ORACLE

ORACLE

Oracle Database Technology Night #33

19c Multitenant Architecture – 応用・実践編

クラウド事業戦略統括
データベースソリューション部
橋本 琢爾

Dec, 2019

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。

Program agenda

1. Oracle Database Technology Night #32 を振り返る
2. マルチテナント関連の新機能(18c / 19c)
3. 統合・集約におけるリソース分離性
4. アプリケーション・コンテナと実装例
5. まとめ

Tech Night #32 まとめ - 19c マルチテナント・アーキテクチャ再入門

- ライセンス・ポリシーの変更 (SE2でも最大３つのPDBを実装可能)
- PDBの基礎的な利・活用についてはCDB/非CDBで大差無し(※)
- EE Option など拡張機能においても概ね制約無し(※)
- 統合・集約の進め方やリソース配分も従来(ベアメタル+非CDB)と同様

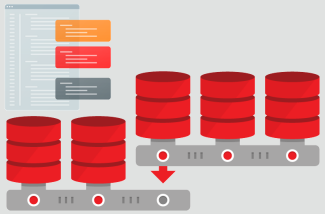
- 「共有と分離」を意識した設計・実装・運用・監視/管理の要素
- 新しいデータ共有の環境を提供 – Application Container

(※) 製品・機能の組み合わせや運用にも依存します

Tech Night #32 まとめ – 応用編の予告

18c / 19c 新機能確認

リソースの分離性・補足



Application Container とその活用

Program agenda

1. Oracle Database Technology Night #32 を振り返る
2. マルチテナント関連の新機能(18c / 19c)
3. 統合・集約におけるリソース分離性
4. アプリケーション・コンテナと実装例
5. まとめ

Tech Night #32 より

最新のライセンス・制約

Option / Pack				DBCS				
	SE2	EE	EE-ES	SE	EE	EE-HP	EE-EP	ExaCS
Oracle Multitenant - PDBの数	3	252	4096	3	3	4096	4096	4096
CDBフリート管理	N	N	Y	N	Y	Y	Y	Y
PDBスナップショット・カルーセル	N	N	Y	N	Y	Y	Y	Y
リフレッシュ可能PDBスイッチオーバー	N	N	Y	Y	Y	Y	Y	Y

https://docs.oracle.com/cd/F19136_01/dblic/Licensing-Information.html#GUID-0F9EB85D-4610-4EDF-89C2-4916A0E7AC87

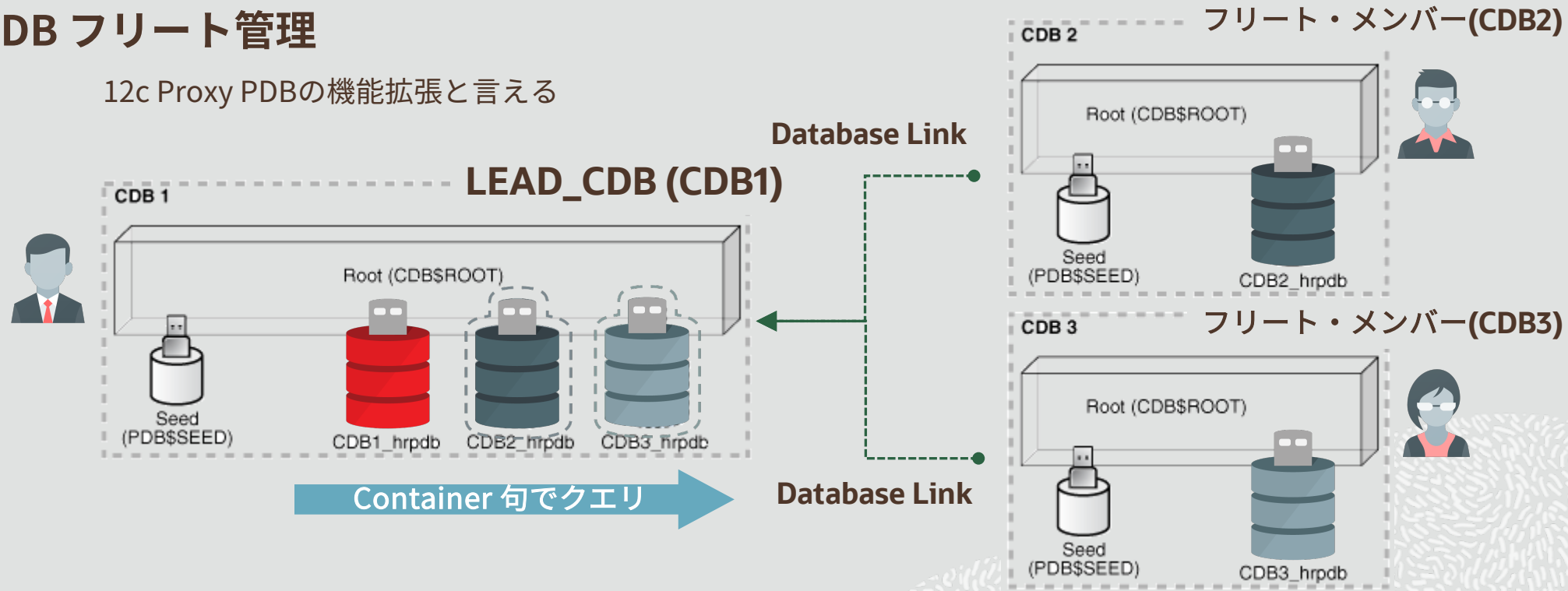
マルチテナント関連の新機能(18c / 19c)

対象項目	概要	
CDB フリート管理	複数のCDB-ROOTを論理的に一元管理可能	NEW IN 18c
PDB スナップショット・カルーセル	定期的にPDBのCloneを最大8つ取得(古いCloneは自動的にパージ)	
リフレッシュ可能 PDB switchover	リフレッシュ可能PDBのロール変換が可能	
スナップショット拡張	データベース管理PDBスナップショットの拡張機能	
PDB機能拡張	PDBのワークロードの取得およびリプレイ	
PDB機能拡張	PDBのADDM分析	
DB Vault 連携	インフラストラクチャ・データベース管理者向けのDatabase Vault操作の制御	NEW IN 19c
CDB/PDB機能拡張	同じCDBの複数のPDBシャードのサポート	
PDB機能拡張	PDBの自動再配置	
DBCAの機能向上	DBCAを使用したリモートPDBのクローニング /リモートPDBの再配置	
PDB機能拡張(資格証明)	データ・ポンプ・インポートに対するクラウドのオブジェクト・ストアのサポート	

マルチテナント関連の新機能(18c / 19c)

CDB フリート管理

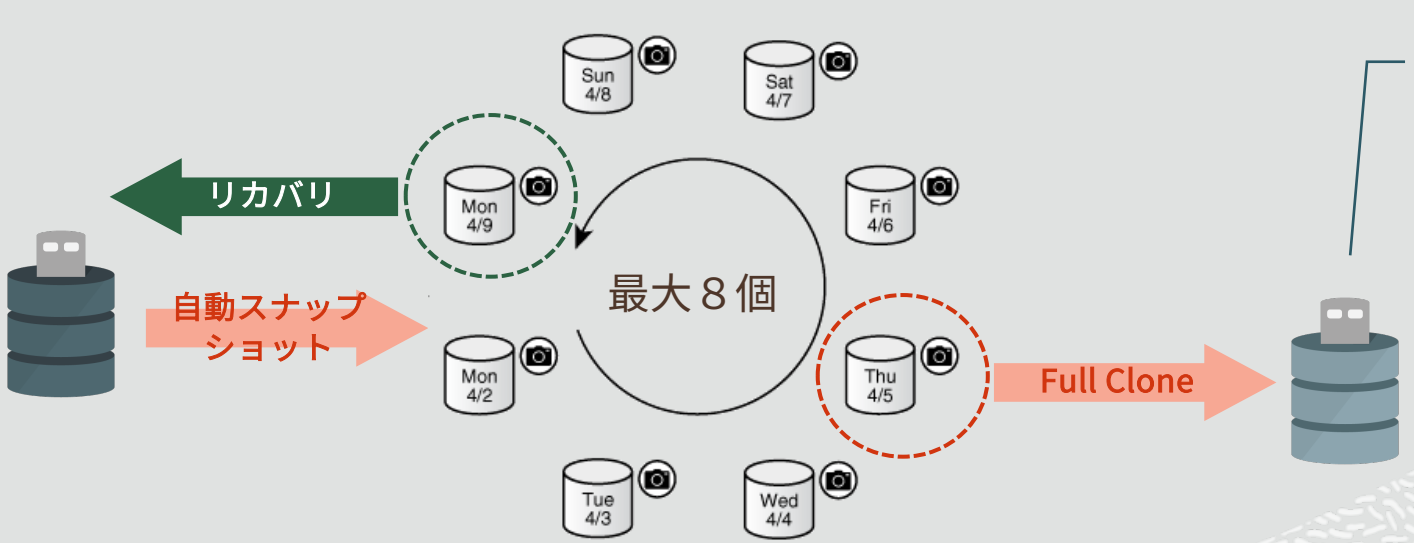
12c Proxy PDBの機能拡張と言える



https://docs.oracle.com/cd/F19136_01/multi/administering-cdb-fleet.html#GUID-0AA7FC65-F350-473B-96D2-976313514069

マルチテナント関連の新機能(18c / 19c)

CDB スナップショット・カルーセル



スナップショットを活用した、開発・検証

過去データによる、障害・不具合の再現ケース

EVERY 間隔句で自動的に作成
(EVERY snapshot_interval [MINUTES|HOURS])

https://docs.oracle.com/cd/F19136_01/multi/administering-pdb-snapshots.html#GUID-87183462-C87C-432C-BE7E-282B1F1202F4

マルチテナント関連の新機能(18c / 19c) – 確認!!

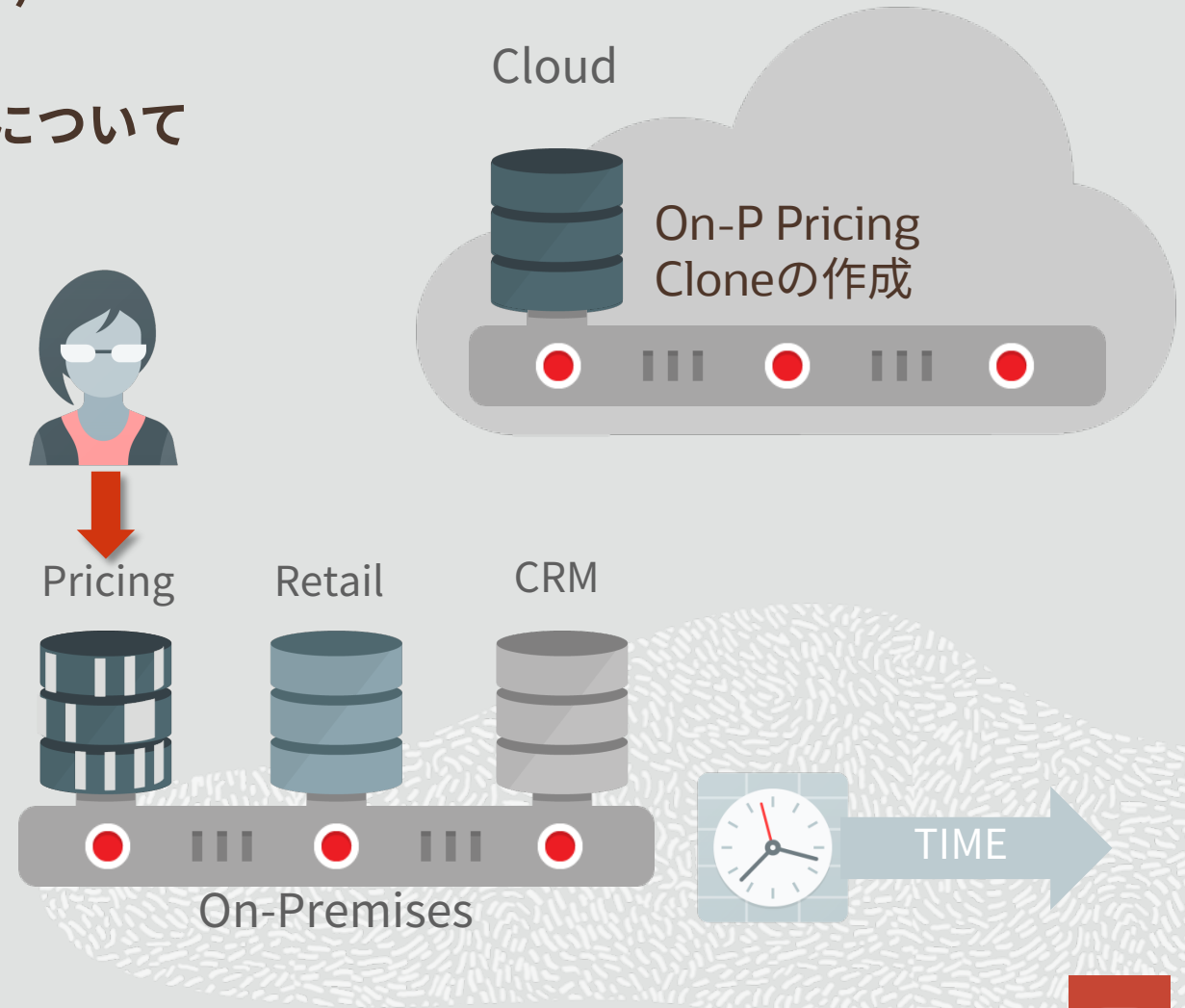
Tech Night #32 - B-2: 可搬性(ポータビリティ)について

プロビジョニングの強化

オンラインでクローンを作成可能

リフレッシュ可能クローン

最新データでクローンを差分更新



マルチテナント関連の新機能(18c / 19c)

