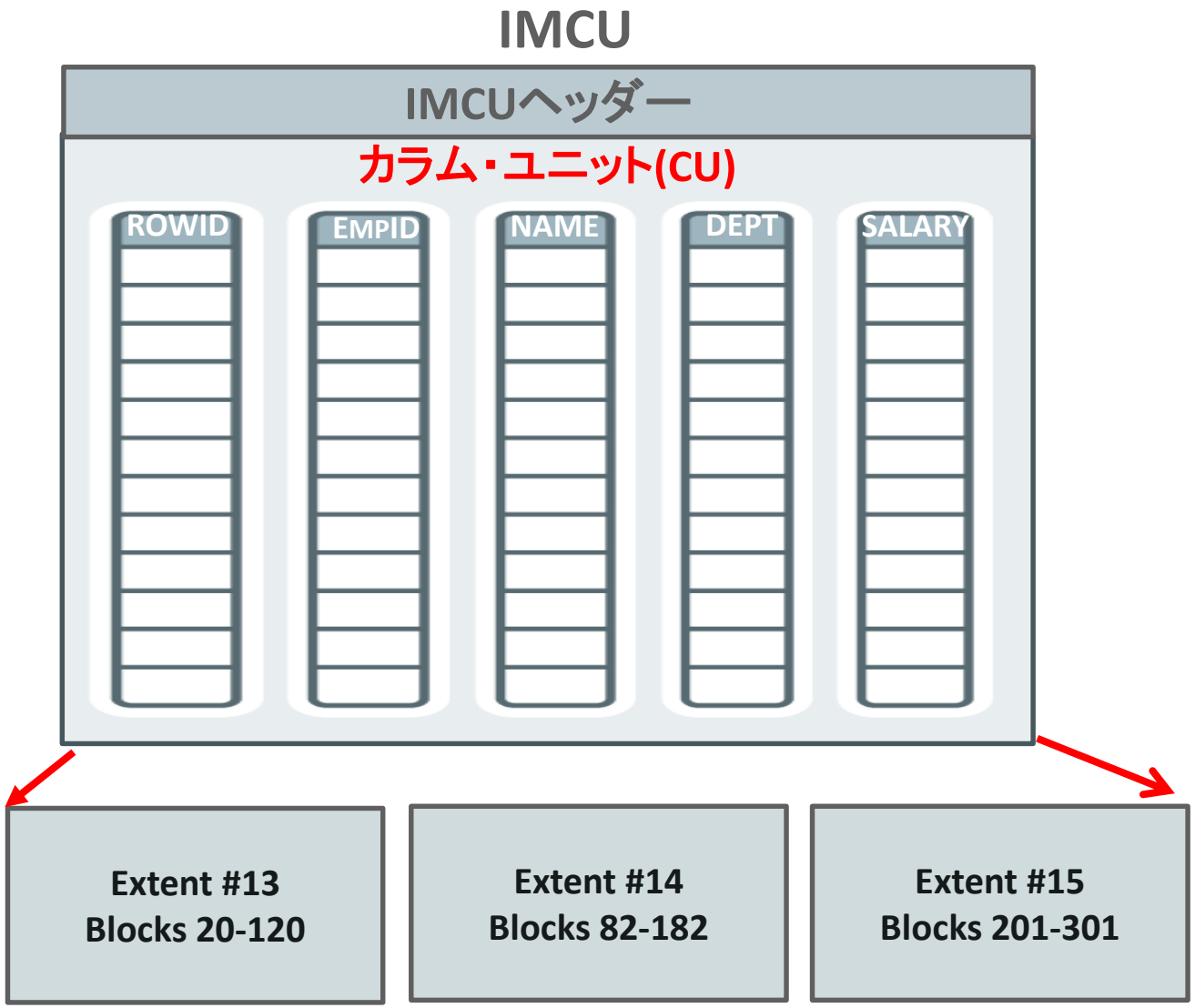
- 新しいインメモリ・カラム・フォーマットのデータを格納
- 初期化パラメータ「INMEMORY_SIZE」により設定
  - 最小サイズ: 100MB
- SGA_TARGETはこのインメモリ領域を格納できる十分大きな値の設定が必要
- 静的な領域で自動メモリー管理により増減しない

インメモリ領域: 構成



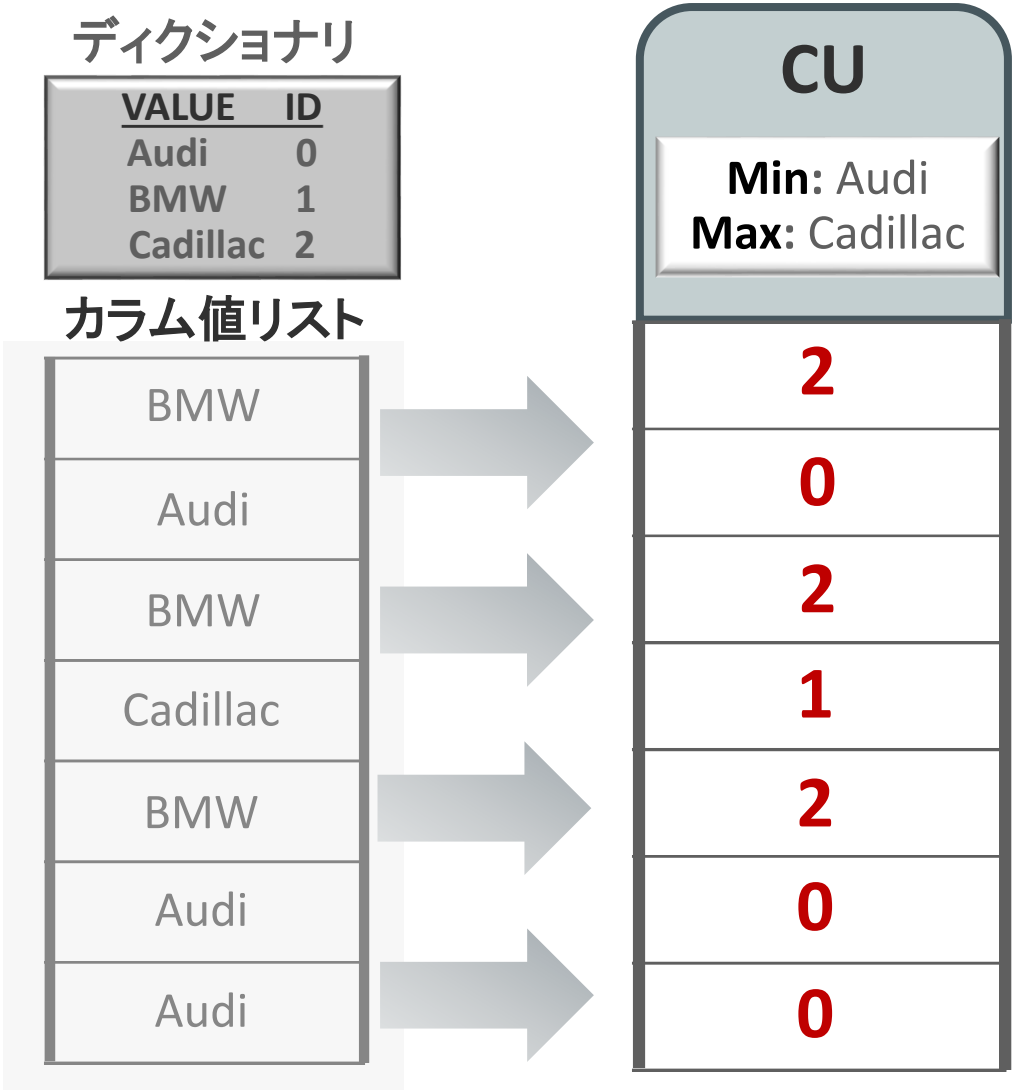
- 2つのサブプールで構成:
  - IMCU(1MB)プール:
    IMCU(In-Memory Compression Units)を格納
  - SMU(64KB)プール:
    SMU(Snapshot Metadata Units)を格納
- IMCUはカラム書式のデータを格納
- SMUはメタデータとトランザクション情報を格納

IMCU： In-Memory Compression Unit



- カラム型オブジェクトの管理単位
  - カラム型データをある程度の行数のセットで保持（例：50万行程度）
  - 格納される行は1つ以上の表エクステントから取得
- IMCUの実サイズは行サイズ、圧縮率等により変化（固定値ではない）
- カラム毎に分離／近接したカラム・ユニット(CU)として保存
  - Rowidも同様に1つのCUとして保存

CU：カラムユニット (Column Unit)

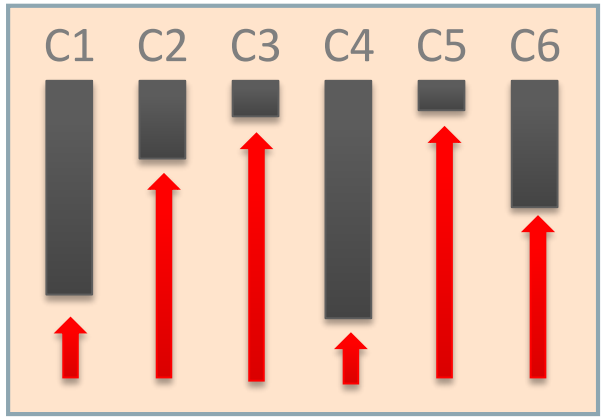


- IMCUに格納されている各カラム値の管理単位
- 全CUは自動的に最小／最大値を保存
  - インメモリ・ストレージ索引
- 圧縮フォーマット
  - 例）ディクショナリ圧縮
    CUは実際の値ではなく、サイズの小さいディクショナリIDをデータ値として格納
    → 圧縮した状態で検索が可能
  - 他の圧縮方式と組合わせることも可能

分析クエリの高速化:
カラム型表は何故分析用クエリーが高速か?

ポイント1:
集計に必要なカラムのみアクセス

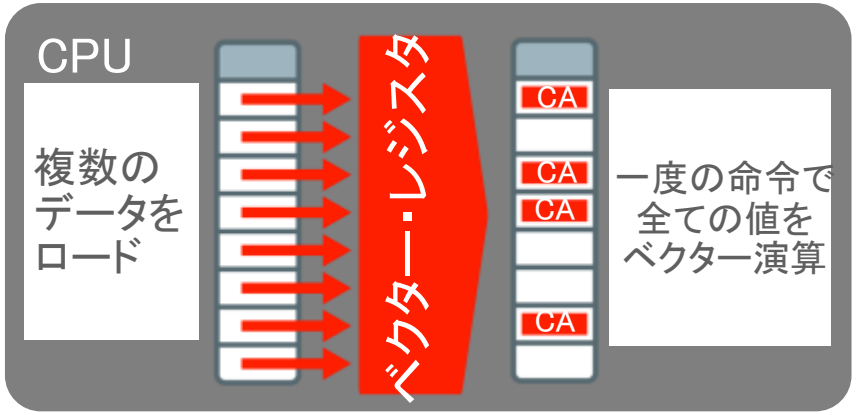
ポイント2:
効果的な圧縮技術により圧縮した状態で検索が可能 (ディクショナリ圧縮)



ポイント3:
インメモリ・ストレージ索引により最小限のIMCUのみスキャン

例) where storeid > 8

ポイント4:
最新のプロセッサで搭載されているSIMDにより高速スキャン

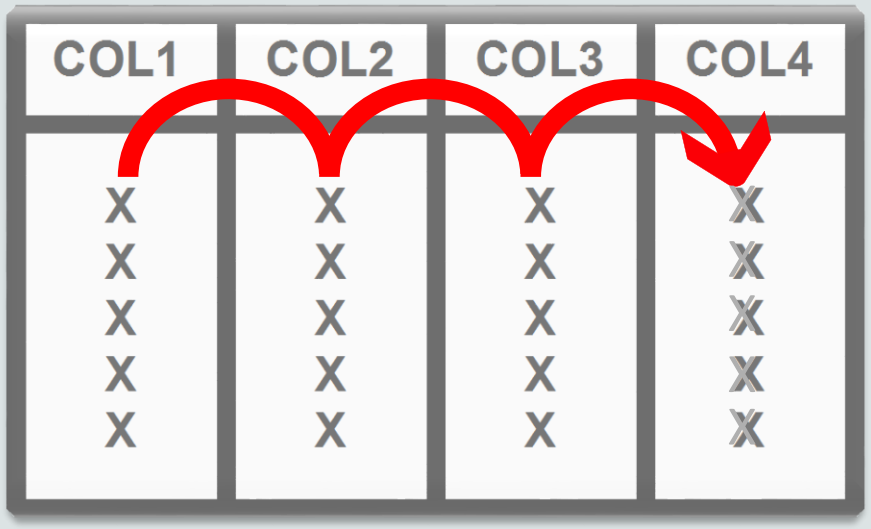


カラム型表は何故分析用クエリーが高速か?