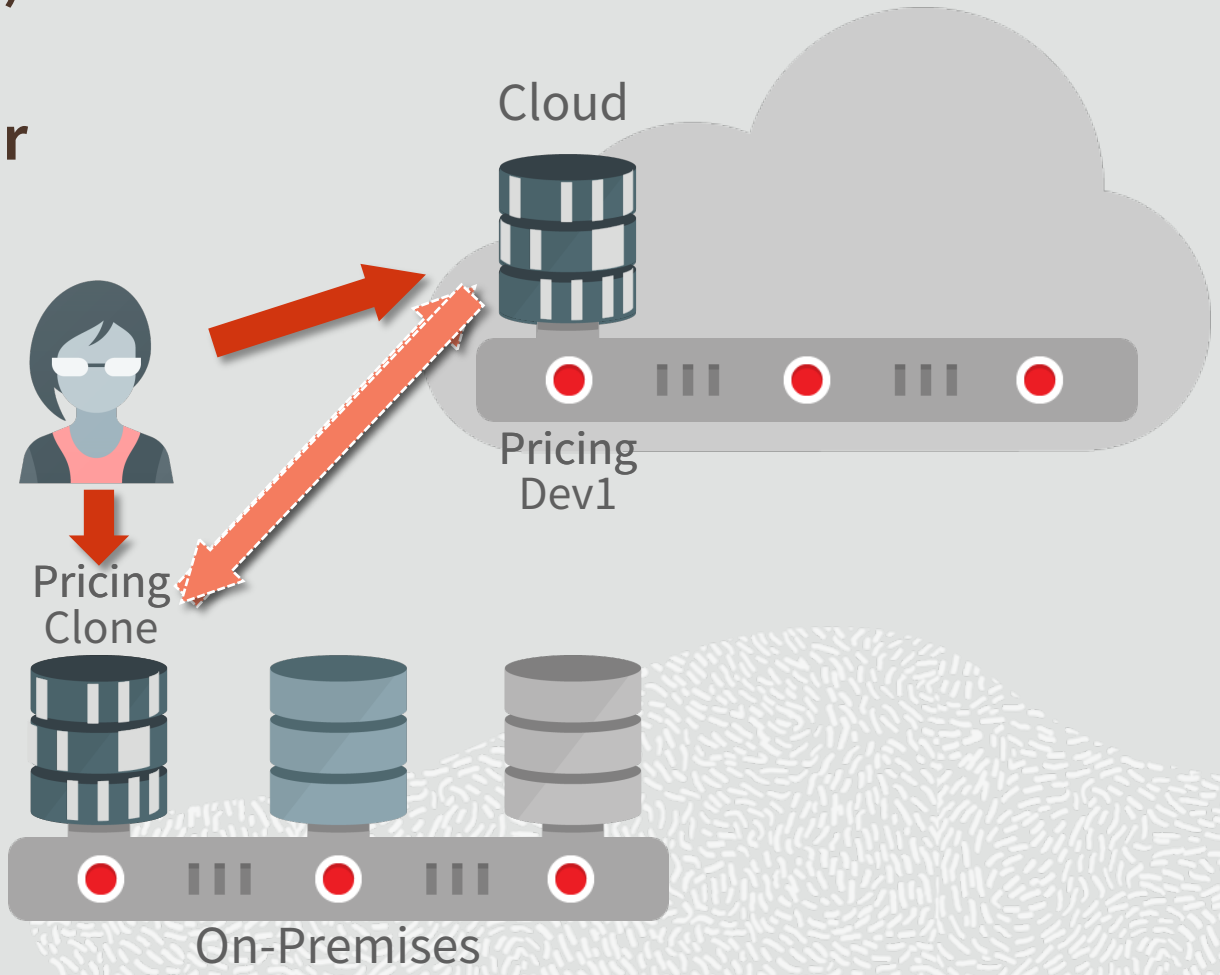
リフレッシュ可能クローン - PDB Switch Over

計画停止時にPDB ロールを切替える事ができる

構文例：

```
ALTER PLUGGABLE DATABASE
   REFRESH MODE MANUAL
   FROM Pricing@cdb2_datalink
   SWITCHOVER;
```

※
インフラの計画停止時にクローン側PDBに対し
残トランザクションを適用(リフレッシュ)後、
主PDBとしてロール切替えする事が可能



Active Data Guard とは本質的に違う!

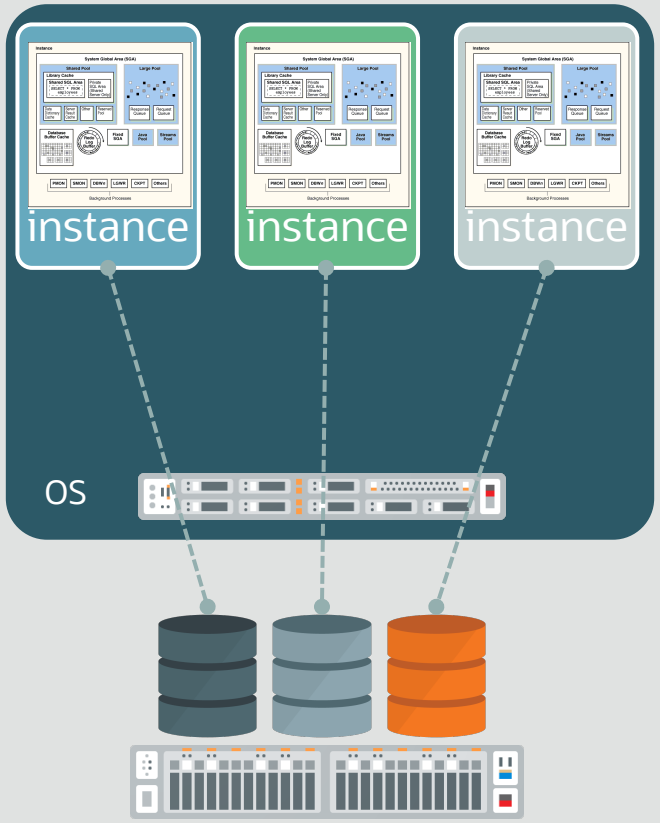
- ADGはデータを提供するサービスだけでなくデータも守る
- 本機能の対象の範囲はPDBのみ

Program agenda

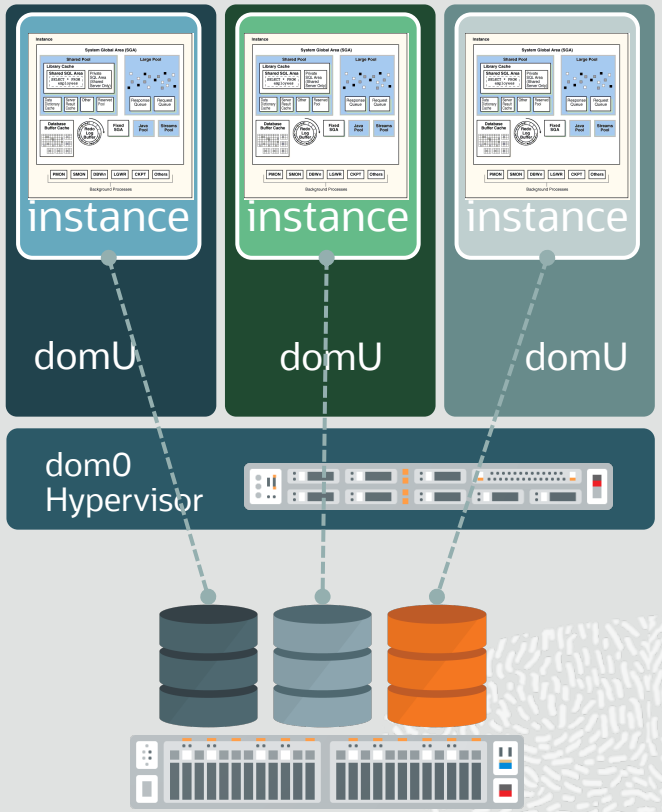
1. Oracle Database Technology Night #32 を振り返る
2. マルチテナント関連の新機能(18c / 19c)
3. 統合・集約におけるリソース分離性
4. アプリケーション・コンテナと実装例
5. まとめ

統合・集約におけるリソース分離性 – 集約パターン

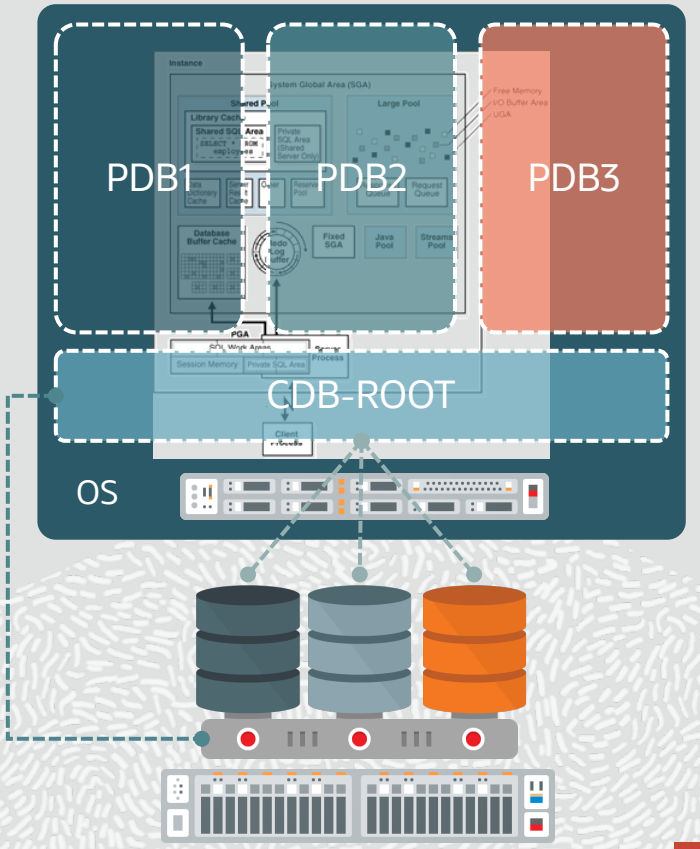
インスタンス共存



仮想化環境（Exadata VM）

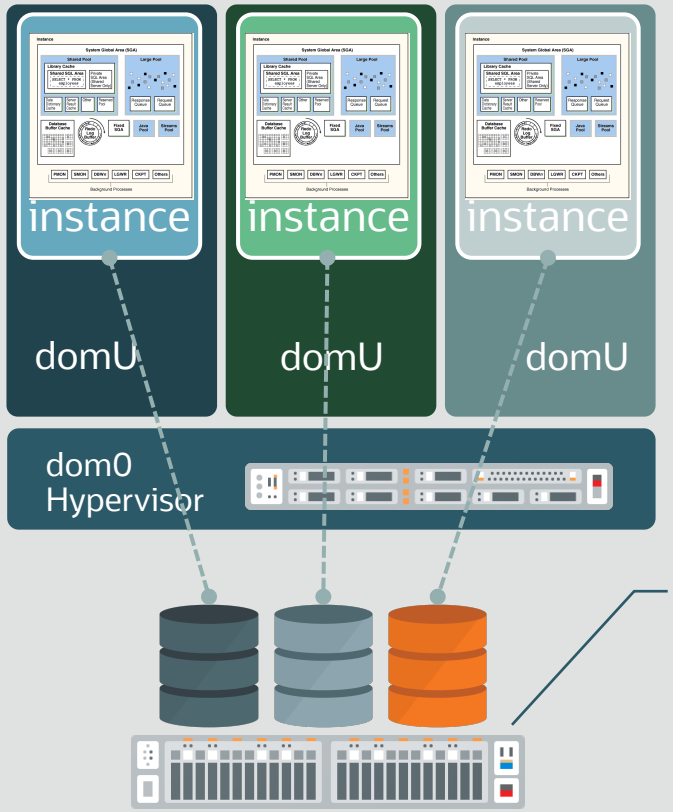


マルチテナント



統合・集約におけるリソース分離性 – VMとマルチテナント

仮想化環境 (Exadata VM)



OSレベルで管理・セキュリティ面を分離できる

OS単位での可搬性もアリ

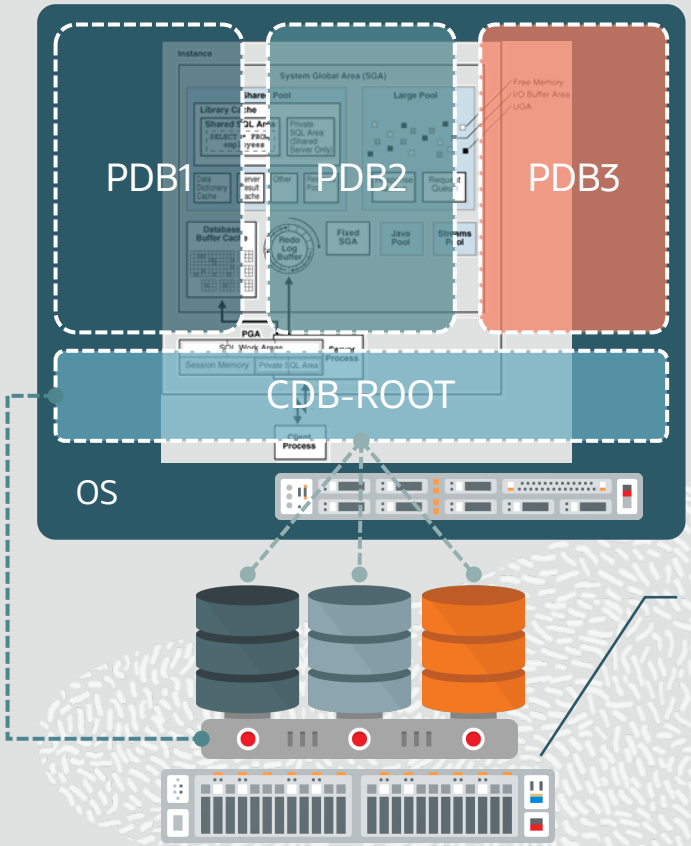
各DB間でのリソース調整は VM(OS) や Hypervisor で管理される