ポイント1

必要なカラムのみアクセス

バッファ・キャッシュ



行フォーマット

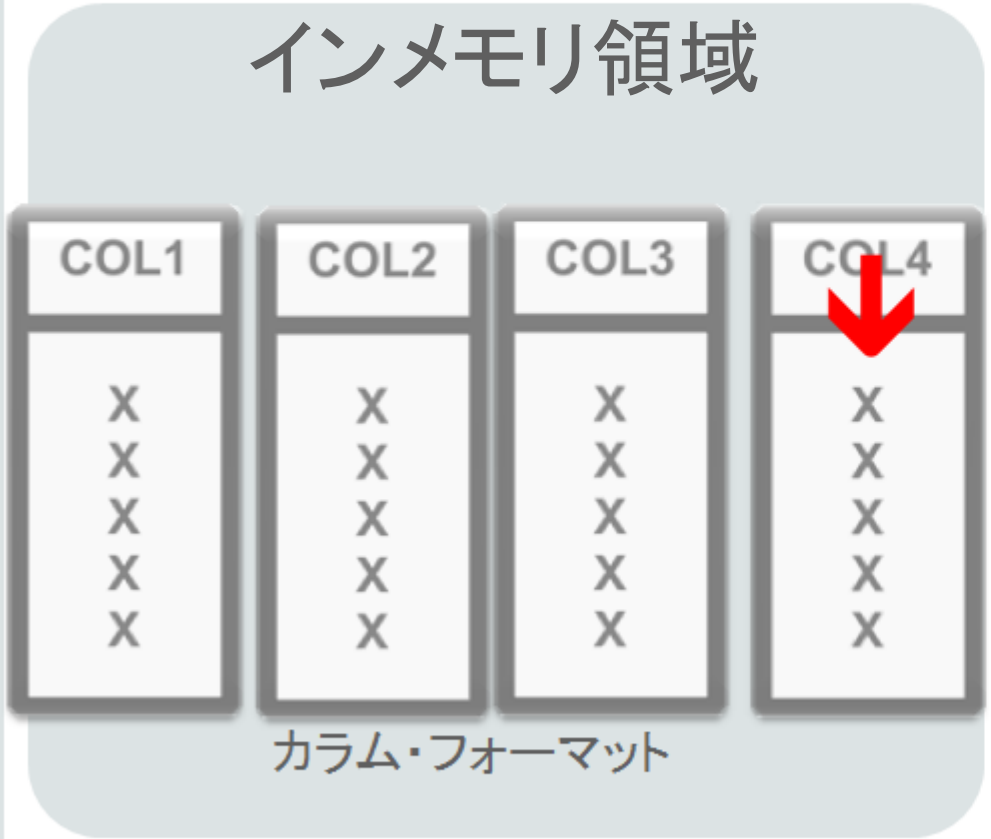
```
SELECT COL4 FROM MYTABLE;
```

X X X X X

カラム型表は何故分析用クエリーが高速か?

ポイント1

必要なカラムのみアクセス




```
SELECT COL4 FROM MYTABLE;
```

結果

X X X X X

必要なカラムのみアクセス

データの読込量少ない

カラム型表は何故分析用クエリーが高速か？

ポイント2

ディクショナリ圧縮

非圧縮

EMP表のJOB列

CLERK
SALESMAN
SALESMAN
MANAGER
SALESMAN
MANAGER
MANAGER
ANALYST
PRESIDENT
SALESMAN
CLERK
CLERK
ANALYST
CLERK

97 bytes

ディクショナリ圧縮

ディクショナリ(distinctされた値)

ソートされた値

カラム値	ディクショナリ値	ビット表現
ANALYST	0	000
CLERK	1	001
MANAGER	2	010
PRESIDENT	3	011
SALESMAN	4	100

カラム値サイズ合計＋ビット値合計
→ 36 bytes + 3bit * 5 = 38 bytes

エンコードされた各行の値 → 行

001	100	100	010	100	010	010	000	011	100	001	001	000	001

3bit * 14行 = 5.25bytes → 38 + 5.25 = **44 bytes (1/2.2 圧縮)**

ディクショナリ圧縮は圧縮した状態で検索が可能

Where job = ‘MANAGER’

Where job = 010
に内部的に変換

※圧縮状態で検索可能

カラム型表は何故分析用クエリーが高速か？

ポイント3

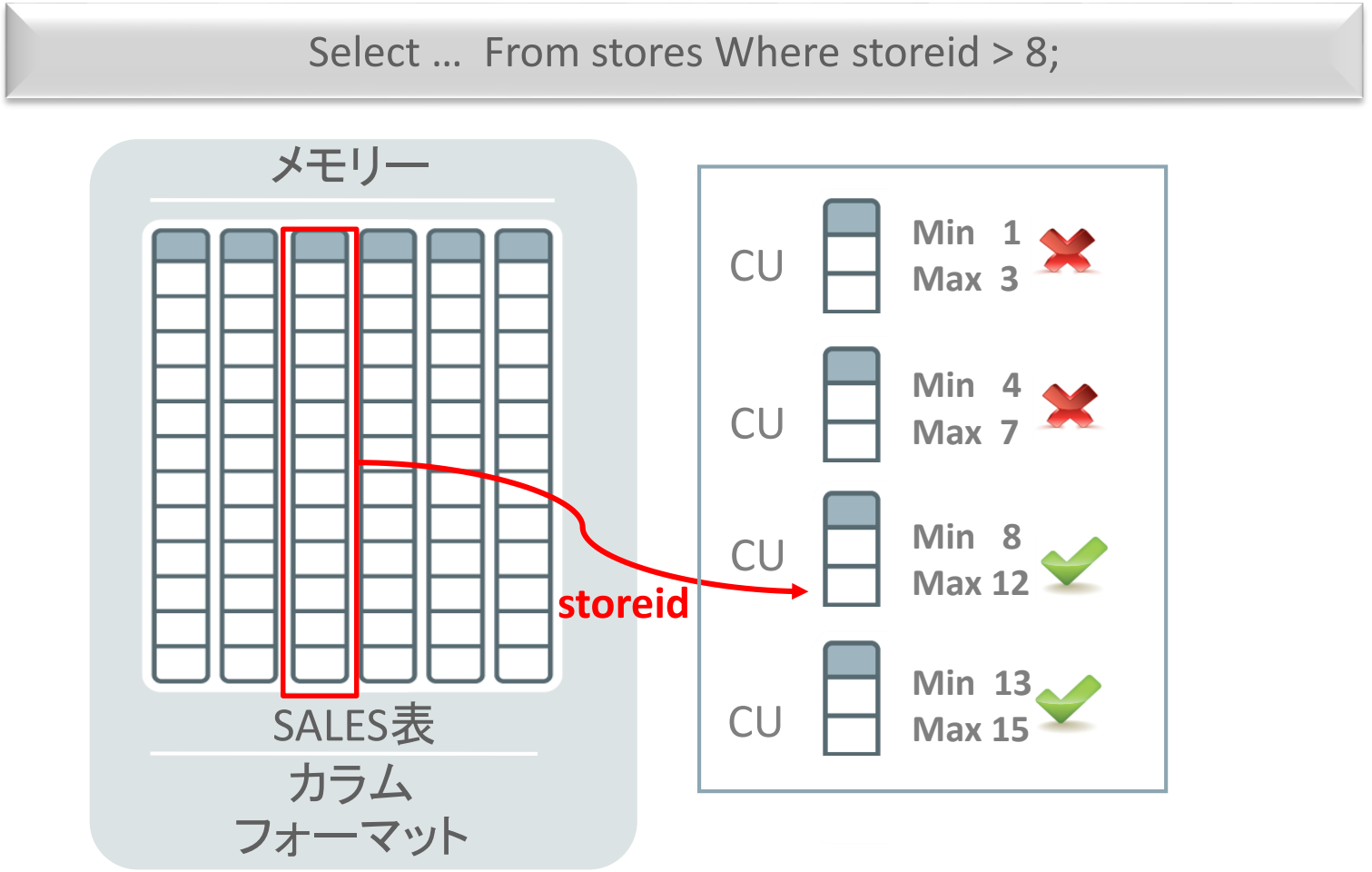
インメモリ・ストレージ索引 (※メモリー内に定義される)

各カラムは複数のカラム・ユニット(IMCU)で構成される

各IMCUで最小値/最大値を自動的に記録

WHERE句の条件に合致する領域だけを読み込み

すべての検索でパーティション・プルーニングと同様のパフォーマンスを提供



*1: IMCU - In-Memory Compression Unit

カラム型表は何故分析用クエリーが高速か？

ポイント4

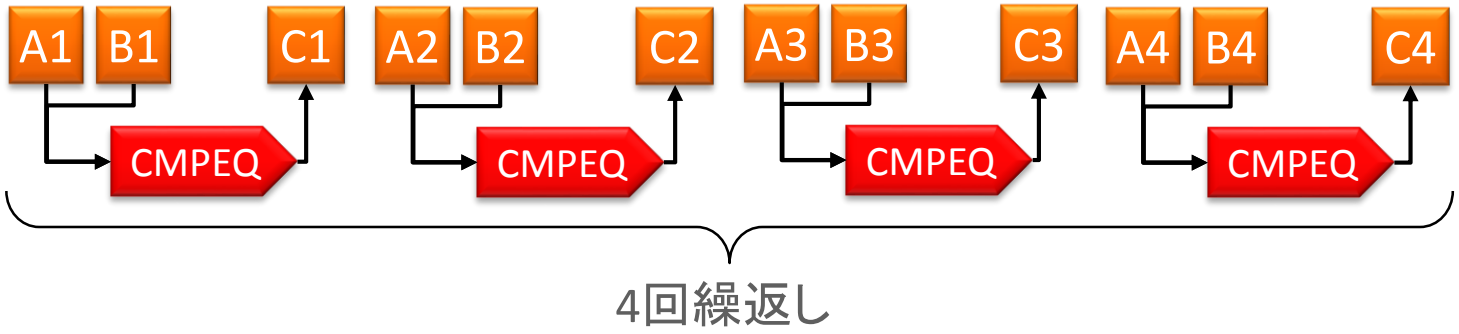
最新のプロセッサで搭載されているSIMD命令セットにより高速スキャン

SIMD：Single Instruction Multiple Data

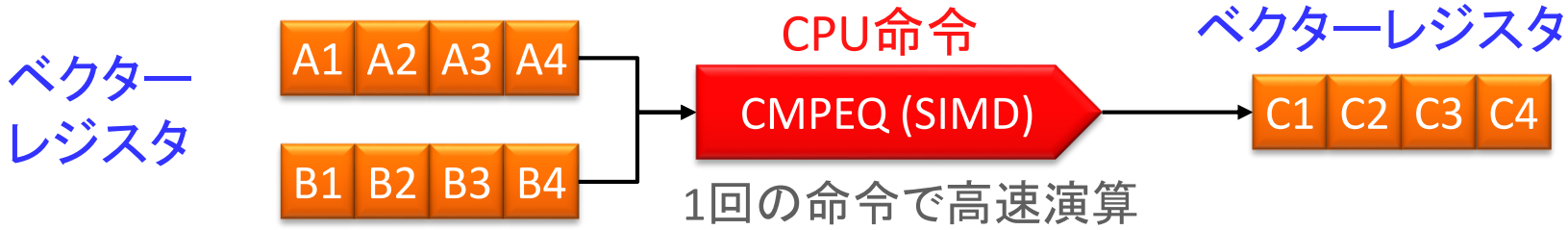
4回の一致比較の場合

IF (A1 ＝ B1) → C1
IF (A2 ＝ B2) → C2
IF (A3 ＝ B3) → C3
IF (A4 ＝ B4) → C4

通常の命令セットの場合(1組のデータ演算から1つの結果を算出)



SIMD命令セットの場合(複数のデータを1回の演算命令で高速実行)



カラム型表は何故分析用クエリーが高速か？

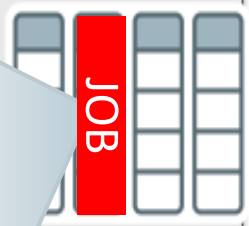
ポイント4

最新のプロセッサで搭載されているSIMDにより高速スキャン

インメモリ領域

EMP表

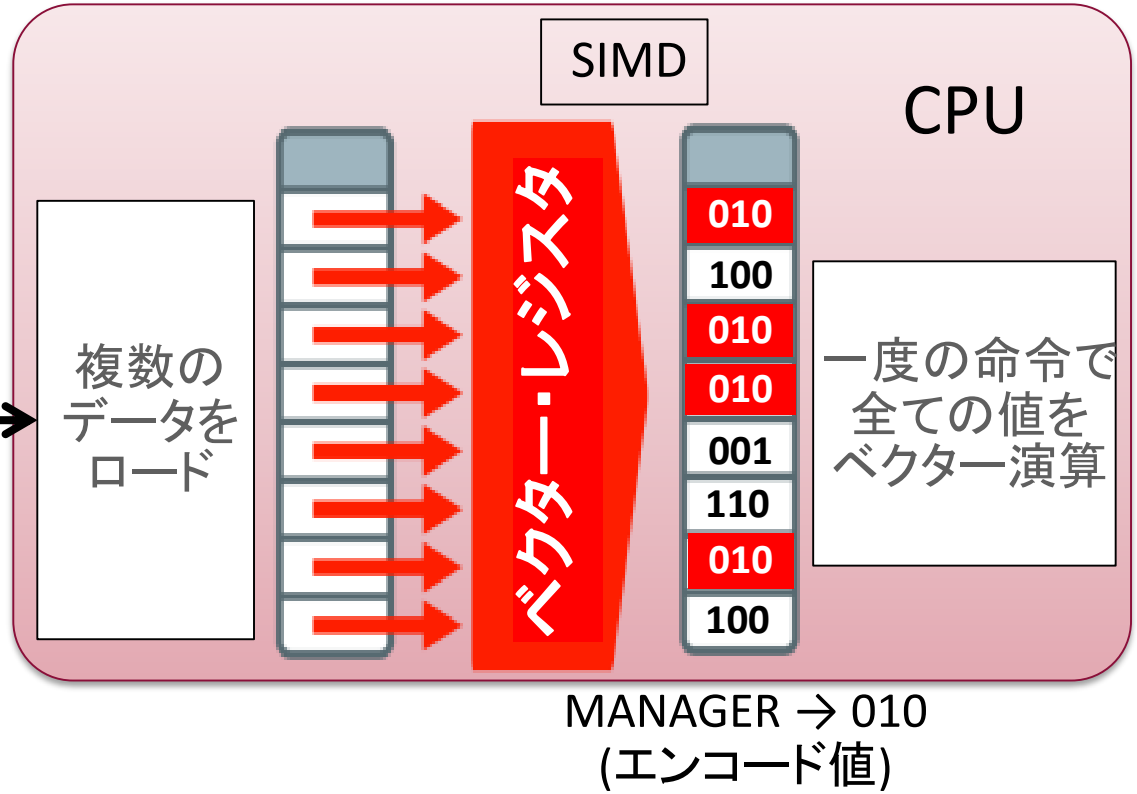
JOBカラム値	ディクショナリ値	ビット表現
ANALYST	0	000
CLERK	1	001
MANAGER	2	010
PRESIDENT	3	011
SALESMAN	4	100



001 | 100 | 100 | 010 | 100 | 010 | 010 | 000 | 011

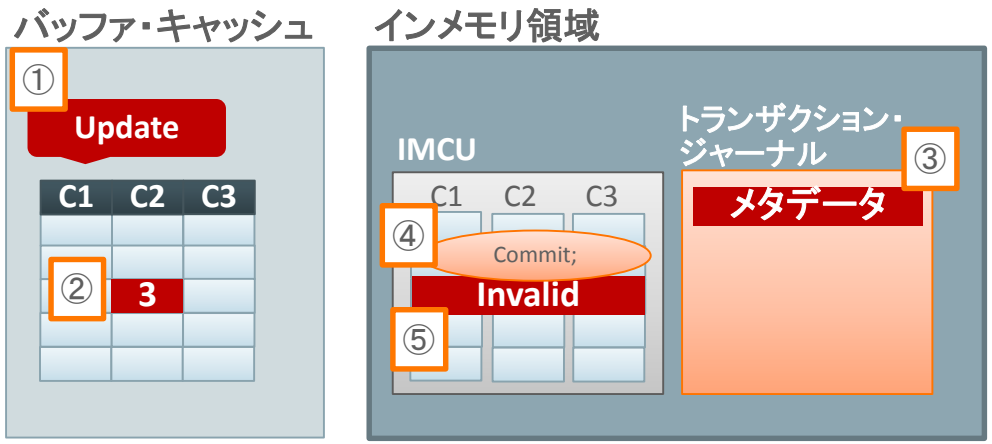
ディクショナリ圧縮により
実データ値をビットデータとして扱うことで**より多くのデータをCPUレジスタにロード可能**

例:「MANAGER」職種を検索
(MANAGER → 010)



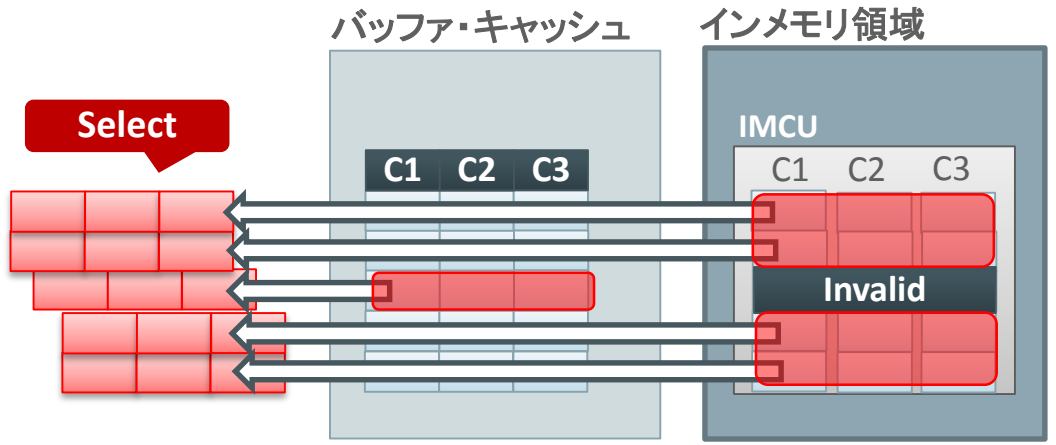
インメモリ・デュアル・フォーマットでの読取一貫性の確保

- 従来通りバッファ・キャッシュ(ロー型)でデータ・ブロックを更新
- 同時にトランザクションの一時的な情報をトランザクション・ジャーナルに記録
- コミット時IMCU内の該当行に対する無効化



従来の更新操作に加えIMCU内の行の無効化を実施

- インメモリ領域は古い更新前データを利用しない(常に最新のデータを返す)
- 無効な行はバッファ・キャッシュもしくはトランザクションジャーナルから読取り
- 読取一貫性はインメモリ領域とバッファ・キャッシュの内容の結合で保たれます。

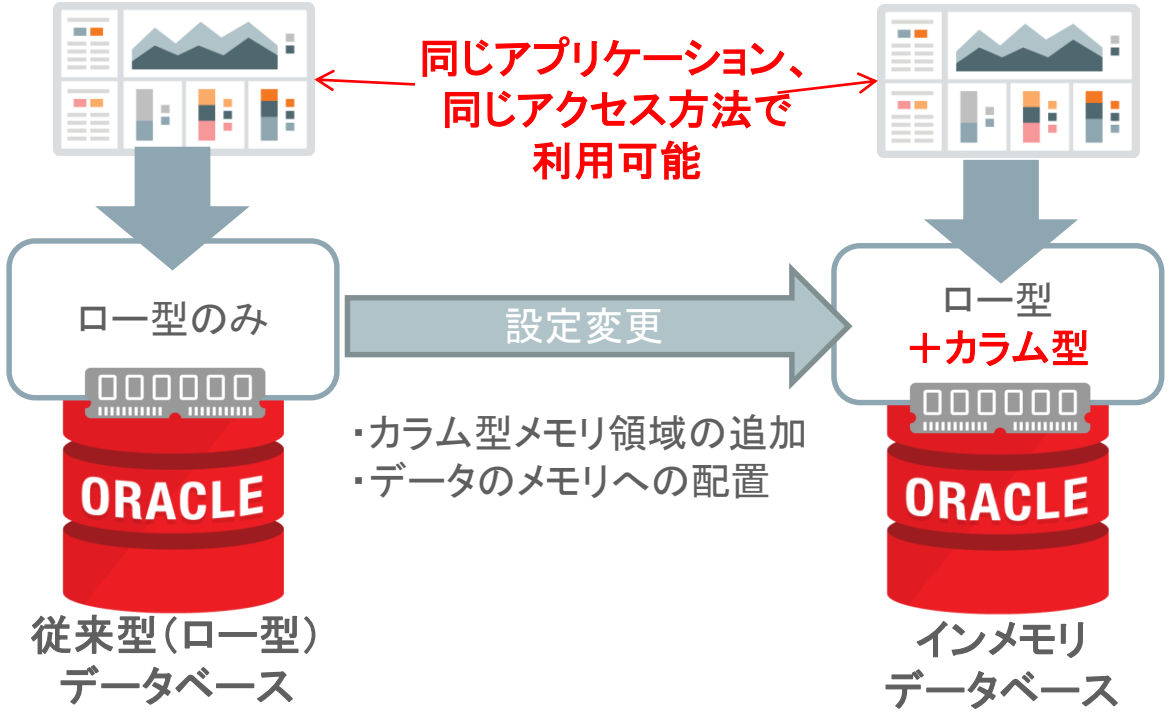


インメモリ領域とバッファ・キャッシュの情報を結合

バッファ・キャッシュとインメモリ領域の両方にデータを保持しますが、両者の整合性を保ちながら動作するため読み取り一貫性が確保されます。

Oracle Database In-Memoryを活用したアプリケーション開発

通常のOracle Databaseと同様の手法でアプリケーションを開発できます。



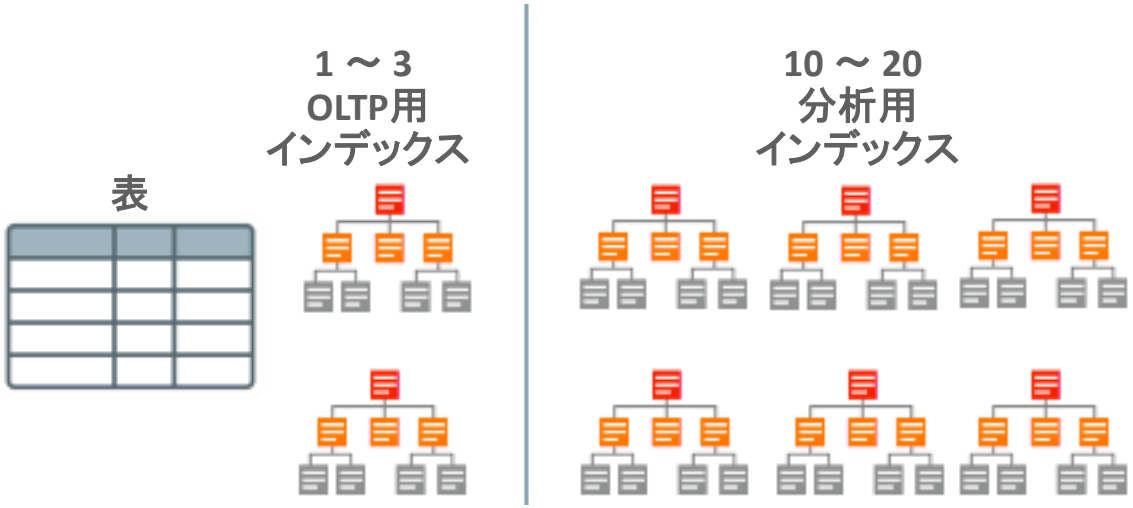
- **Oracleのフル機能を利用可能**
  - SQLに制限なし
- **容易な実装**
  - 簡単な設定変更でインメモリ化可能
- **通常Oracleとの完全互換**
  - 既存アプリケーションの修正は不要
- **プラットフォーム非依存**
  - これまで通り自由にH/Wを選択可能