EE Option の差異はOK

分離性・高

物理的に全て重複

マルチテナント



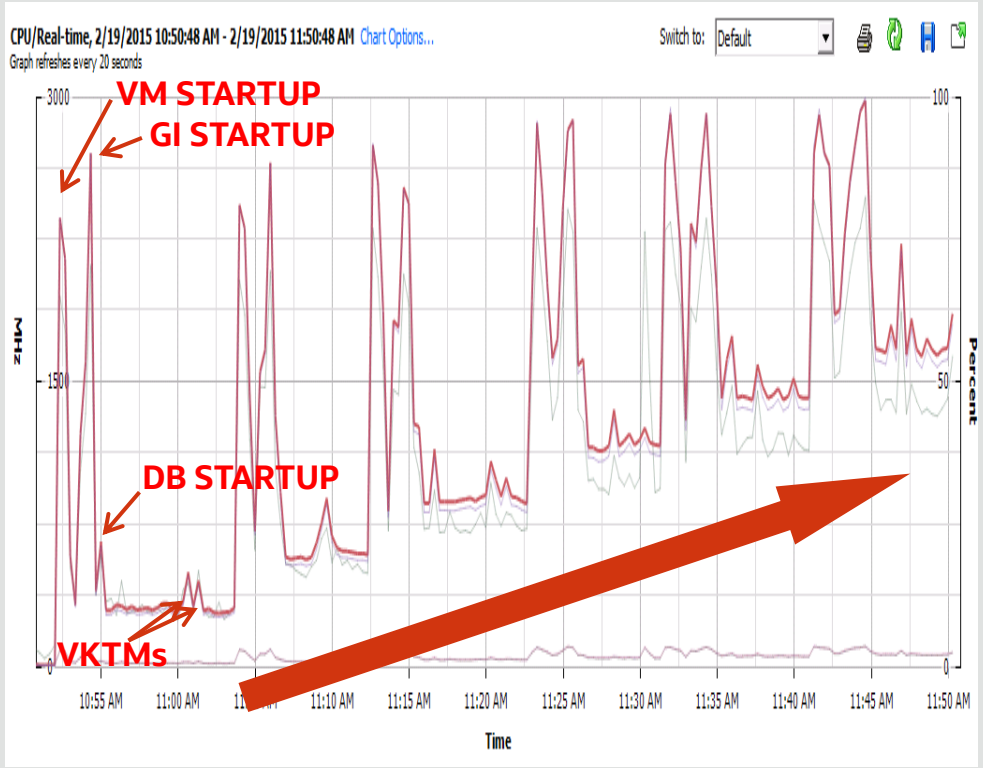
スキーマ分離型のDB統合と比較して統合が容易
可搬性に加え一元管理も容易になる
(環境一元化の必要性アリ)

各DB間でのリソース調整は Oracle および、Oracle Resource Manager で管理可能

柔軟/協調・高

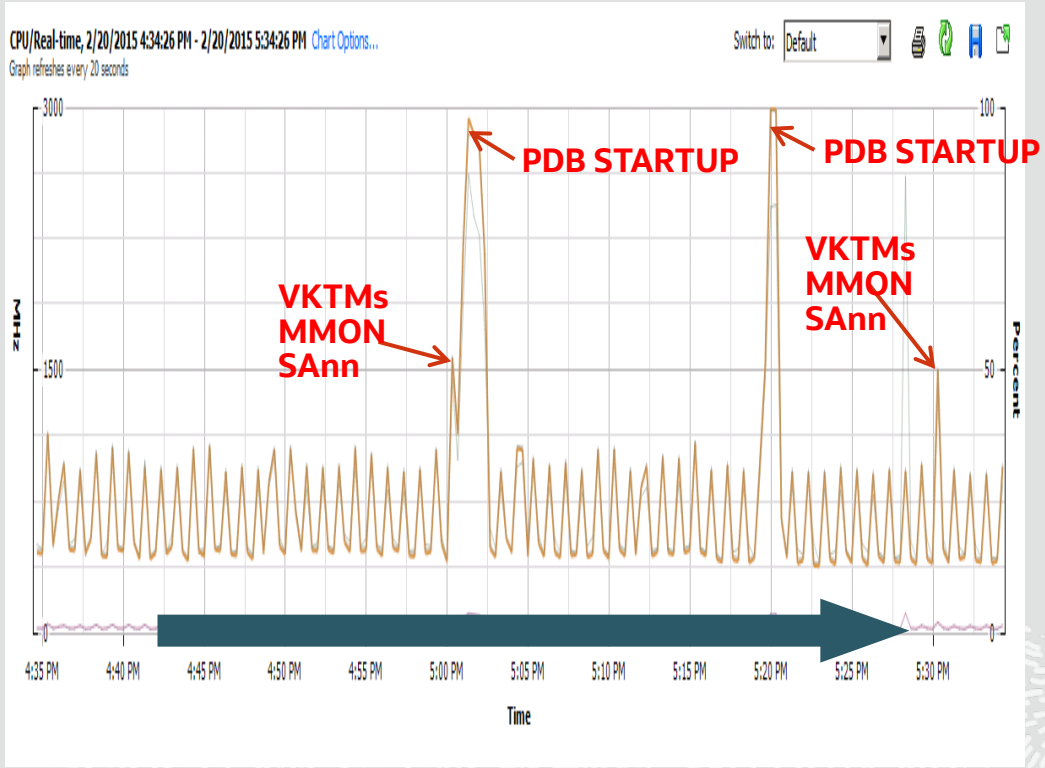
物理的な重複なし

【参考資料】– 集約と性能 : 仮想マシン/マルチテナント比較



non-CDB VM

- VMが起動するたびにCPU使用率が高くなっている
- 6VMを起動した時点で50%以上を消費している



PDB CDB

- PDBの起動する前と後の平均CPU使用率に大きな差はない
- 10 PDB以上起動させた場合でもCPU使用率は低く保てている

【参考資料】– DBの集積比較 - 252 non-CDB / 252 PDB

	集積スループット	平均応答時間	メモリ使用量	Storage IOPS
252 non-CDB	72,600 tps	6.7ミリ秒	1702MB	271,400
252 PDB	130,300 tps	9.9ミリ秒	208MB	131,200
比較	+80%	+3ミリ秒	1/8	1/2

- スループットが80%向上
- IOPSが1/2に低減
- バックグラウンドプロセス処理の集約効果
- 1DBあたり1.5GBのメモリ容量節約
- バックグラウンドプロセス数が1/92に低減

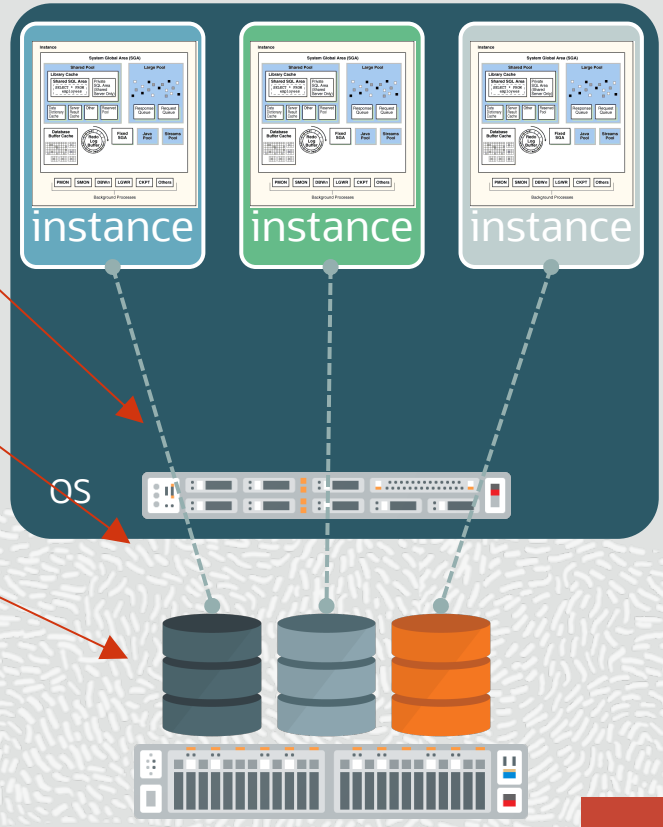
リソースにおけるバックグラウンドプロセスの稼働比率が高い

小さなI/O要求が大量に発生し、結果として大きな待機となる

強制的なコンテキスト・スイッチの発生及びスレッド移行が多発

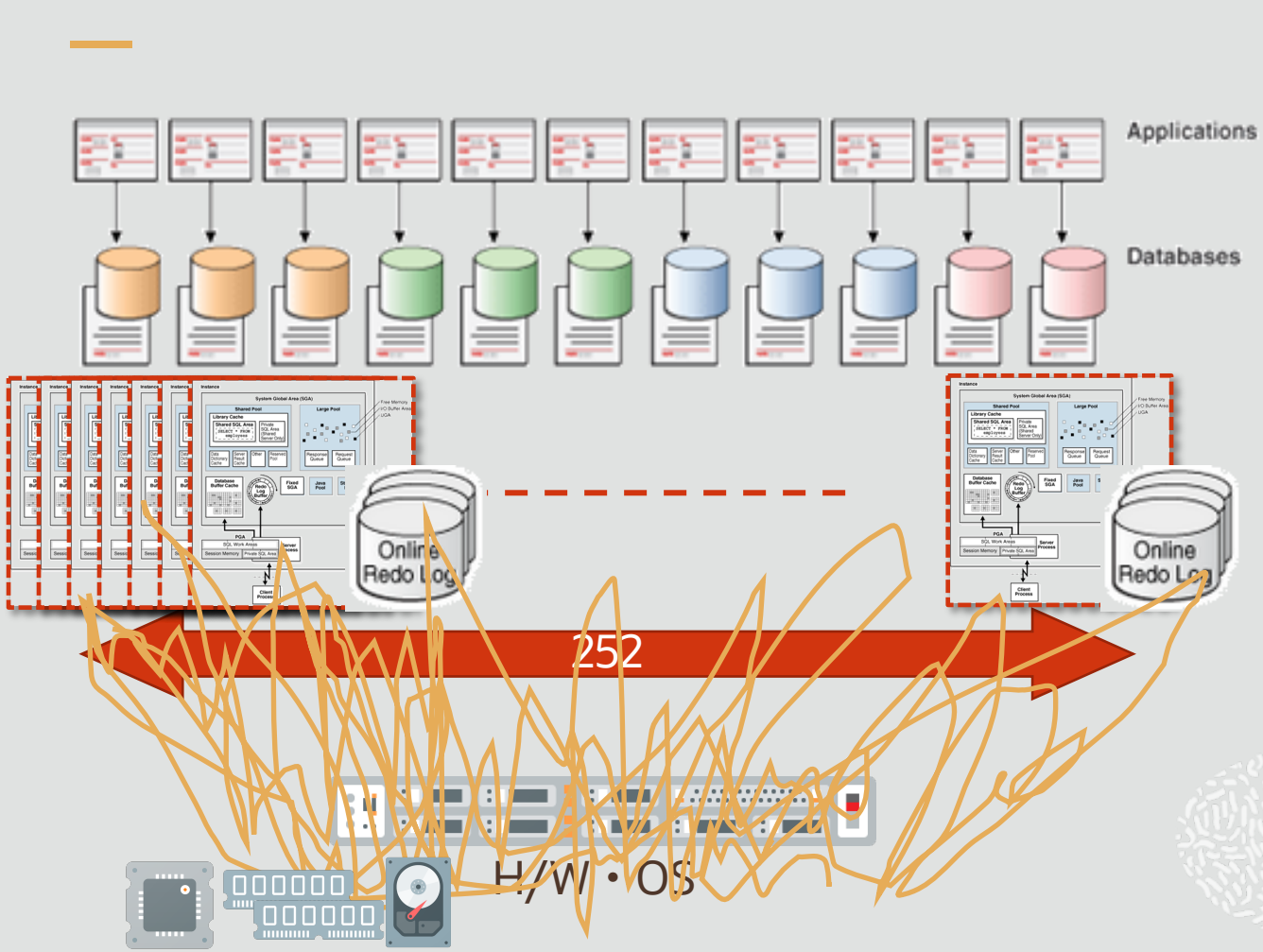
注意：
前提としてREDOログのIOPS(処理能力)は252non-CDBと、252PDB(+CDB-ROOT)において同等とする