通常のOracle Databaseと同様の言語、開発手法が利用できるため、これまで通りの方法でアプリケーションを開発できると共に、既存アプリケーションからの移行も容易です。

分析処理におけるインデックスが不要

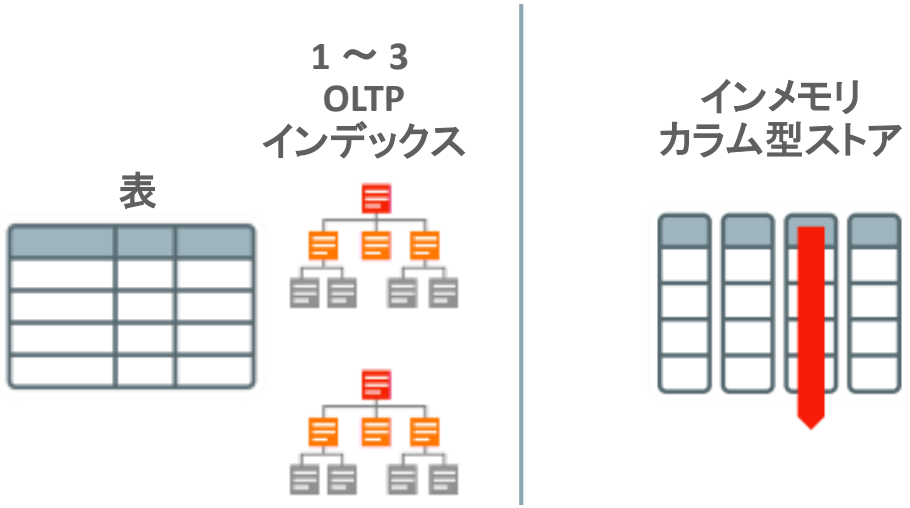
既存のOracle Database

- インデックスは事前予測可能なパターンのみ高速化
- 更新処理は10～20個のインデックスの更新が必要
  ※更新パフォーマンス低下の原因となります。



Oracle Database In-Memory

- インデックスなしで、すべての列の処理を高速化
- **アドホッククエリ（自由検索）も可能**
- カラム型ストアのためのディスクI/Oはなし



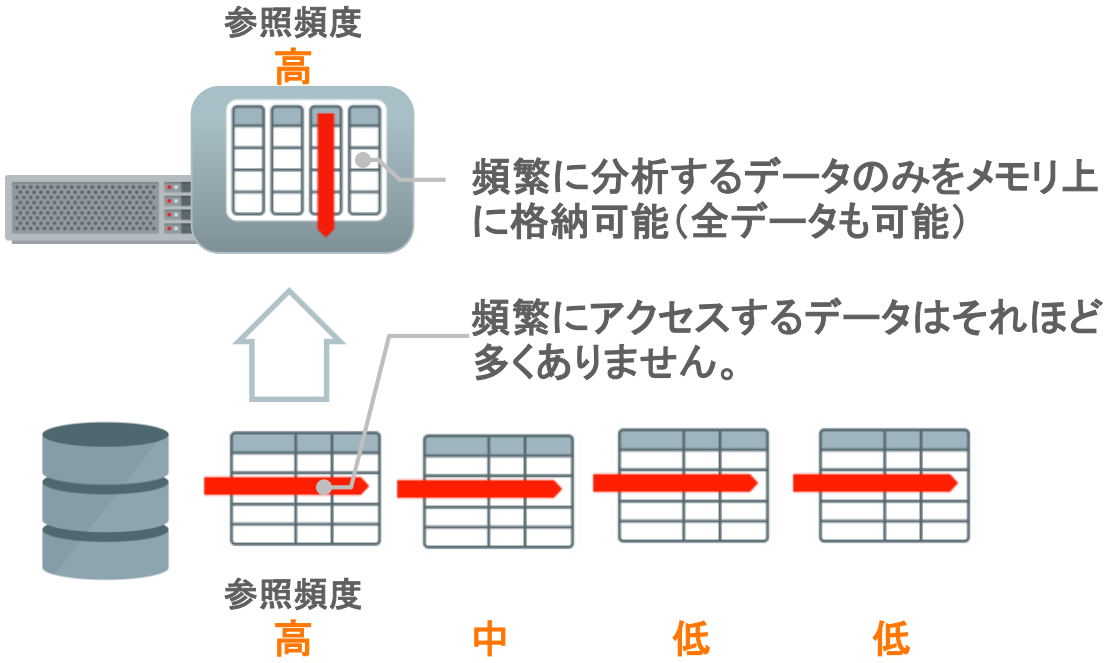
既存のOracle Databaseでは分析処理で適切な性能を担保するために、インデックスをメンテナンスする必要がありました。Database In-Memoryでは、カラム型の特性によりインデックス不要で最適な性能を担保可能です。

コスト効率性の高いインメモリデータ配置

スモールスタートから全データのインメモリ化まで可能なアーキテクチャ

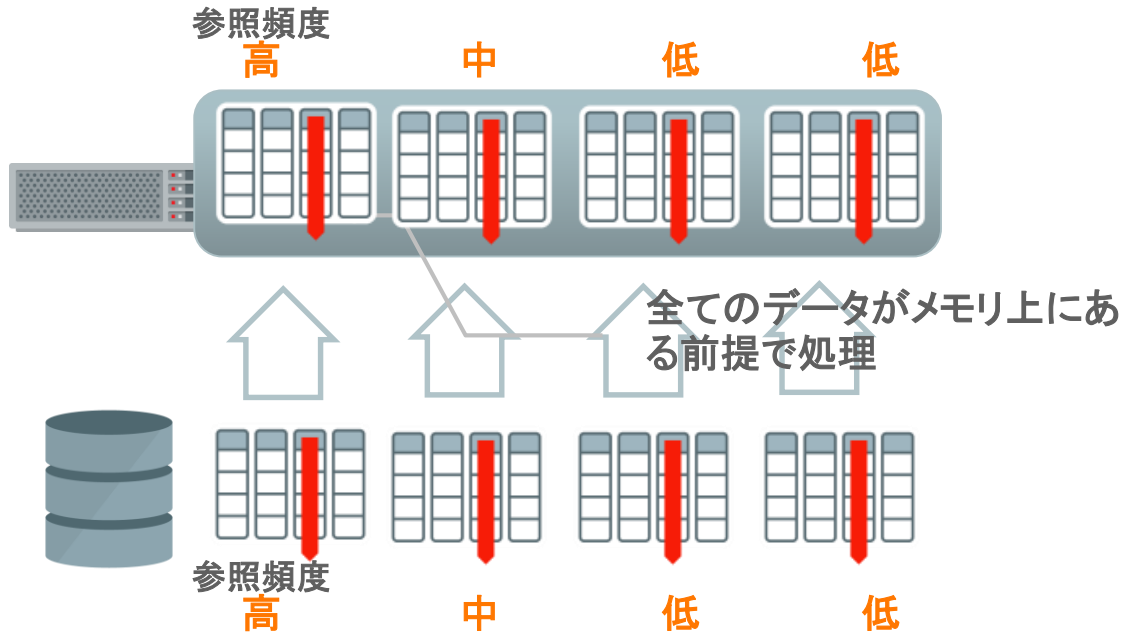
Oracle Database In-Memory

- 高速化対象データのみをメモリ上に展開



一般的なインメモリ・データベース

- 基本的に全てのデータをメモリ上に展開



すべてのデータを載せるタイプでは、あまり利用しないデータまでメモリに載せるため効率が悪くなることもありますが
Oracle DBはDRAM,Flash,Diskを効果的に利用するため、メモリ搭載量に依存せず大容量データが利用できます。

Q)Oracle Database In-Memoryを使った検索はどのようなSQLで効果が高いのか？

→ 特に大量行の検索で効果が高い

DBIMの利用効果の大きい処理

少数列に対する大量行の処理に効果大

1. 処理特性
ディスクI/Oの物理読込（Physical Reads）量が大きいもの

- 大量データの全表走査（Full Table Scan）
- 大量データの索引走査（Index Range Scan/Bitmap Index Scan）

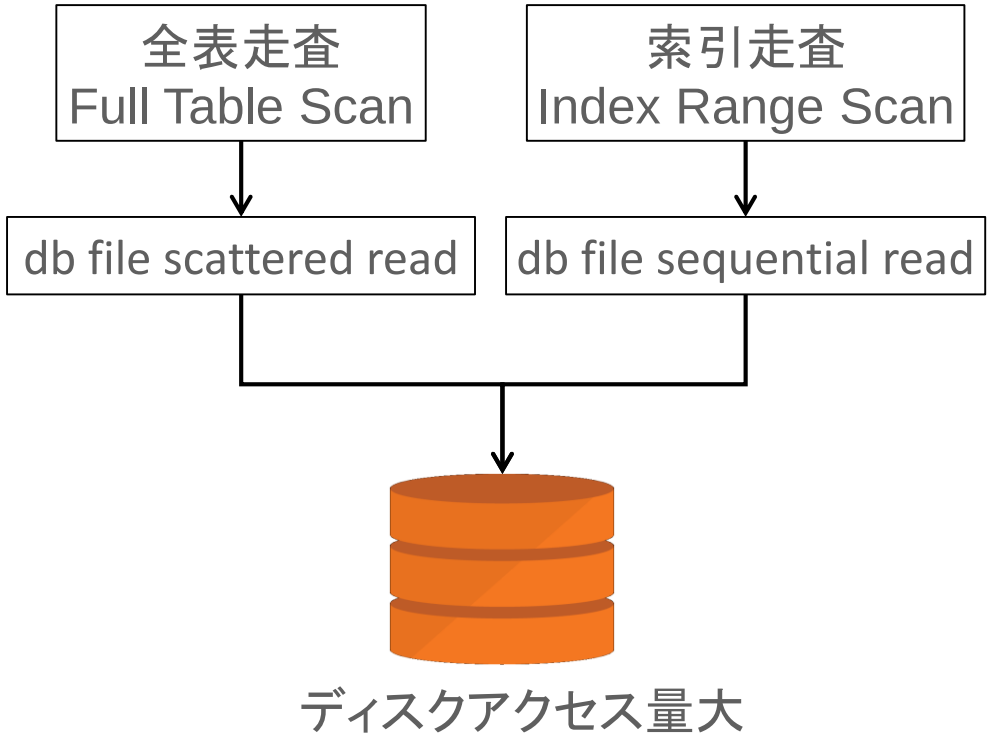
2. SQLの特徴
大量行の表を含む分析クエリ

- 複数表を利用した結合処理とフィルタ条件（WHERE xxx = :abc）
- 集計演算処理（特に MAX, MIN, COUNT）
- 中間一致検索（ユニーク値の列は除く）

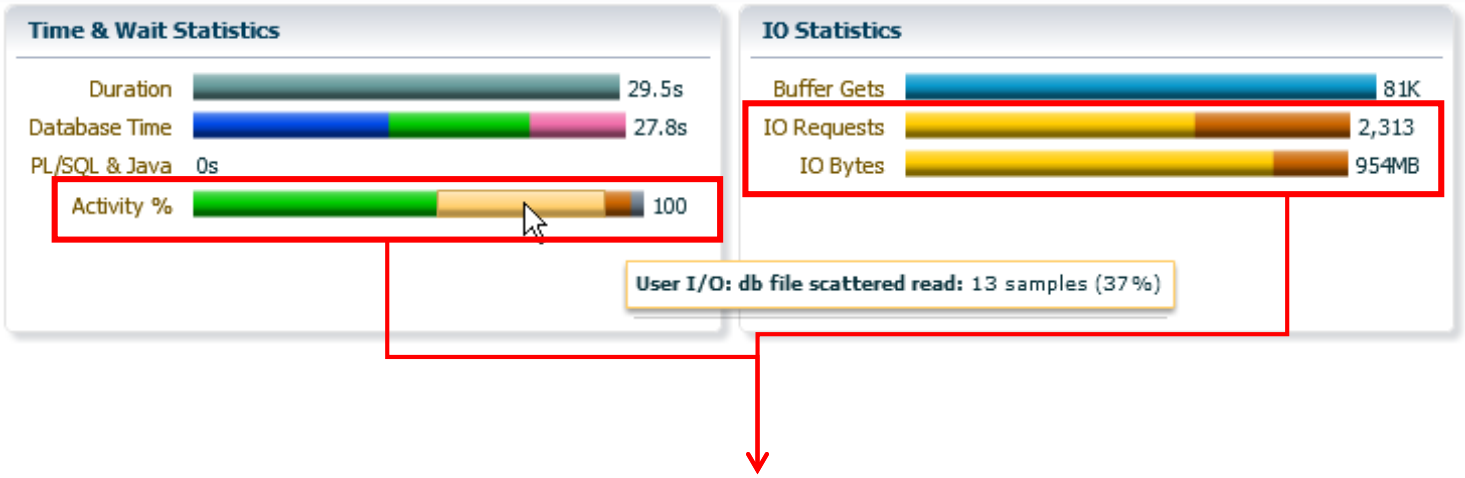
DBIMの利用効果の大きい処理

ディスクI/Oの物理読込（Physical Reads）量が大きいSQLとは？

インメモリ化によるパフォーマンス改善が
期待できる処理



SQL Monitorによる確認例



SQLの処理で多くのディスク・アクセスが発生している
(User I/O: db file scattered read)

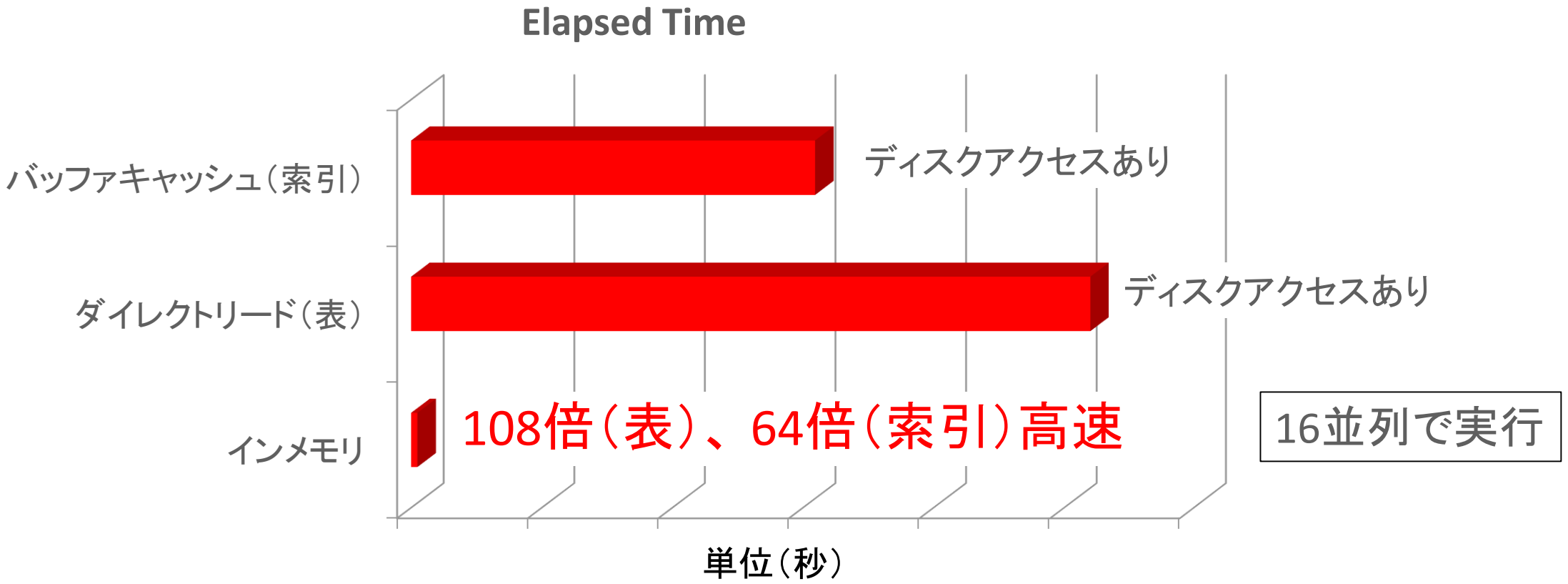
このようなSQLであればインメモリ化による
パフォーマンス改善の効果を期待できる

DBIMの利用効果の大きい処理

大量行の表を含むジョイン処理、集計処理でディスクアクセスがある場合と比較

```
select sum(lo_quantity) from lineorder, customer
where lo_custkey = c_custkey and c_nation in ( 'JAPAN', 'CHINA' ) ;
```

LINEORDER: 3億件
CUSTOMER: 150万件



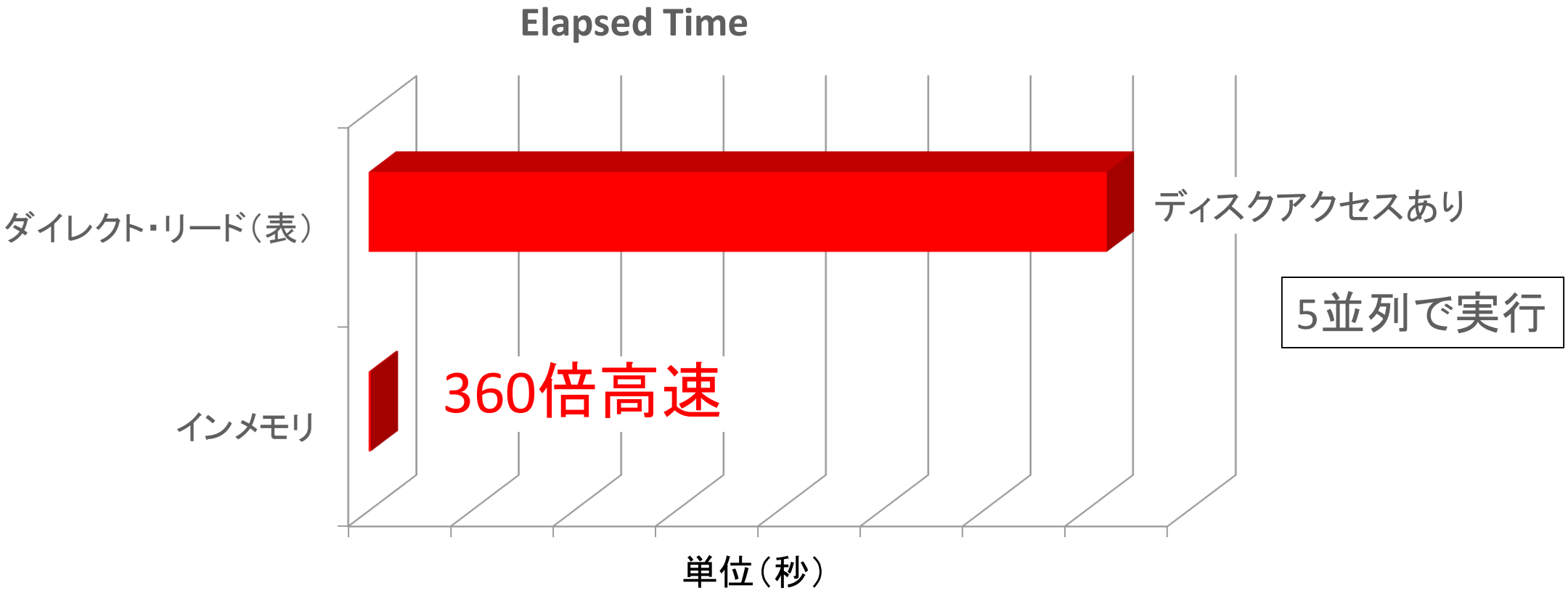
16並列で実行

DBIMの利用効果の大きい処理

大量行の表を含むジョイン処理、集計処理でディスクアクセスがある場合と比較

```
select max(lo_quantity) from lineorder
where lo_orderkey between 1 and 50000000 ;
```

LINEORDER: 3億件



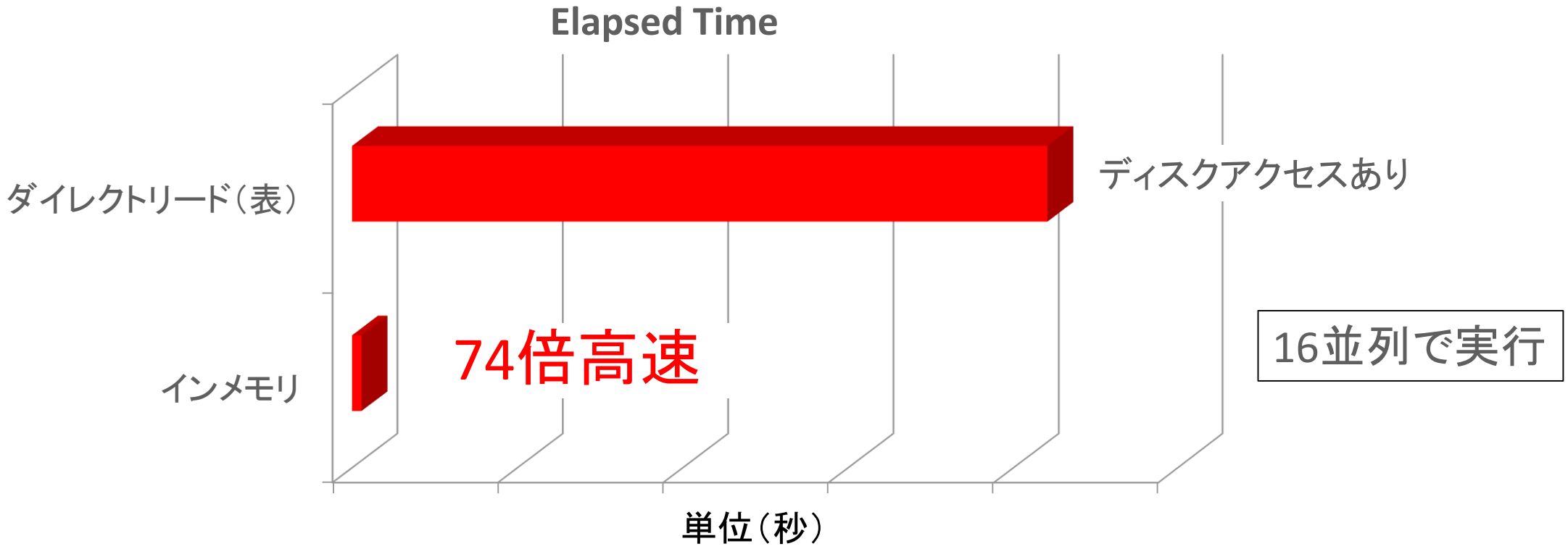
5並列で実行

DBIMの利用効果の大きい処理

大量行の表に対する中間一致検索でディスクアクセスあり

```
select sum(lo_quantity) from lineorder, customer
 where lo_custkey = c_custkey and c_region like '%RICA%' ;
```