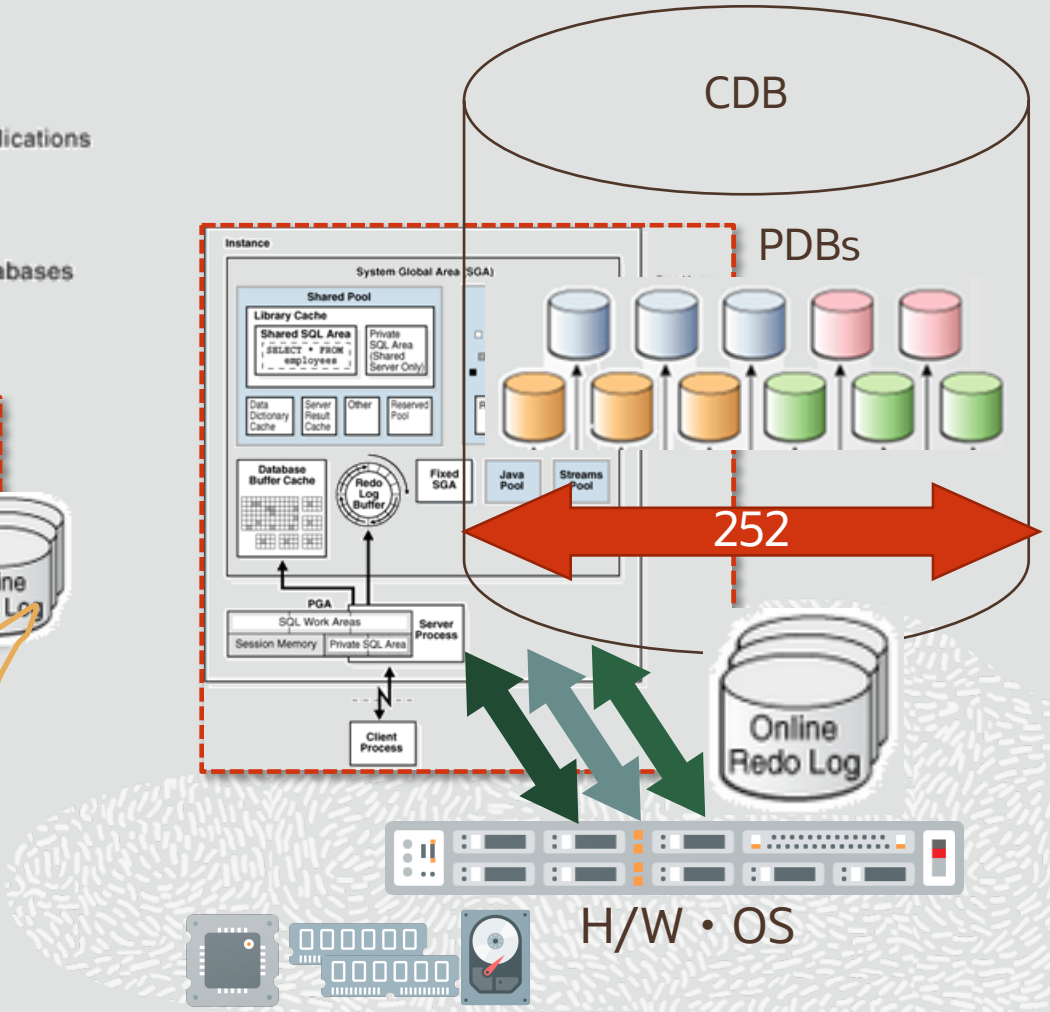
インスタンス共存



出典: White Paper: Oracle SuperCluster T5-8 Oracle Multitenant データベース統合の効率に関する事例
http://www.oracle.com/technetwork/jp/database/multitenant/learn-more/oraclemultitenantt5-8-final-2185108-ja.pdf

【参考資料】– DBの集積比較 - 252 non-CDB / 252 PDB





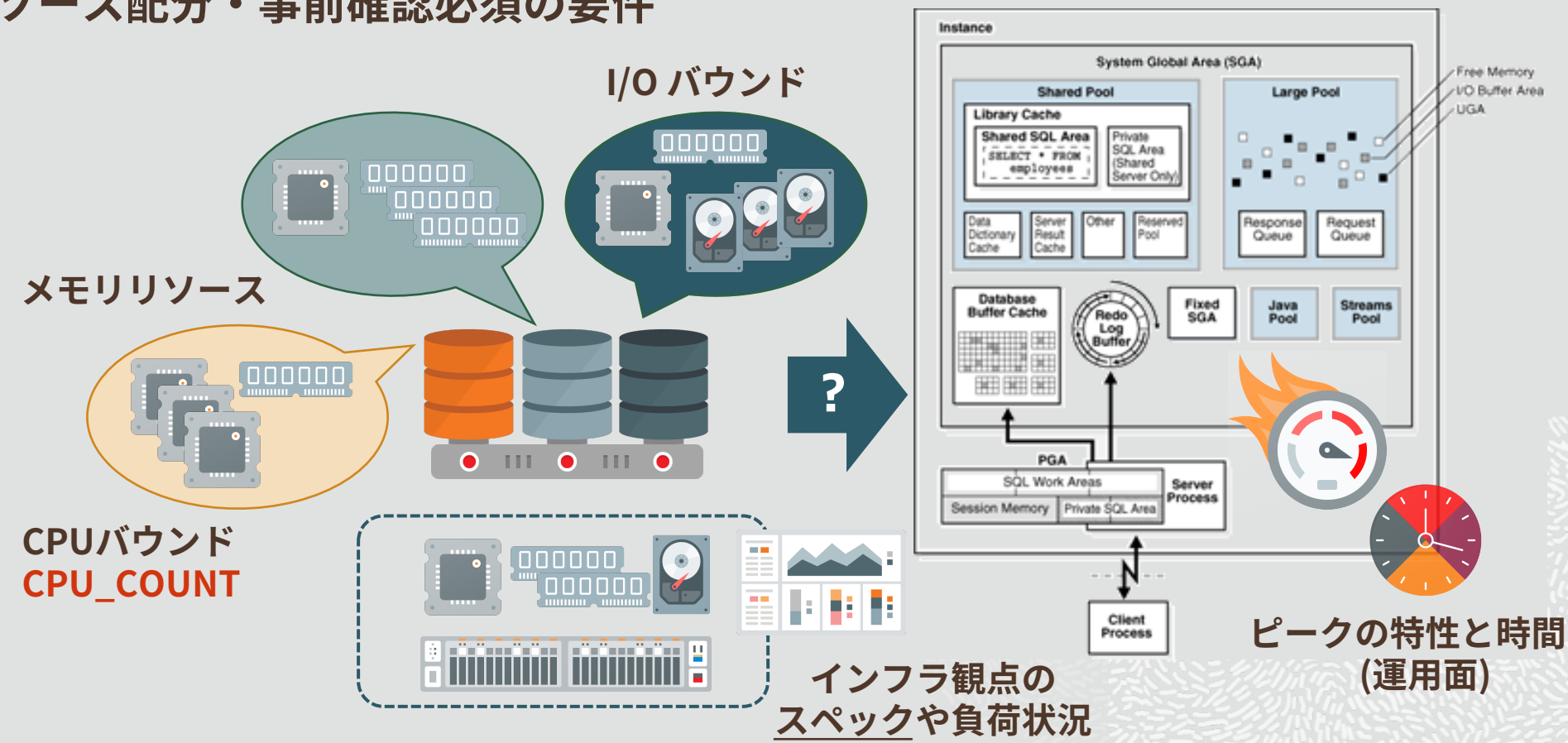
Tech Night #32 より - マルチテナント環境を理解する

A-7: 環境の共有と分離 - PDB単位で設定可能な初期化パラメータ

```
SQL> select name  from v$parameter where ispdb_modifiable = 'TRUE' order by name;

NAME
--------------------
db_cache_size
max_iops
max_mbps
max_pdbs
max_string_size
open_links
optimizer_
parallel_
pdb_file_name_convert
pdb_lockdown
pdb_os_credential
pga_aggregate_limit
pga_aggregate_target
session_cached_cursors
sessions
sga_min_size
sga_target
shared_servers
skip_unusable_indexes
smtp_out_server
sort_spatial_vector_acceleration
sql92_security
sql_trace
……
……
sqltune_category
star_transformation_enabled
statistics_level
temp_undo_enabled
timed_os_statistics
timed_statistics
undo_management
undo_retention
undo_tablespace
utl_file_dir
workarea_size_policy
xml_db_events

193行が選択されました。(R19.4)
->FALSEは253行
```

統合・集約におけるリソース分離性

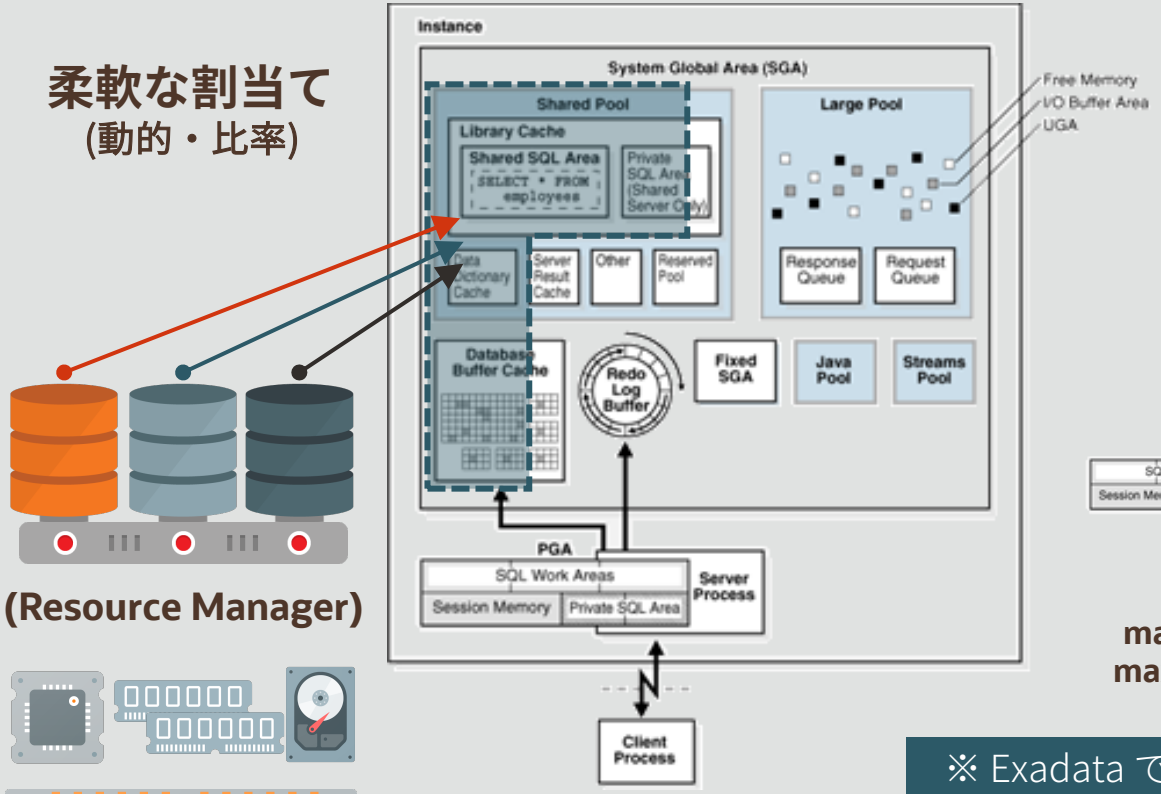
リソース配分・事前確認必須の要件



Tech Night #32 より - マルチテナント環境を理解する

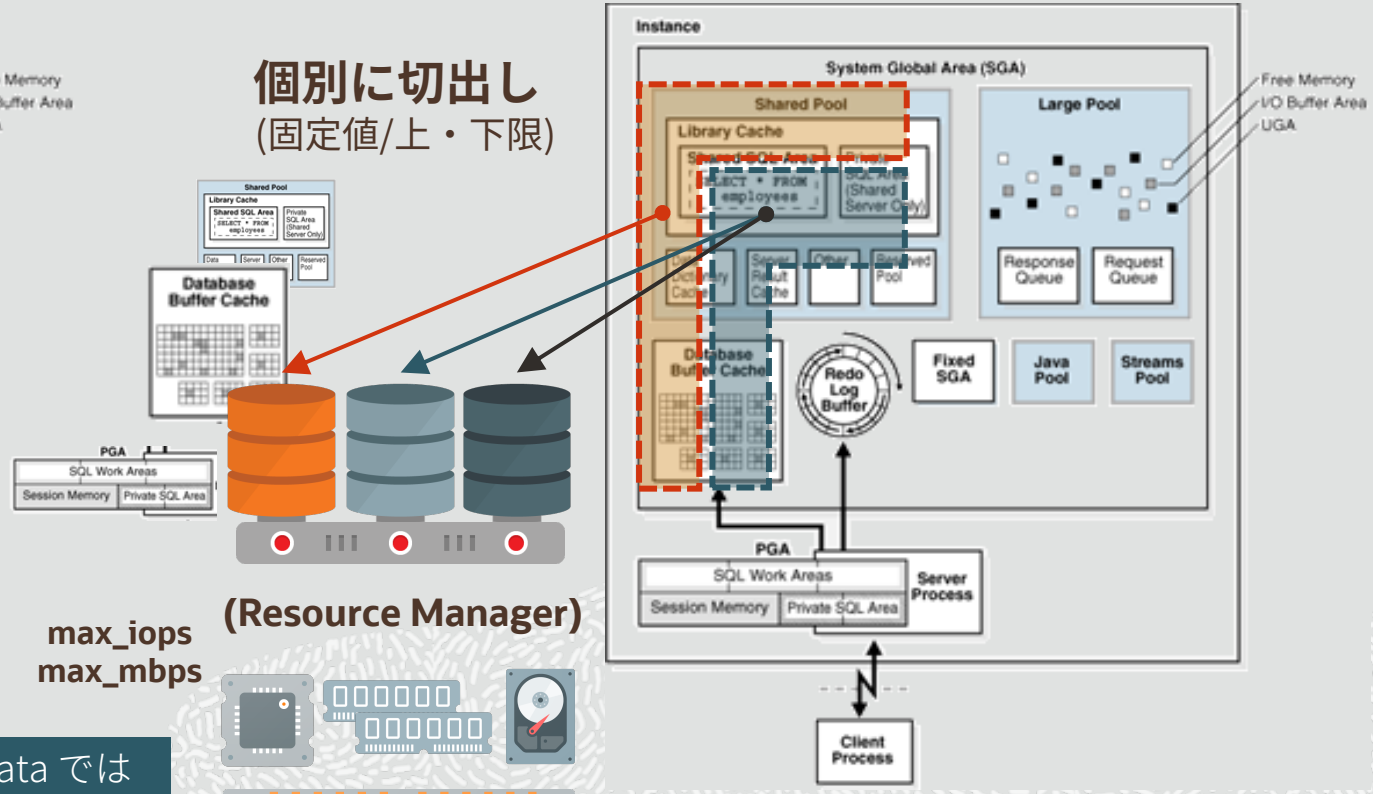
A-7: 環境の共有と分離 – ガイドラインとして

柔軟な割当て
(動的・比率)