LINEORDER: 3億件
CUSTOMER: 150万件



16並列で実行

Q)大きなサイズのバッファ・キャッシュを確保して、全データをキャッシュ内に保持して実行すればインメモリ検索と変わらないのか？

→ **No**

検証1）フル・キャッシュ（行） vs インメモリ（列）

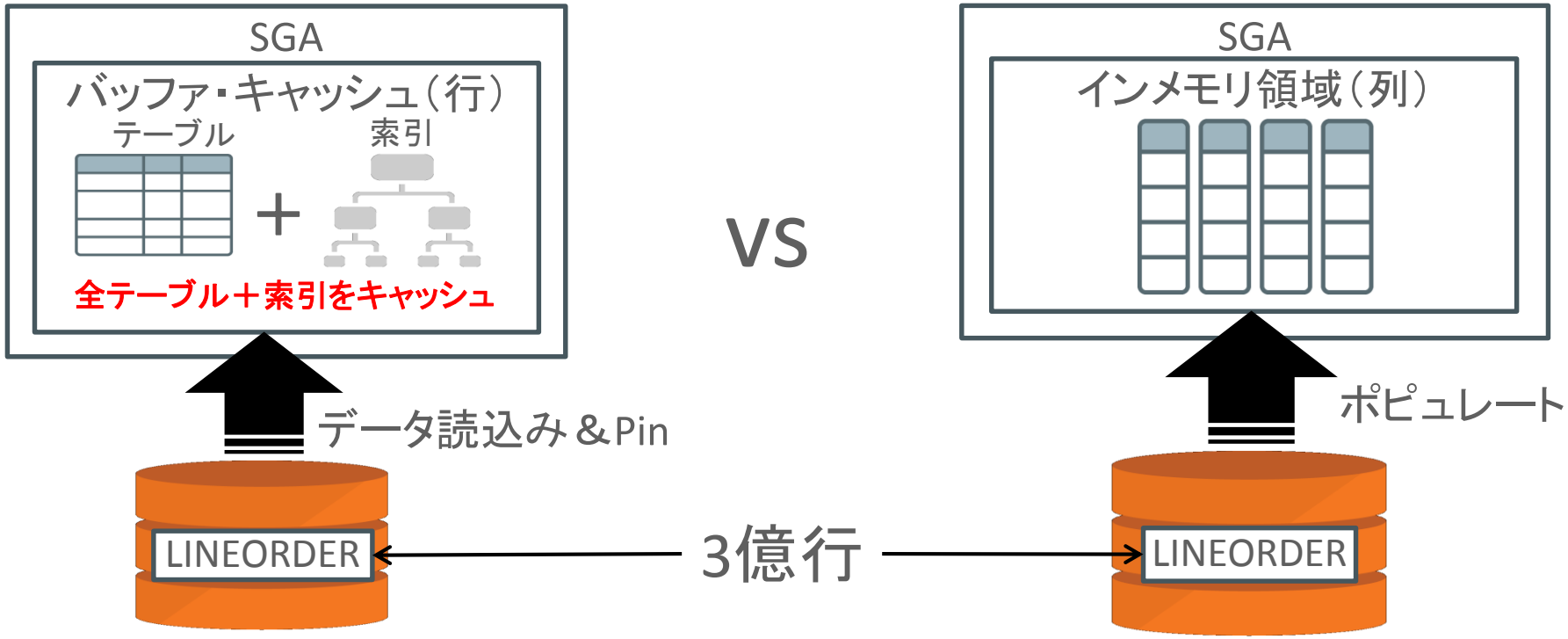
全表スキャン vs 索引スキャン vs インメモリ・スキャン

バッファ・キャッシュに表の全データがキャッシュされている場合とインメモリ検索の場合のパフォーマンスの違いを計測

実行SQL

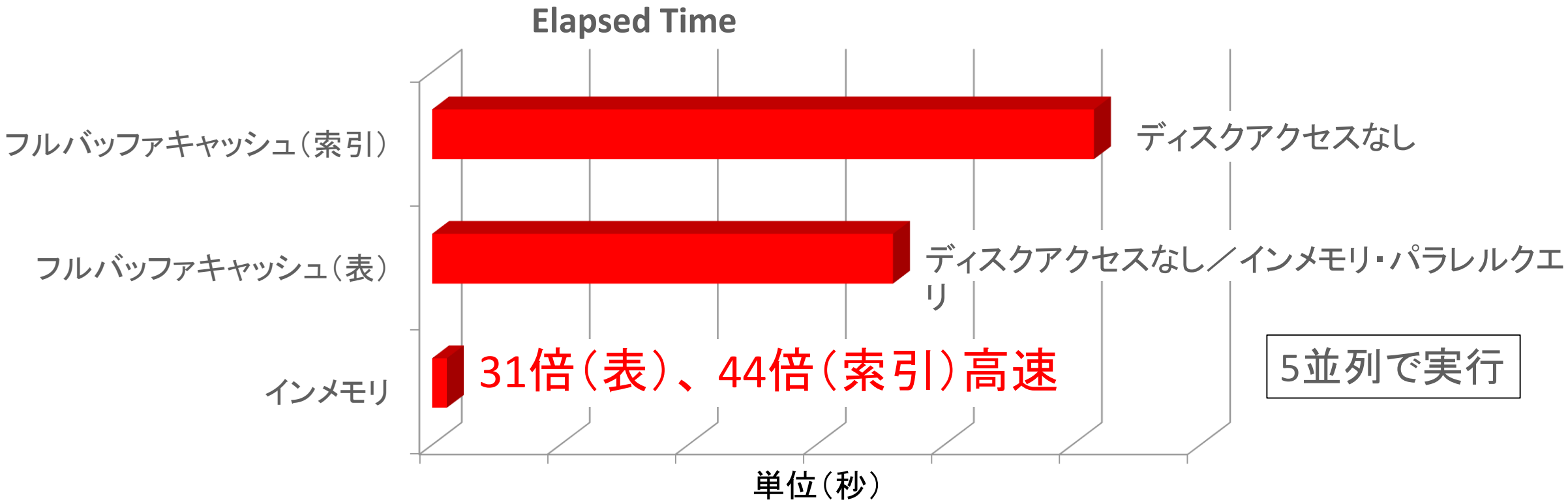
3億行から5,000万行を抽出する

```
select sum(lo_quantity) from lineorder
where lo_orderkey between 1 and 50000000 ;
```



検証1）フル・キャッシュ（行） vs インメモリ（列）

3億行の表から5,000万行を抽出するSQL



5千万行のアクセスを高速にアクセス出来る理由

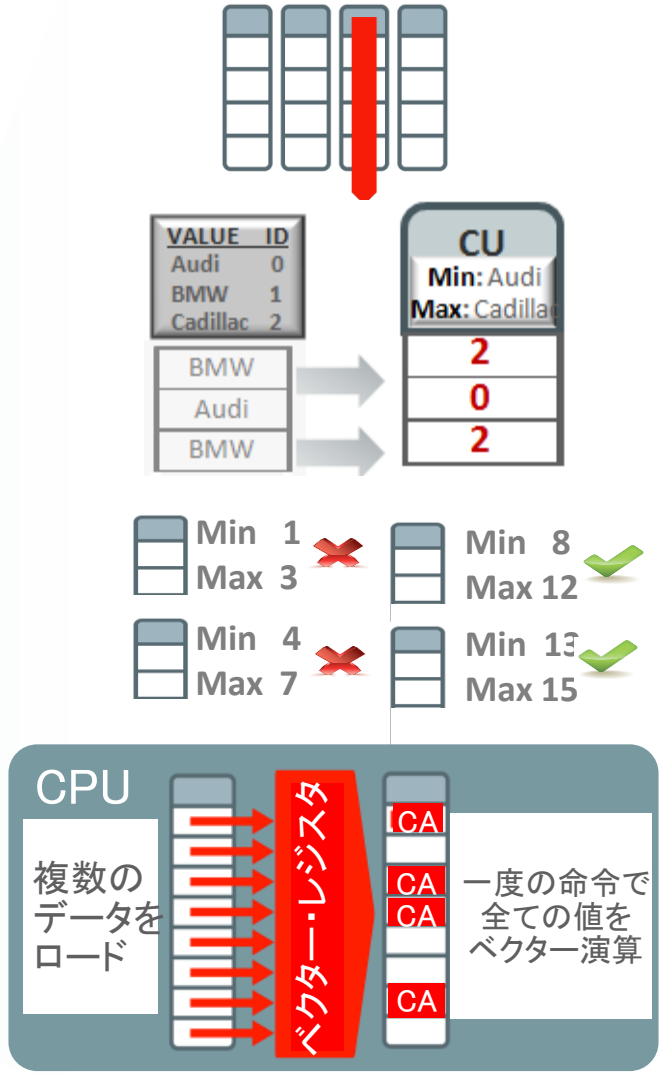
再掲）インメモリ・スキャンの高速フィルタリング

全表走査

Table Access Inmemory Full

データ読込み
（インメモリ・スキャン）

フィルタリング



1. 必要なカラムのみアクセス

2. 効果的な圧縮技術により圧縮した状態で検索が可能
(ディクショナリ圧縮)

3. インメモリ・ストレージ索引により
最小限のIMCUのみスキャン

4. 最新のプロセッサで搭載されているSIMDにより高速スキャン
（ベクター・スキャン）

Q)インメモリ化すると全ての検索処理を
高速化出来るのか？

→ **No**

DBIMの利用効果が大きくない処理

1. 索引により最適化されている
（大きな表から少ない行データを検索する）

- 数億／数千万行の表から1～数百行取得するような検索
- ハードウェア・リソースの観点からインメモリ検索の
並列度を高く設定出来ない

2. 集計値を別の仕組みで保持している

- マテリアライズド・ビューにより集計値を保持
- 同一内の別カラム、また別表に集計値を保持

少ない検索結果件数による比較

索引 vs インメモリ

- 6億行の表から530行を検索するSQL（シリアル実行）

```
select lo_quantity from lineorder where lo_custkey = 1499999 ;
```

	索引	インメモリ
Elapsed Time	0.04秒	0.13秒

→ 表の格納行数が多く、検索結果件数が少ないほど索引が若干速くなるケースが多い

→ SQLヒントを指定しないと索引アクセス処理になる

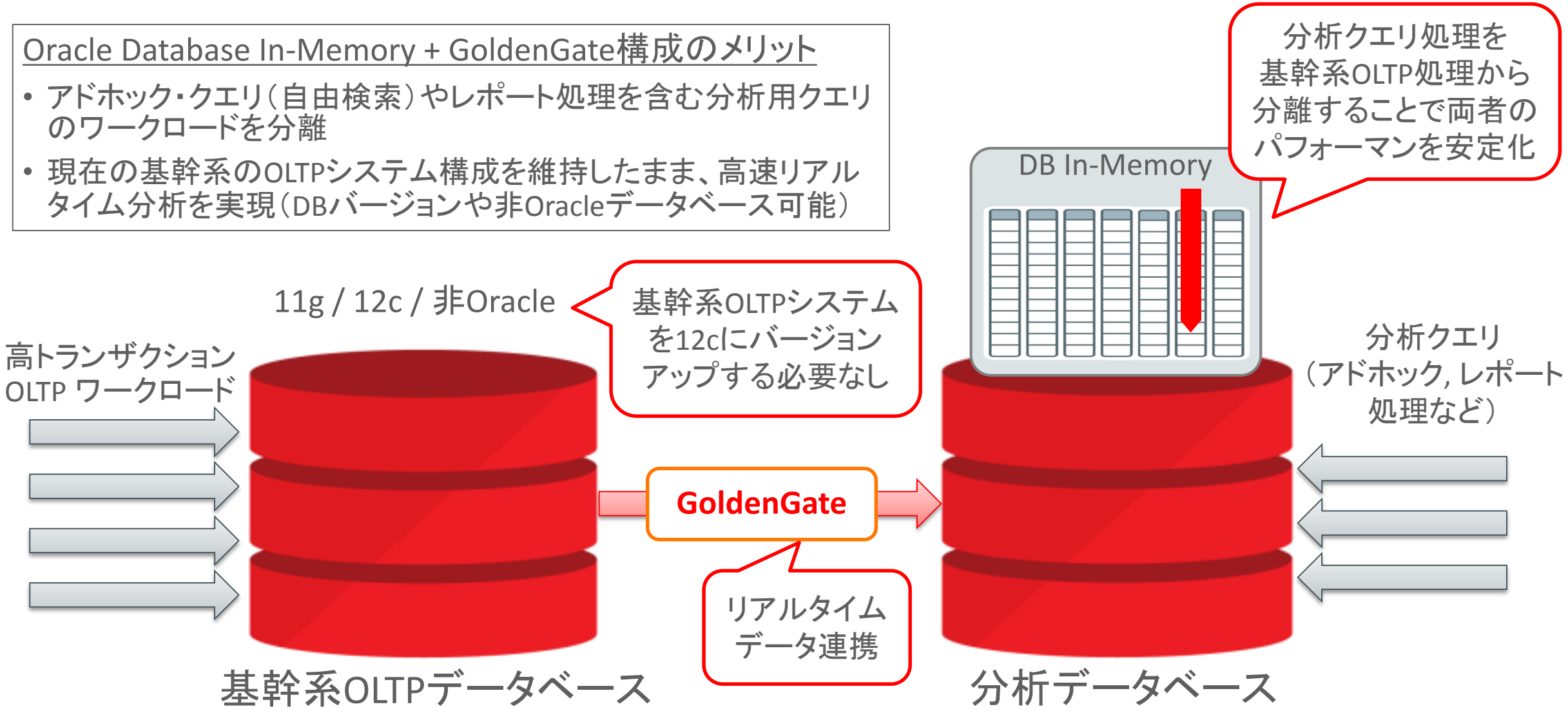
本日お伝えしたい内容

1. Oracle Database In-Memoryご紹介
2. 事例紹介　※スクリーンのみ表示
3. Oracle Database In-Memoryの応用構成

ご参考）Oracle Database In-Memory + GoldenGate 構成

Oracle Database In-Memory + GoldenGate構成のメリット

- アドホック・クエリ（自由検索）やレポート処理を含む分析用クエリのワークロードを分離
- 現在の基幹系のOLTPシステム構成を維持したまま、高速リアルタイム分析を実現（DBバージョンや非Oracleデータベース可能）