(Resource Manager)

個別に切出し
(固定値/上・下限)



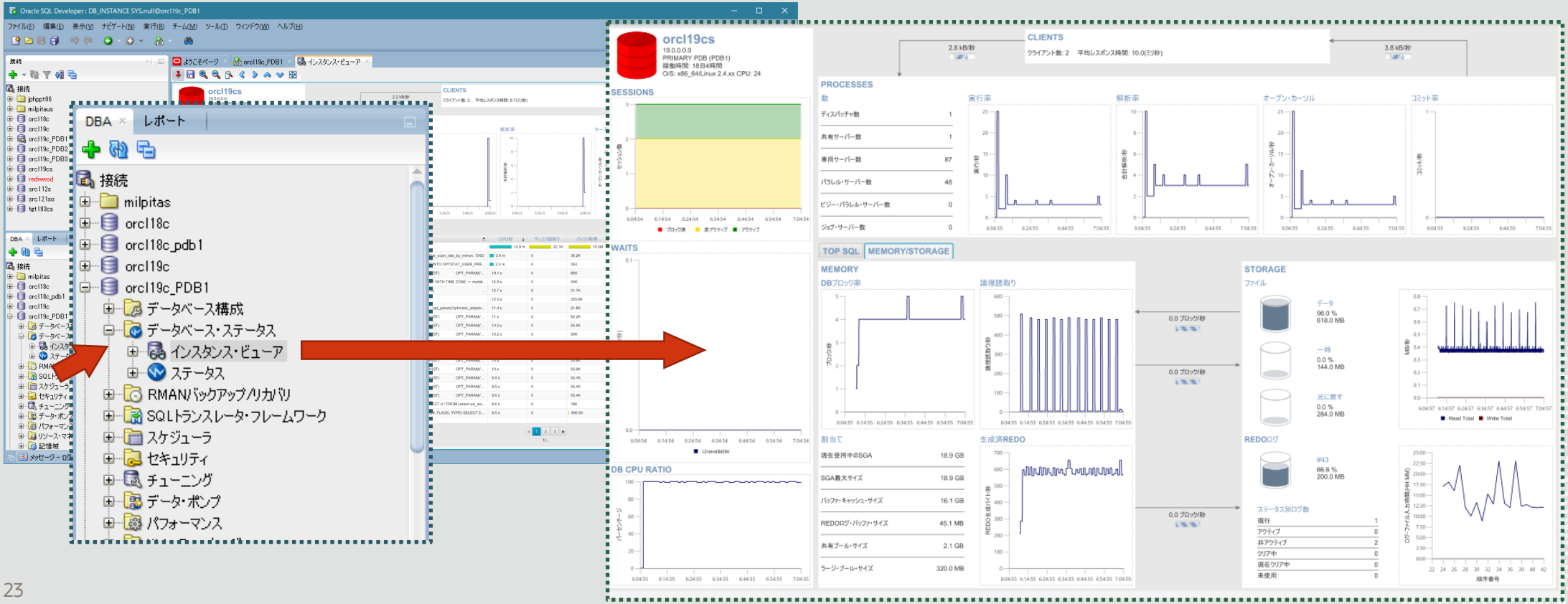
max_iops
max_mbps

(Resource Manager)

※ Exadata では
IORMと連携

統合・集約におけるリソース分離性

リソース配分・事前確認必須の要件 – SQL Developer – データベース・ステータス



統合・集約におけるリソース分離性

リソース利用状況の把握 – Doc ID 2254288.1

マルチテナント・データベースのPDBs 間のメモリ使用量
(SGA とPGA 両方) を制御及び監視する方法 - 12.2 新機能

```
SELECT
r.CON_ID, p.PDB_NAME, r.SGA_BYTES, r.PGA_BYTES, r.BUFFER_CACHE_BYTES, r.SHARED_POOL_BYTES
FROM V$RSRCPDBMETRIC r, CDB_PDBS p
WHERE r.CON_ID = p.CON_ID;
```

	CON_ID	PDB_NAME	SGA_BYTES	PGA_BYTES	BUFFER_CACHE_BYTES	SHARED_POOL_BYTES
1	3	PDB1	992208528	0	247533856	744674672
2	6	PDB3	763518200	0	311524064	451994136
3	5	PDB2	604546812	0	192767076	411779736

- SGA_BYTES - このPDBによるSGAの現在の使用状況(バイト)
- BUFFER_CACHE_BYTES - このPDB によるバッファ・キャッシュの現在の使用状況(バイト)
- SHARED_POOL_BYTES - このPDB による共有プールの現在の使用状況(バイト)
- PGA_BYTES - このPDB によるPGA の現在の使用状況(バイト)

統合・集約におけるリソース分離性

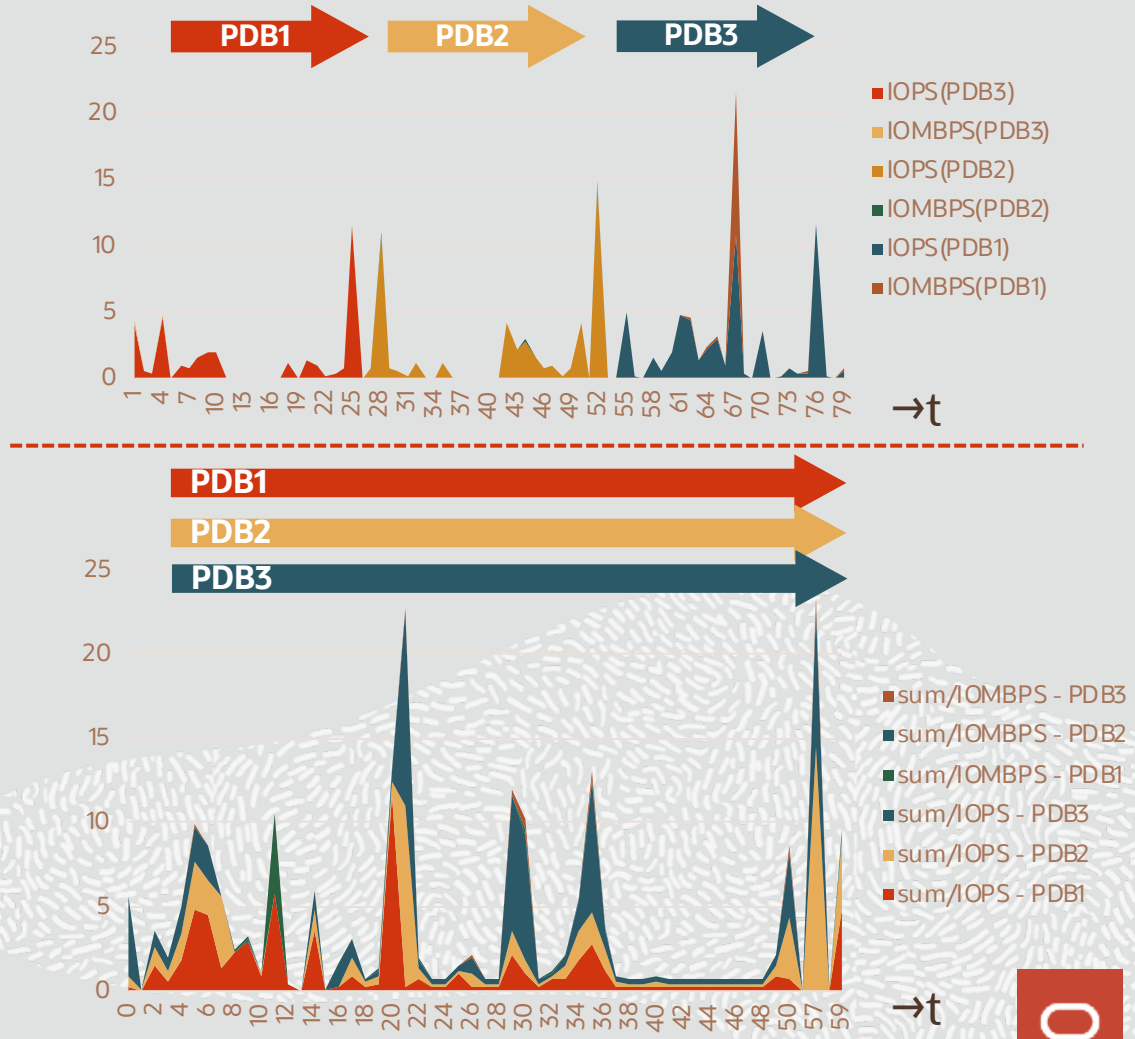
リソース利用状況の把握 – Doc ID 2254288.1

マルチテナント・データベースのPDBs 間のメモリ使用量
(SGA とPGA 両方) を制御及び監視する方法 - 12.2 新機能

```
SELECT
    r.CON_ID,p.PDB_NAME,
    TO_CHAR(r.BEGIN_TIME,'YYYY/MM/DD HH24:MI:SS'),
    TO_CHAR(r.END_TIME  , 'YYYY/MM/DD HH24:MI:SS'),
    r.SGA_BYTES, r.PGA_BYTES, r.BUFFER_CACHE_BYTES,
    r.SHARED_POOL_BYTES, r.iops, r.iombps
FROM V$RSRCPDBMETRIC_HISTORY r, CDB_PDBS p
WHERE r.CON_ID = p.CON_ID;
```

https://docs.oracle.com/cd/F19136_01/refrn/V-RSRCPDBMETRIC_HISTORY.html#GUID-63468705-52A9-4639-96EA-2323B22124BF

IOPS/IOMBPS推移を抽出した
サンプルのグラフ化



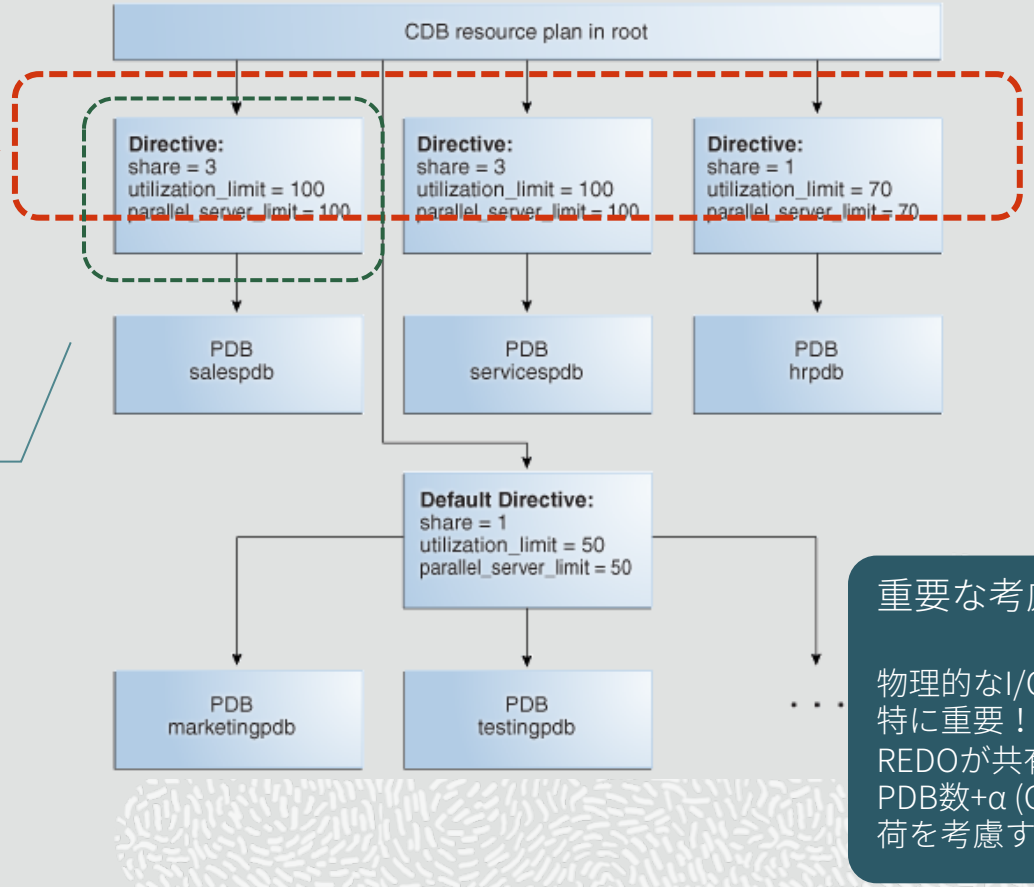
統合・集約におけるリソース分離性

マルチテナントとリソース・マネージャ連携

CDB全体のリソースから各PDBの優先順位を考慮しつつ、競合関係にある複数のPDBワークロードを管理する

PDBにおけるワークロードを管理する
通常 (非CDB環境で)のリソース管理に準ずる

中・長期的な統合／集約においてはリソース(特に I/O性能)のキャパシティ・プランニングに留意



重要な考慮点:

物理的なI/O性能への配慮は特に重要！
REDOが共有されているため、PDB数+α (CDB-ROOT) の負荷を考慮すべき

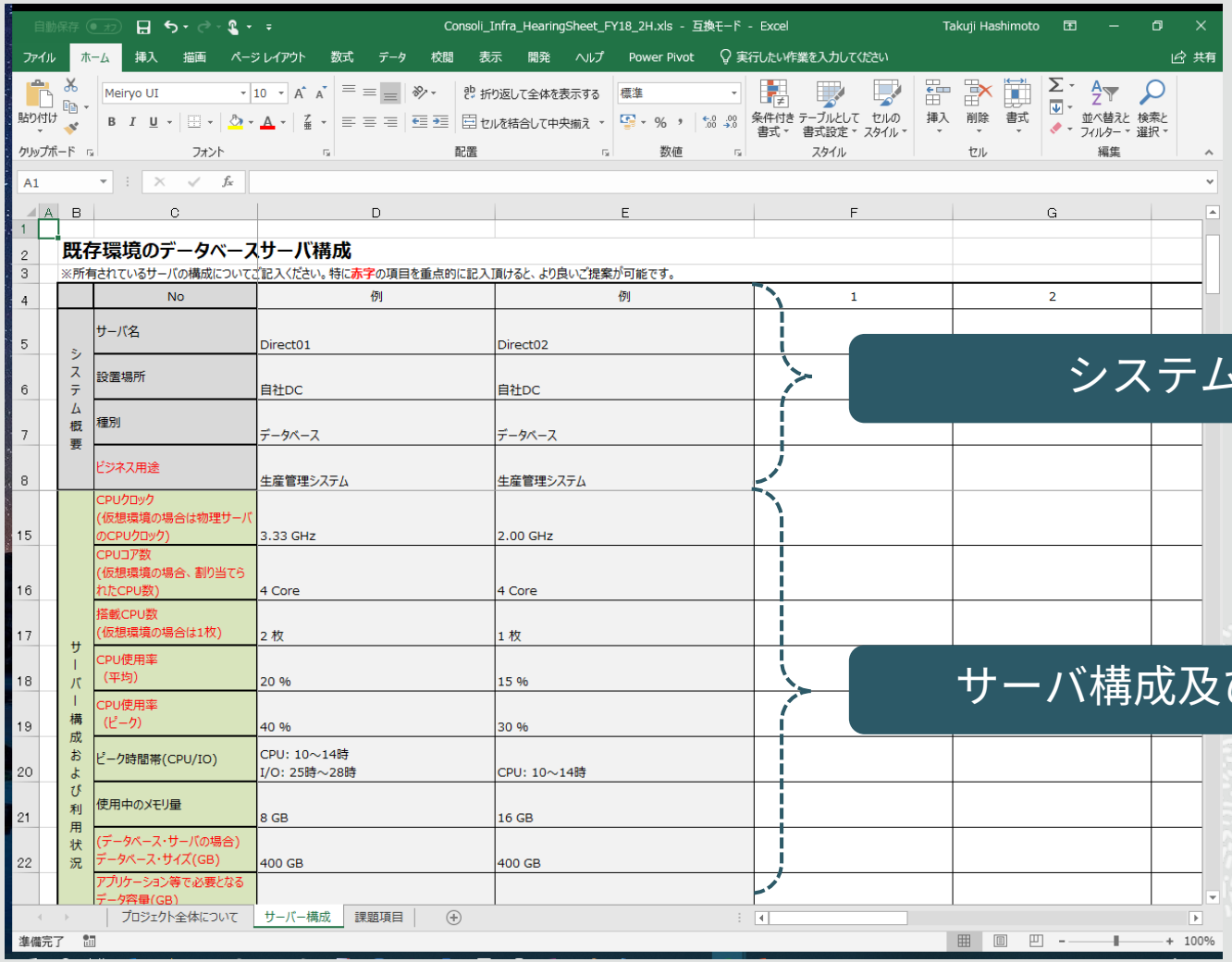
https://docs.oracle.com/cd/F19136_01/multi/using-oracle-resource-manager-for-pdbs-with-sql-plus.html#GUID-2708E76D-E18B-4586-920A-BD4B904AE14D

まとめ - 統合・集約におけるアセスメント項目の一例

システム概要	現行(または新規)システムの情報など概要
サーバ構成及び利用状況	H/Wモデル・スペック+ピーク時の負荷や時間帯など(OS負荷)
可用性対策・要件	Active-Passive / クラスタ / DR構成など (MAAレベル)
セキュリティ	CDB-ROOT と PDBのセキュリティ / 暗号化
運用・監視・管理	リソース最適化 / バックアップ / パッチ&アップグレード

まとめ - 統合・集約におけるアセスメント項目の一例

サンプル



既存環境のデータベースサーバ構成

※所有されているサーバの構成についてご記入ください。特に赤字の項目を重点的に記入頂けると、より良いご提案が可能です。

	No	例	例	1	2
システム概要	サーバ名	Direct01	Direct02		
	設置場所	自社DC	自社DC		
	種別	データベース	データベース		
	ビジネス用途	生産管理システム	生産管理システム		
サーバ構成および利用状況	CPUクロック（仮想環境の場合は物理サーバのCPUクロック）	3.33 GHz	2.00 GHz		
	CPUコア数（仮想環境の場合、割り当てられたCPU数）	4 Core	4 Core		
	搭載CPU数（仮想環境の場合は1枚）	2 枚	1 枚		
	CPU使用率（平均）	20 %	15 %		
	CPU使用率（ピーク）	40 %	30 %		
	ピーク時間帯(CPU/IO)	CPU: 10～14時 I/O: 25時～28時	CPU: 10～14時		
	使用中のメモリ量	8 GB	16 GB		
	(データベース・サーバの場合)データベース・サイズ(GB)	400 GB	400 GB		
	アプリケーション等で必要となるデータ容量(GB)				

システム概要

サーバ構成及び利用状況

Program agenda

1. Oracle Database Technology Night #32 を振り返る
2. マルチテナント関連の新機能(18c / 19c)
3. 統合・集約におけるリソース分離性
4. アプリケーション・コンテナと実装例
5. まとめ

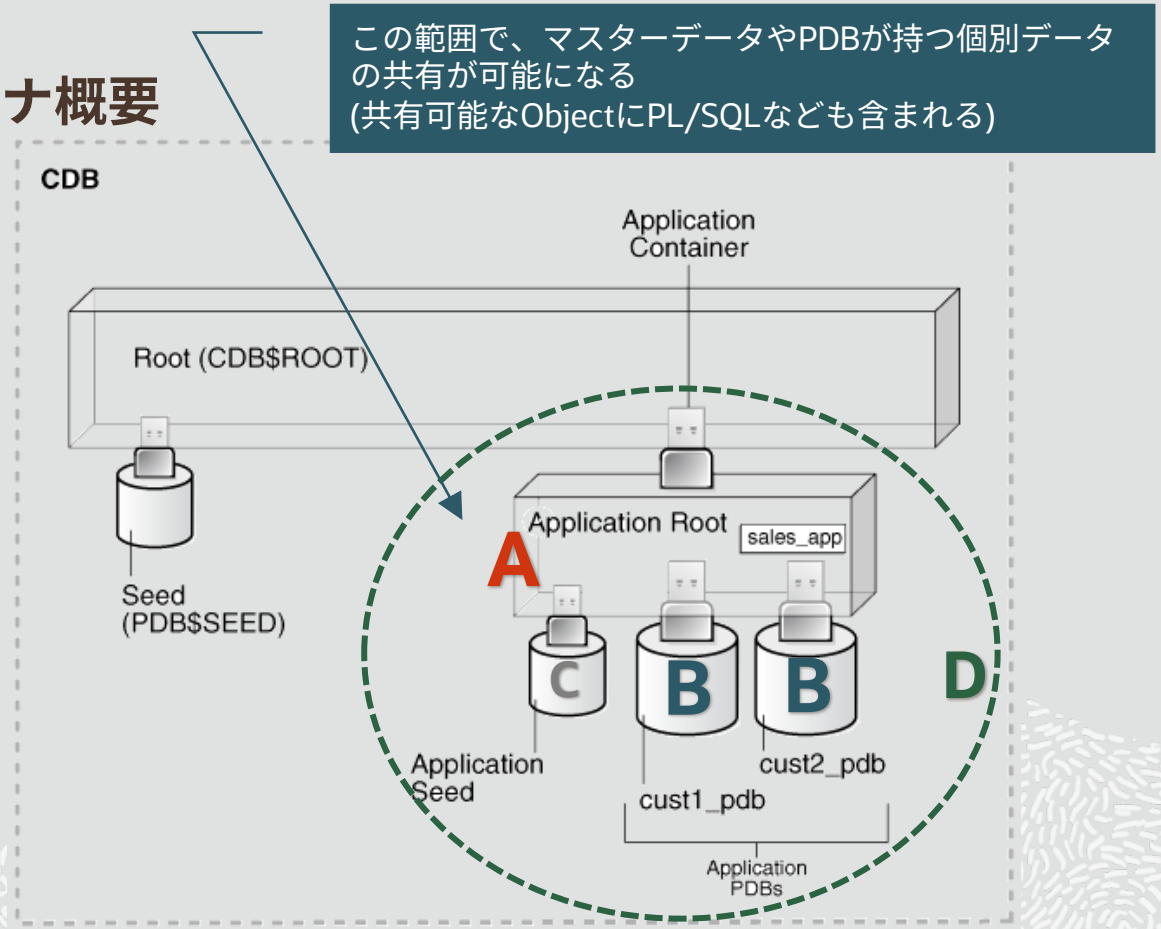
アプリケーション・コンテナと実装例

Tech Night #32 - B-4: アプリケーション・コンテナ概要

- R12.2 からの新機能
- 共通メタデータおよびデータ共有の仕組み
- バージョニングされたセットのプロビジョニング

Application Container の構成要素

名称	解説	図
アプリケーション・ルート	アプリケーション・コンテナを構成する際に必要となる起点PDBを指し、CDB-root上で稼働するAP固有の疑似的なCDBであると表現できる	A
アプリケーション・PDB	アプリケーション・コンテナ内に属するPDB	B
アプリケーション・シード	CDBシードと同様にアプリケーション・PDBを作成する場合の補助PDB	C
アプリケーション	APコンテナを作成すると有効になるオブジェクト=管理単位でバージョニングできる	D

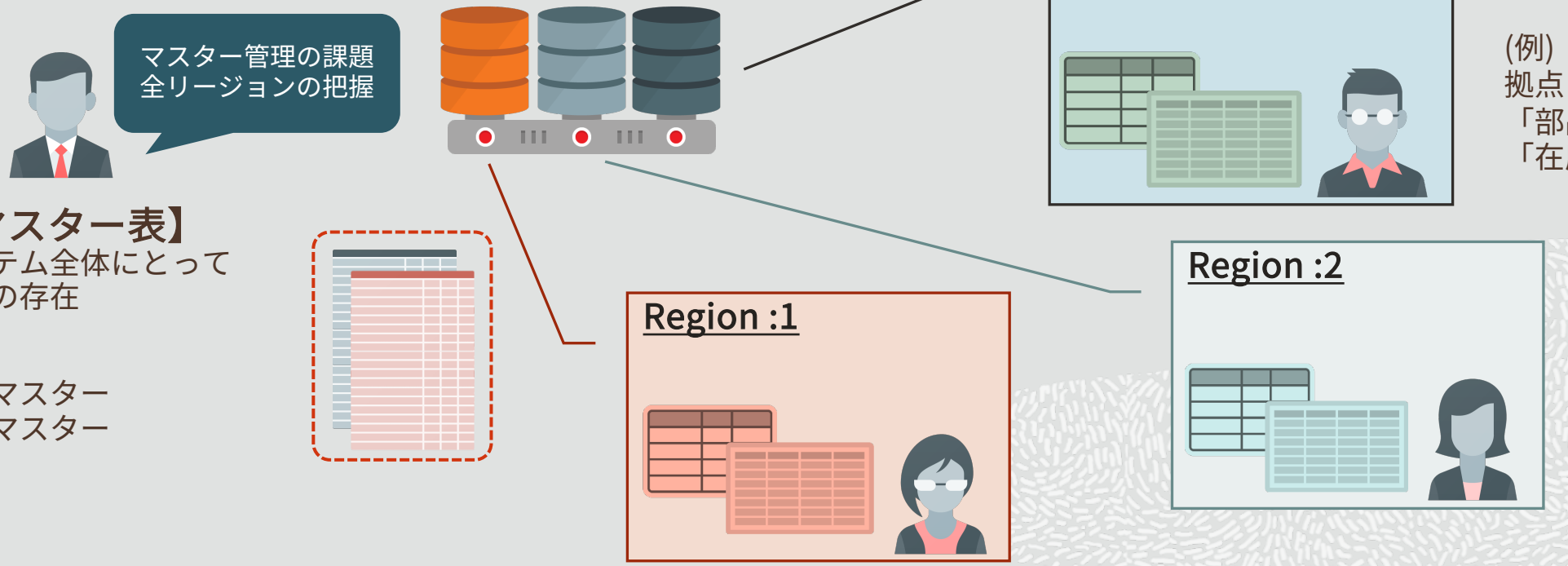


https://docs.oracle.com/cd/F19136_01/multi/overview-of-the-multitenant-architecture.html#GUID-8C2E6389-7AFF-4071-8948-E44372E66050