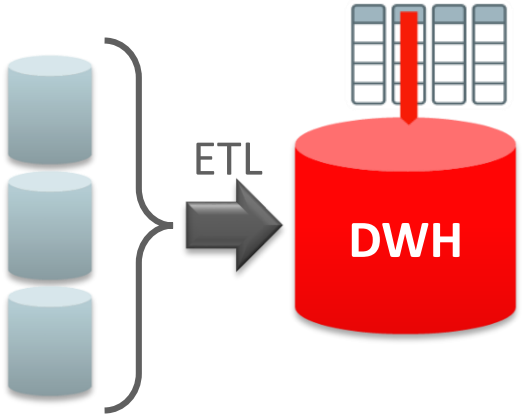


本日お伝えしたい内容

1. Oracle Database In-Memoryご紹介
2. 事例紹介
3. Oracle Database In-Memoryの応用構成

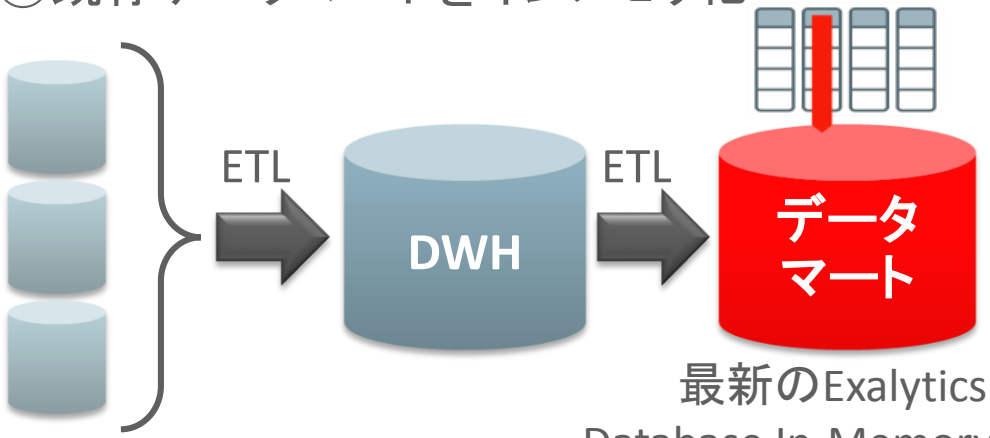
Oracle Database In-Memoryの基本構成パターン

①既存DWHをインメモリ化（現在の主流事例）



- 既存Oracle 11gのDWHを12cに移行してインメモリ化
- 既存DWHアプライアンスをOracle 12c にリプレース

②既存データマートをインメモリ化



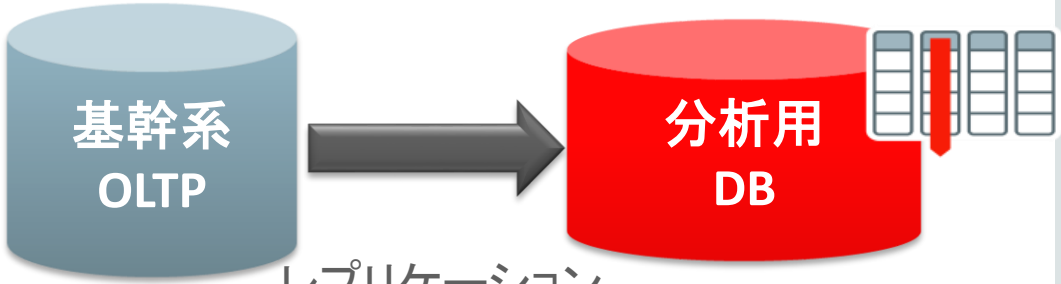
最新のExalyticsは
Database In-Memoryを包含

③OLTPシステムを直接インメモリ化

- 分析、レポーティングの高速化
- ダッシュボード画面の高速化（経営／業務系）
- リアルタイム分析の促進

④基幹系OLTPシステムとの分離構成

既存システムを極力変更しない　分析処理を負荷分散

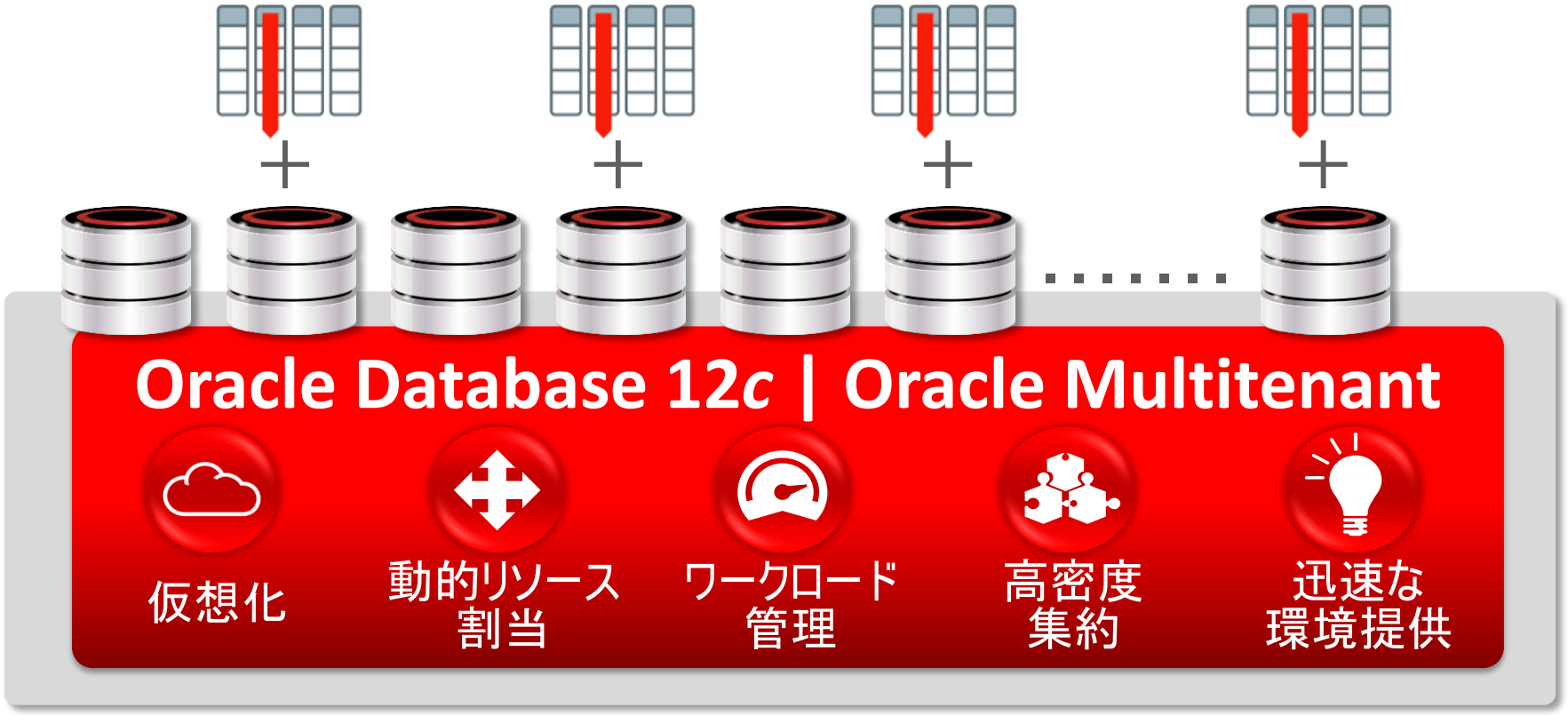


レプリケーション
（GoldenGateなど）

Oracle Database In-Memoryの応用パターン

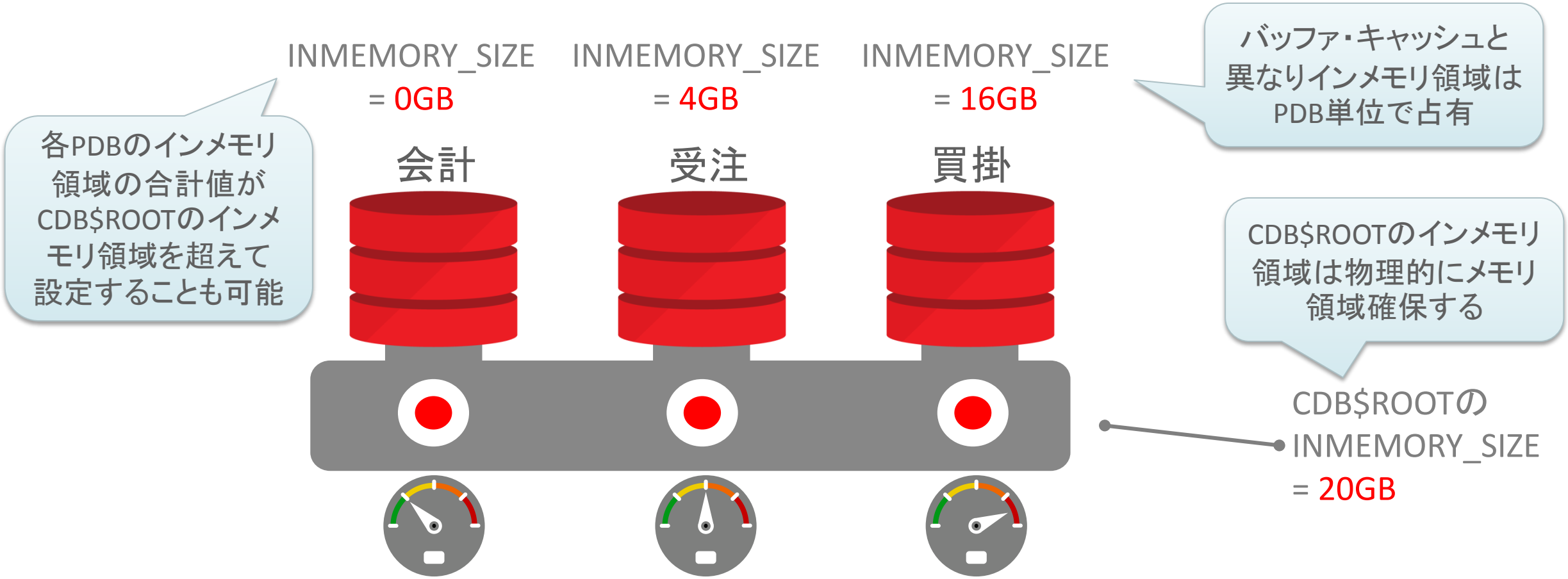
Oracle Multitenantを利用したPrivate Cloud（DBaaS）へのインメモリ展開

Oracle Multitenantは、データベースのPrivate Cloud化を行う最適なプラットフォームであり、ユーザのリクエストに応じて、各PDB内にOracle Database In-Memoryのインメモリ領域を構成可能です。