アプリケーション・コンテナと実装例

マルチテナントにおける”データ”統合・集約の課題

共通マスターの存在と個別データの共有・分離・連携の例



【マスター表】
システム全体にとって
唯一の存在

(例)
顧客マスター
商品マスター

【個別管理の表】

表構成(構造)は類似している
が格納されるデータは各拠
点などに依存しているもの

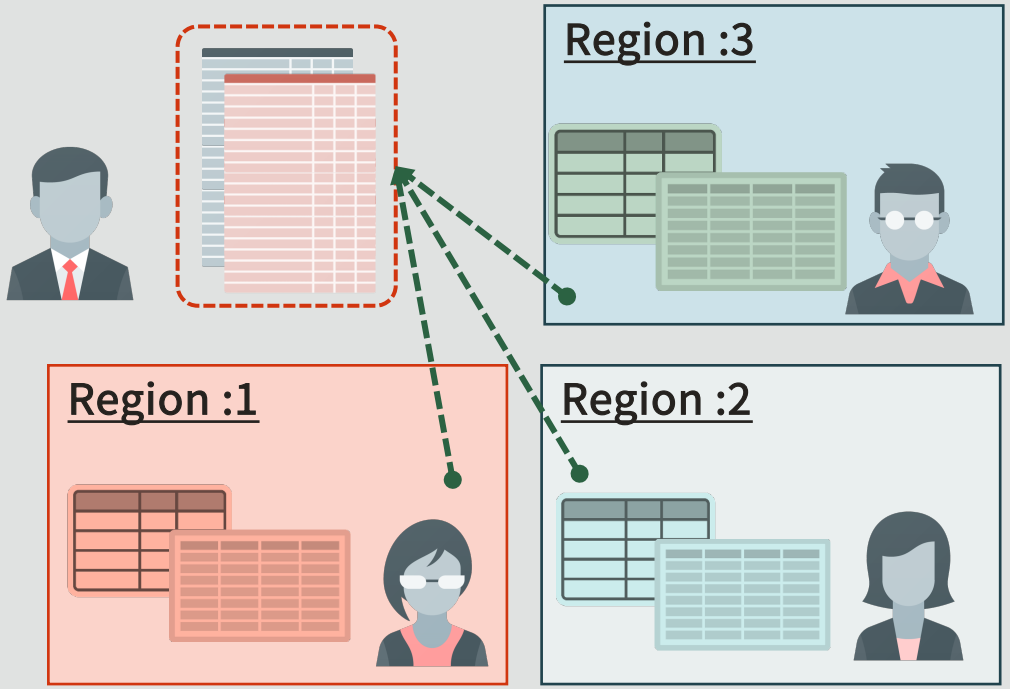
(例)
拠点ごとに内容が異なる、
「部品」「調達」「生産」
「在庫」などの実データ

アプリケーション・コンテナと実装例

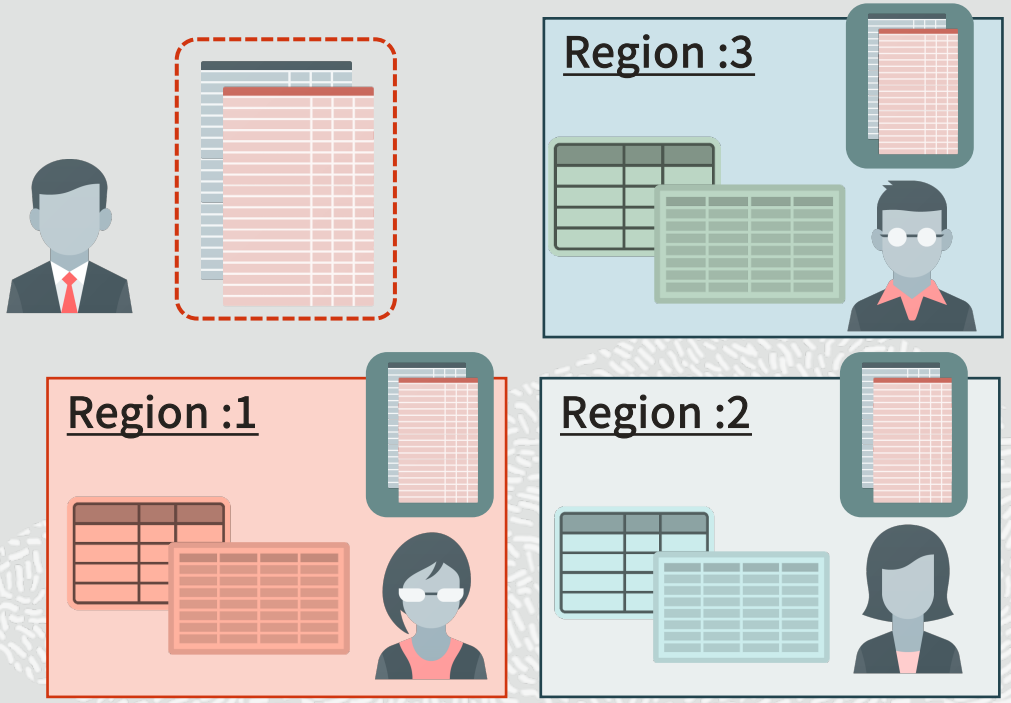
マルチテナントにおける“データ”統合・集約の課題

既存の高コストなデータの共有・分離・連携実装の例

例1) Database Link



例2) データ・レプリカ(マテリアライズド・ビュー/ GoldenGate)



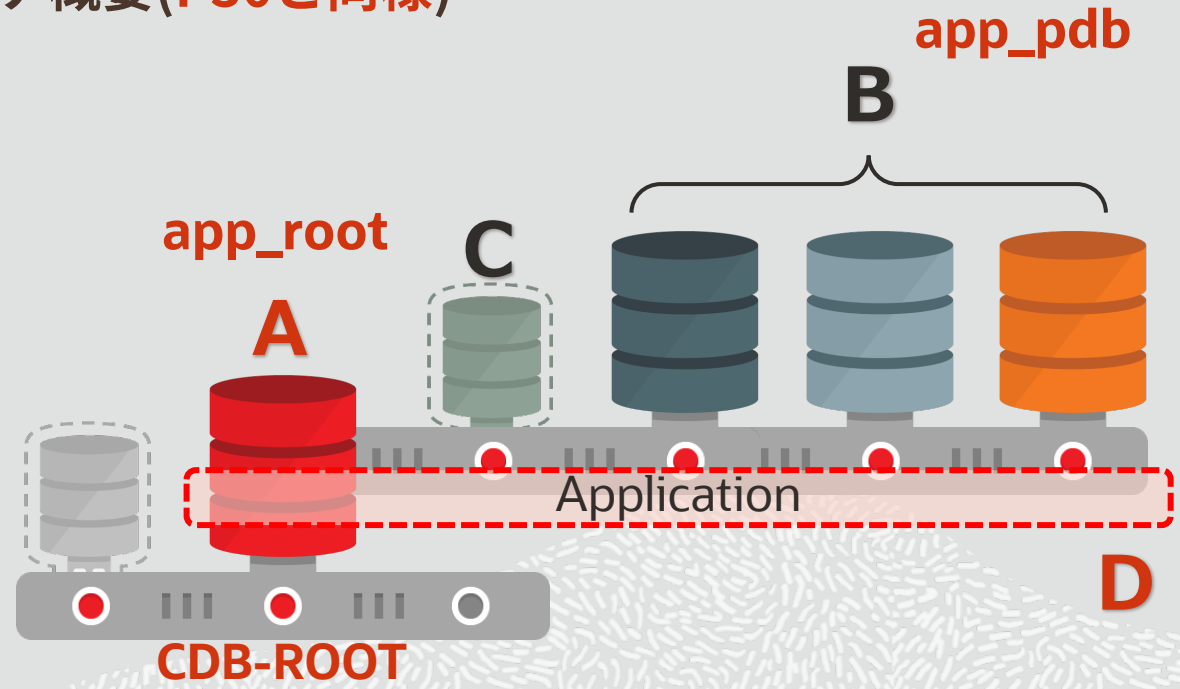
アプリケーション・コンテナと実装例 – 再確認

Tech Night #32 - B-4: アプリケーション・コンテナ概要(P30と同様)

- R12.2 からの新機能
- 共通メタデータおよびデータ共有の仕組み
- バージョニングされたセットのプロビジョニング

Application Container の構成要素

名称	解説	図
アプリケーション・ルート	アプリケーション・コンテナを構成する際に必要となる起点PDBを指し、CDB-root上で稼働するAP固有の疑似的なCDBであると表現できる	A
アプリケーション・PDB	アプリケーション・コンテナ内に属するPDB	B
アプリケーション・シード	CDBシードと同様にアプリケーション・PDBを作成する場合の補助PDB	C
アプリケーション	APコンテナを作成すると有効になるオブジェクト=管理単位でバージョニングできる	D

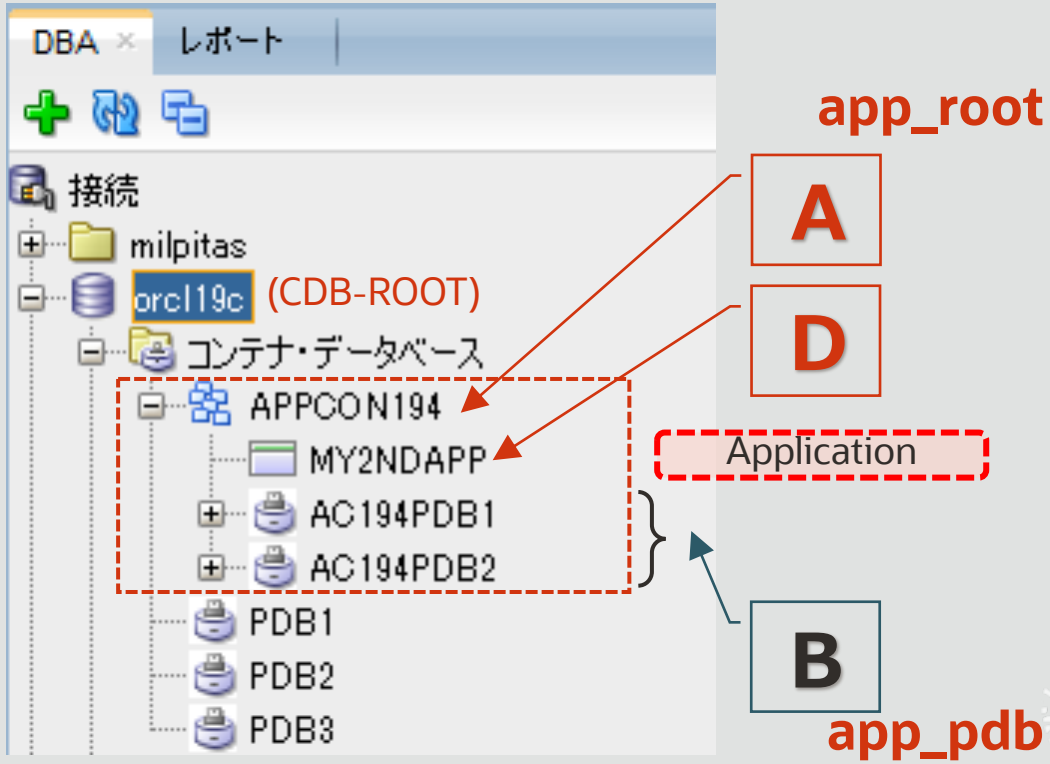


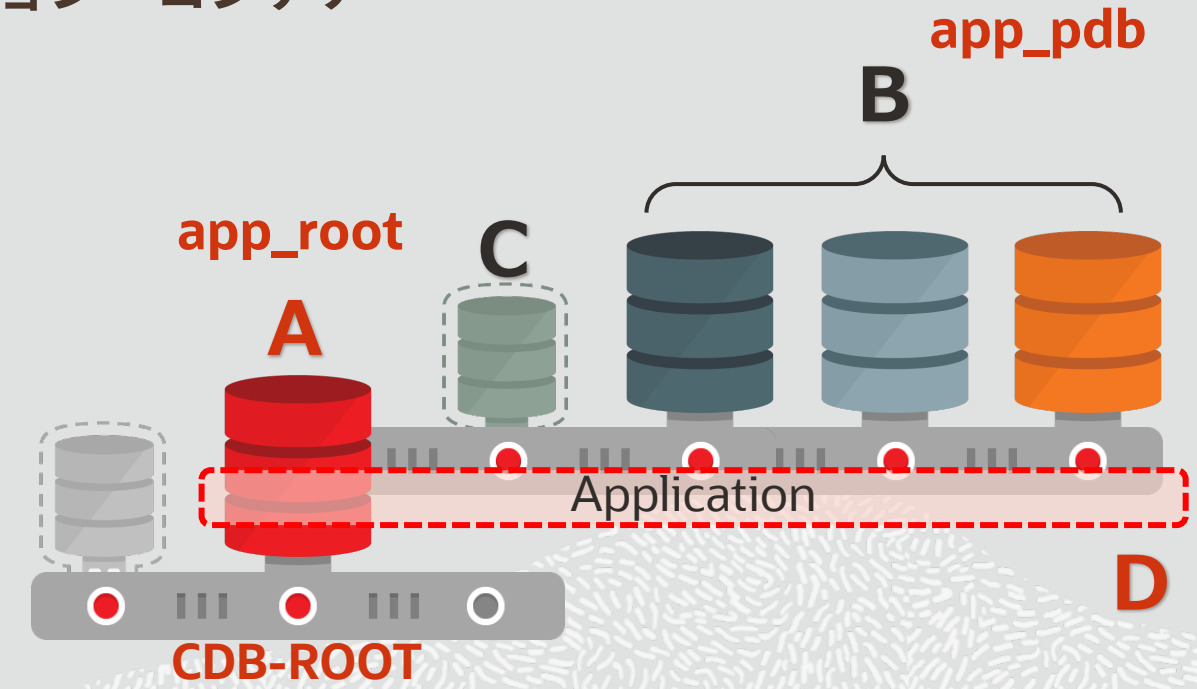
Application Container 概略図

https://docs.oracle.com/cd/F19136_01/multi/overview-of-the-multitenant-architecture.html#GUID-8C2E6389-7AFF-4071-8948-E44372E66050