Oracle Database In-Memory と Multitenantの構成

Oracle Multitenantで定義されたCDB構成にインメモリ領域を定義する場合、CDB全体の設定値（メモリ領域確保）と各PDB単位の設定値（メモリ領域確保はされず予約のみ）を行い、PDB単位で各インメモリ領域を占有します。

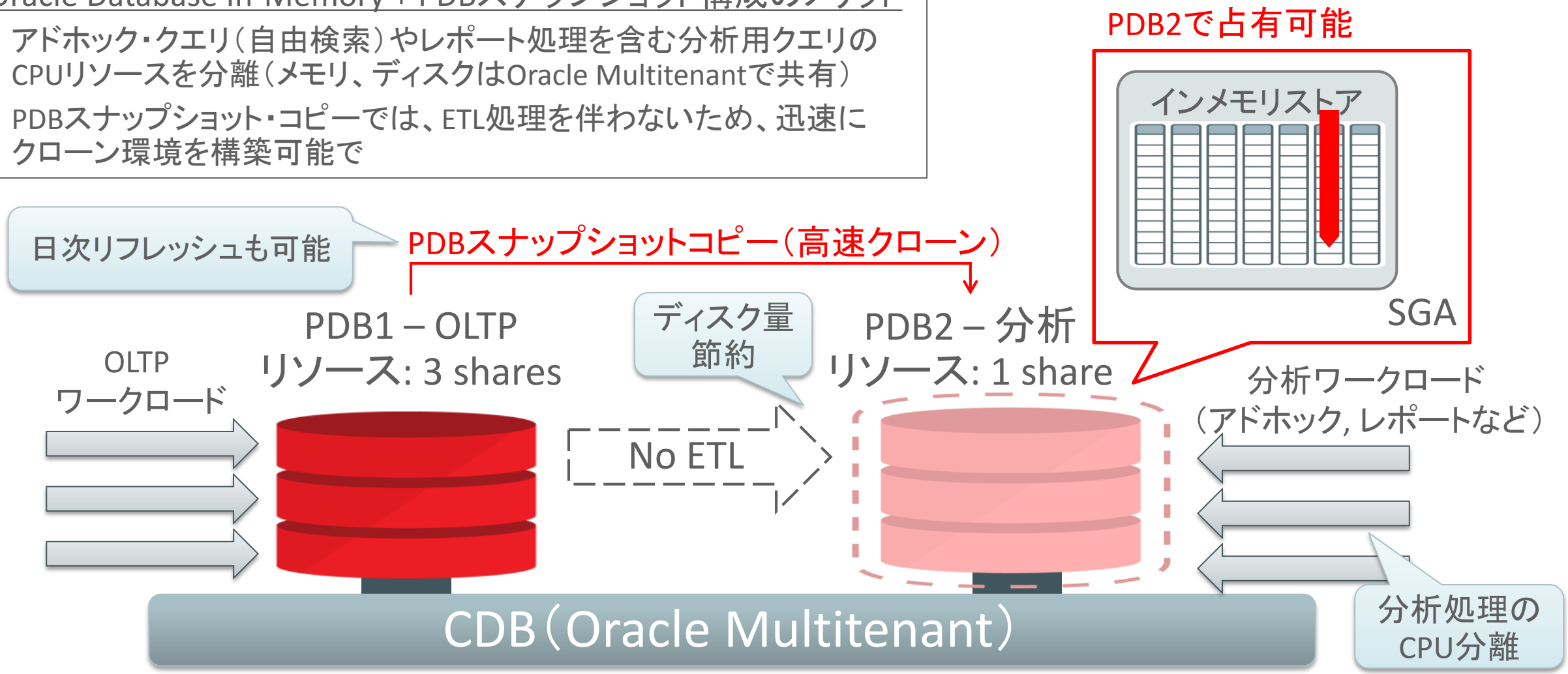


Oracle Database In-Memory + PDBスナップショット・コピー

Oracle MultitenantのPDBスナップショットによる迅速な環境構築

Oracle Database In-Memory + PDBスナップショット構成のメリット

- アドホック・クエリ（自由検索）やレポート処理を含む分析用クエリのCPUリソースを分離（メモリ、ディスクはOracle Multitenantで共有）
- PDBスナップショット・コピーでは、ETL処理を伴わないため、迅速にクローン環境を構築可能で



ご参考）スナップショット・コピーによる環境構築時間とストレージ容量削減

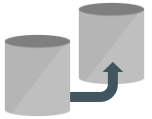
Oracle Sun ZFSストレージ・アプライアンスを使ったテスト（オラクル社内テスト）

Oracle MultitenantのPDBスナップショット・コピー機能を使うと**迅速に検証環境、クローン環境を少ないディスク容量で構築できる**ので、**作業工数、および、ディスクのコストを削減**できます。

フルサイズ (GB)	スナップサイズ (KB)	相対サイズ	フルクローン	スナップクローン	% Savings
24	140	0.00058%	9 min, 52 sec	1 min, 52 sec	80%
216	142	0.00007%	1hr, 21 min	2 min, 11 sec	97%
1300	551	0.00004%	9hr, 7 min	5 min 55 sec	99%

Oracle Database In-Memoryのクラウド展開

Oracle DB Cloud（パブリック・クラウド）の中のOracle Database In-Memory

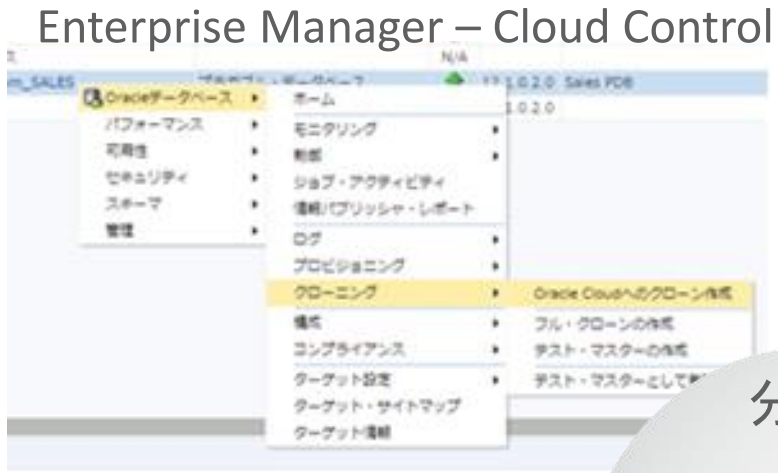
Standard Edition	Enterprise Edition	EE High Performance	EE Extreme Performance
	Adds…	Adds…	Adds…
• 完全なデータベース・インスタンス • 最大16 OCPUまで	• 全てのEE 標準機能 • データベース透過的暗号化（Transparent Data Encryption）	 マルチテナント  Partitioning  Advanced Compression  Advanced Security, Label Security, Database Vault  Real Application Testing  OLAP, Analytics, Spatial and Graph  Management Packs	 DB In-Memory  Active Data Guard * **全てのデータベースオプション機能が利用可能**

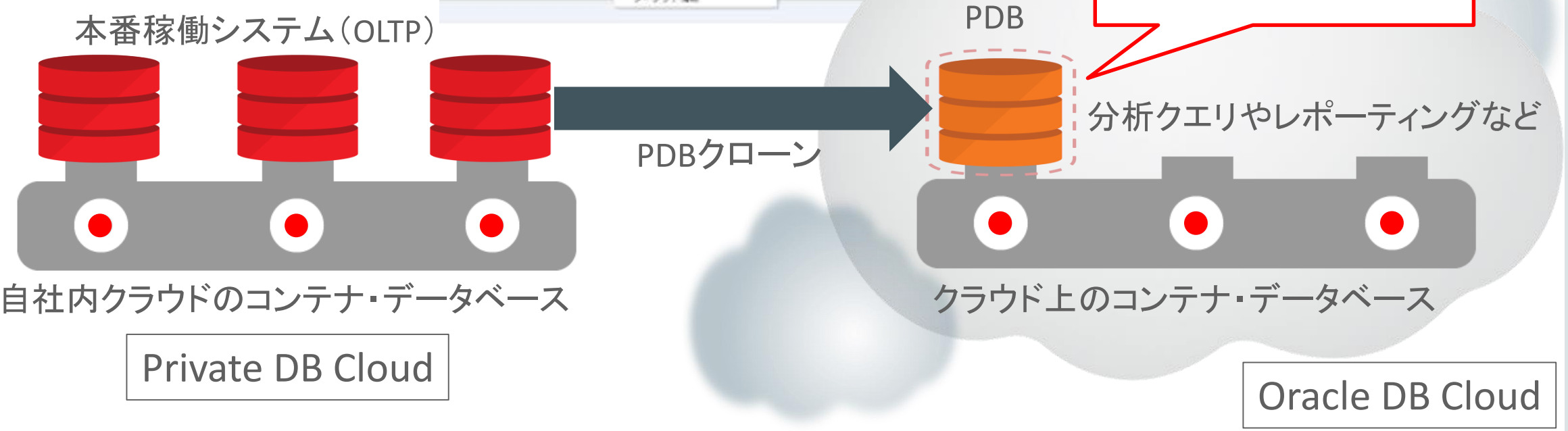
オンプレミスで提供されるのと同じOracle Database ソフトウェアをクラウドでも提供

Database In-MemoryのOracle DB Cloudへの展開

ハイブリッド構成のメリット

- H/Wコスト、S/Wコスト、運用コスト低減が可能
- 本番OLTPシステムから分析システムを分離することで負荷分散
- 利用期間が限定される開発環境構築やインメモリ化の評価を迅速に実施





まとめ

- ビッグデータ時代に向けて大容量のデータを高速処理するインメモリ・データベースのニーズが高まっている
  - Oracle DB In-Memoryはロー型とカラム型の両方のフォーマットをデータベース・オプティマイザが実行するSQLに合わせて最適な方式を選択することで、全てのワークロードを高速化可能
  - Oracle DB In-Memoryは大量行の検索に大きな効果を発揮
  - 大きな表から少ない行を抽出する場合は既存の索引の方が高速な場合もある
- Private / Public Cloud環境における迅速なインメモリ分析環境の構築、リリースが可能
  - Oracle DB In-Memory + Oracle Multitenant
  - Oracle DB Cloudの利用により、分析処理を負荷分散、Oracle DB In-Memoryの評価を迅速に実施
- Oracle DB In-Memoryの国内、海外の稼働事例は順調に増加中

参考情報

Oracle Database 12c Release 1 (12.1.0.2) コアテクセミナー（OTNセミナー・オンデマンド）

http://www.oracle.com/technetwork/jp/ondemand/od12c-coretech-aug2014-2283256-ja.html

YouTubeを使った動画セミナーにより
Oracle Database In-Memoryの製品
機能を解説しています

1. 概要
2. マルチテナント・アーキテクチャ
3. Upgrade / 12.1.0.2における変更点
4. Rapid Home Provisioning
5. Grid Infrastructure
6. JSON対応
7. Oracle Database In-Memory概要
8. Oracle Database In-Memory (2)
9. Oracle Database In-Memory (3)
10. Oracle Database In-Memory (4)
11. Oracle Database In-Memory (5)
12. Systems
13. Data Warehouse (DWH) 関連機能
14. Enterprise Manager

Oracle DBA & Developer Days 2014 資料ダウンロード

https://eventreg.oracle.com/profile/web/index.cfm?PKWebId=0x14073486e5

Oracle Database In-Memoryの講演資料がダウンロード可能

https://www.oracle.com/webfolder/s/delivery_production/docs/FY15h1/doc8/A1-1-print.pdf

トラック	セッション名
A1-1	パフォーマンス：Oracle Database In-Memoryが変革する次世代システムの可能性と導入／開発における考慮点
B1-1	オラクル・コンサルが語る！ Oracle Coherence活用ノウハウ、そして最新ユースケース

ご質問・ご相談等ございましたら、
お気軽にお問い合わせください。

あなたにいちばん近いオラクル

Oracle Direct

0120-155-096

（平日9:00-12:00 / 13:00-18:00）

http://www.oracle.com/jp/direct/index.html

Oracle Direct 検索

各種無償支援サービスもございます。

Integrated Cloud

Applications & Platform Services

Hardware and Software
Engineered to Work Together


ORACLE®