アプリケーション・コンテナと実装例

参考資料: SQL Developer から見たアプリケーション・コンテナ

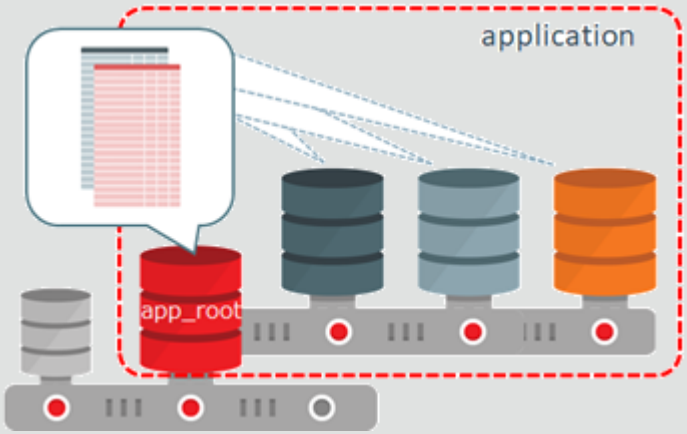




Application Container 概略図

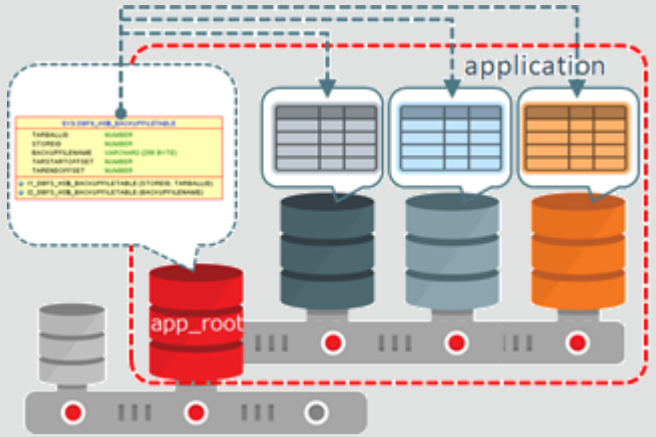
アプリケーション・コンテナと実装例 - データ連携パターン

A) Data Link



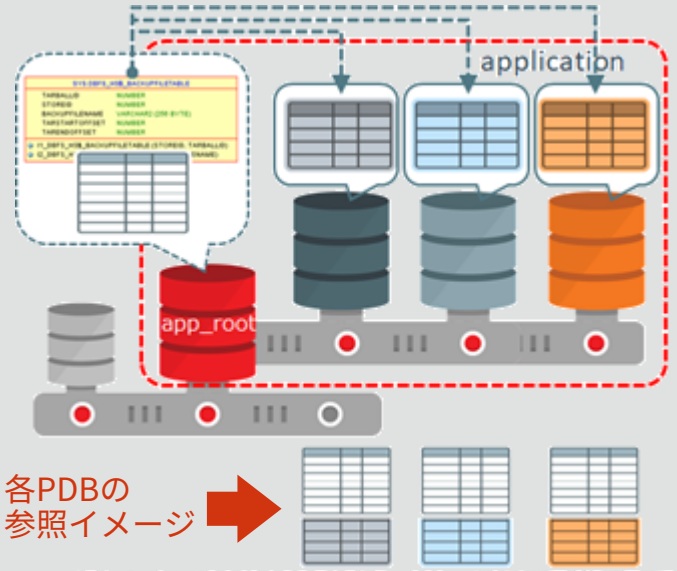
app_root に表構造(メタデータ)と実データを保持し、各アプリケーションPDBからはローカルにある表と同様に参照できる
Database Linkなどを介さず参照できるがデータ更新や構成変更はapp_rootから行う

B) Metadata Link



app_root に表構造(メタデータ)を持ち、各アプリケーションPDBで固有の実データを持ち、干渉する事なく分離される
各PDB単位で参照・更新ができ、実体は各PDBに持つ
app_rootから横断的なクエリが実行できる

C) Extended Data Link

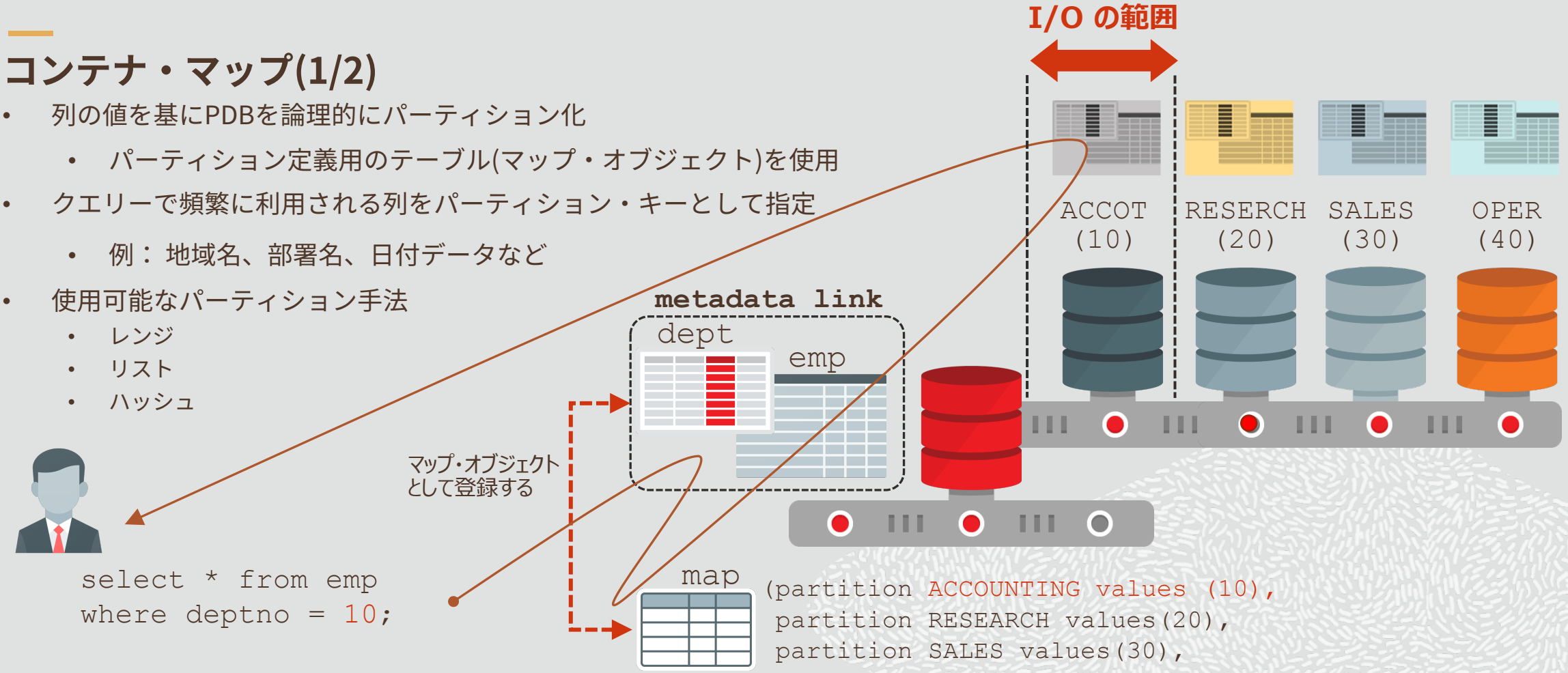


A)とB)を併せ持った構成となり、単一表として　アプリケーションPDBで検索できる
更新可能部分は各PDBにデータが格納されるMetadata Link 部分に限定される

アプリケーション・コンテナと実装例 – Metadata Link の拡張機能

コンテナ・マップ(1/2)

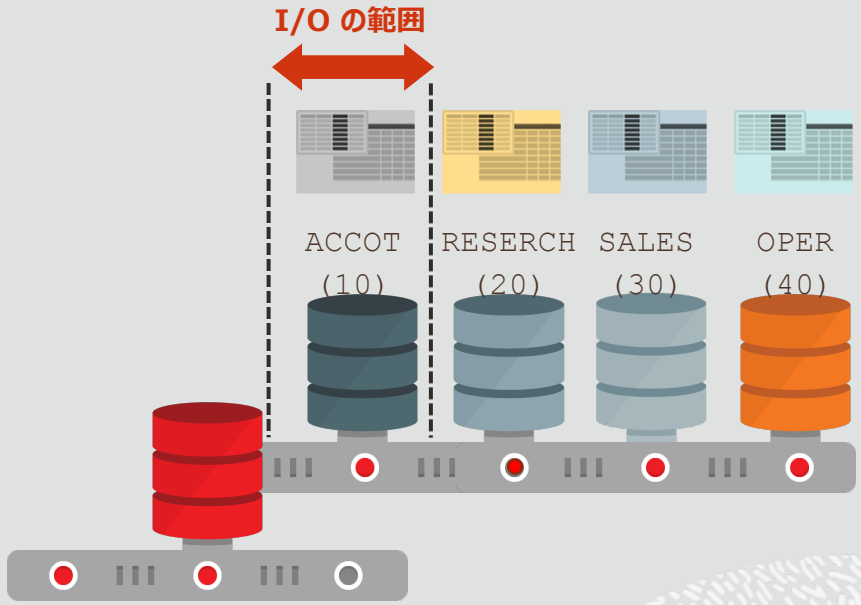
- 列の値を基にPDBを論理的にパーティション化
  - パーティション定義用のテーブル(マップ・オブジェクト)を使用
- クエリーで頻繁に利用される列をパーティション・キーとして指定
  - 例： 地域名、部署名、日付データなど
- 使用可能なパーティション手法
  - レンジ
  - リスト
  - ハッシュ



アプリケーション・コンテナと実装例 – Metadata Link の拡張機能

コンテナ・マップ(2/2)

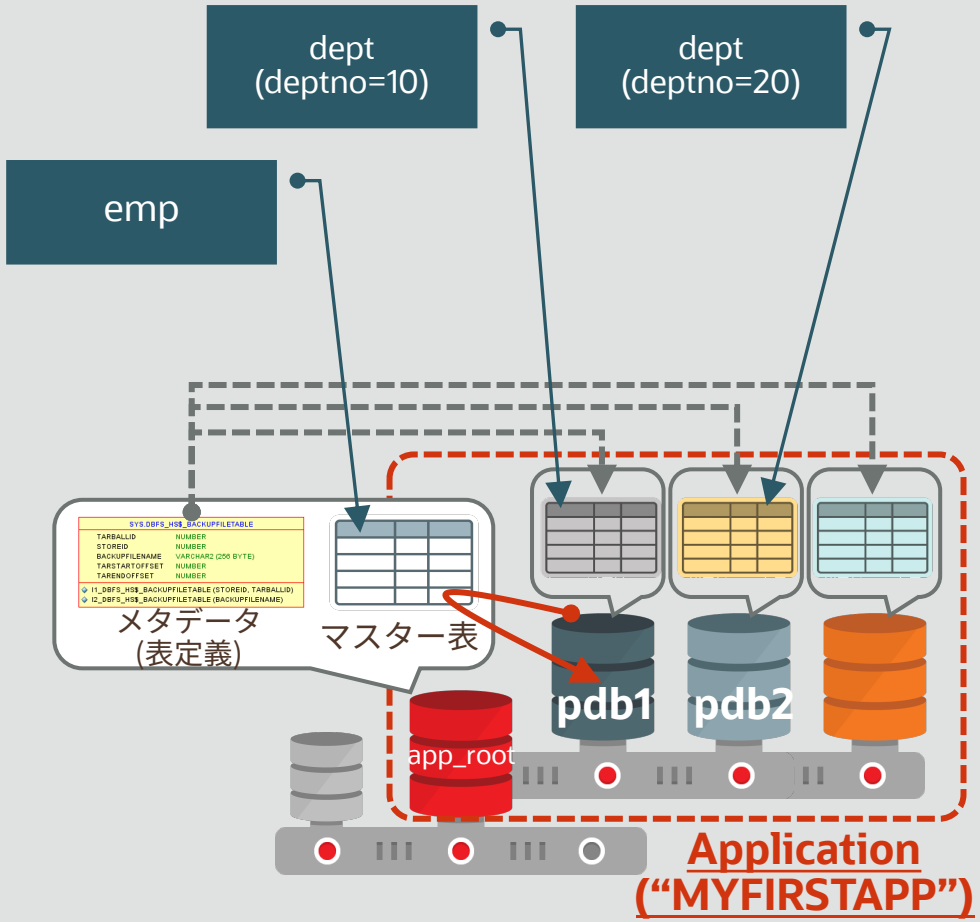
```
SQL> select * from emp where deptno = 10;
```



実行計画

```
----------------------------------------------------------
Plan hash value: 984574782
-------------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                  | Name     | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |    TQ  |IN-OUT| PQ Distrib |
-------------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |          |  1200 |   117K|     4 (100)| 00:00:01 |       |       |        |      |            |
|   1 |  PX COORDINATOR            |          |       |       |            |          |       |       |        |      |            |
|   2 |   PX SEND QC (RANDOM)      | :TQ10000 |  1200 |   117K|     4 (100)| 00:00:01 |       |       |  Q1,00 | P->S | QC (RAND)  |
|   3 |    PX PARTITION LIST SINGLE|          |  1200 |   117K|     4 (100)| 00:00:01 |     1 |     1 |  Q1,00 | PCWC |            |
|   4 |     CONTAINERS FULL        | EMP      |  1200 |   117K|     4 (100)| 00:00:01 |       |       |  Q1,00 | PCWP |            |
-------------------------------------------------------------------------------------------------------------------------------------
```

アプリケーション・コンテナと実装例 - データリンク表・メタデータリンク表の結合(1/2)



app_root [emp]

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80-12-17	800		20
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30
7566	JONES	MANAGER	7839	81-04-02	2975		20
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7788	SCOTT	ANALYST	7566	87-06-13	3000		20
7839	KING	PRESIDENT		81-11-17	5000		10
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
7876	ADAMS	CLERK	7788	87-06-13	1100		20
7900	JAMES	CLERK	7698	81-12-03	950		30
7902	FORD	ANALYST	7566	81-12-03	3000		20
7934	MILLER	CLERK	7782	82-01-23	1300		10

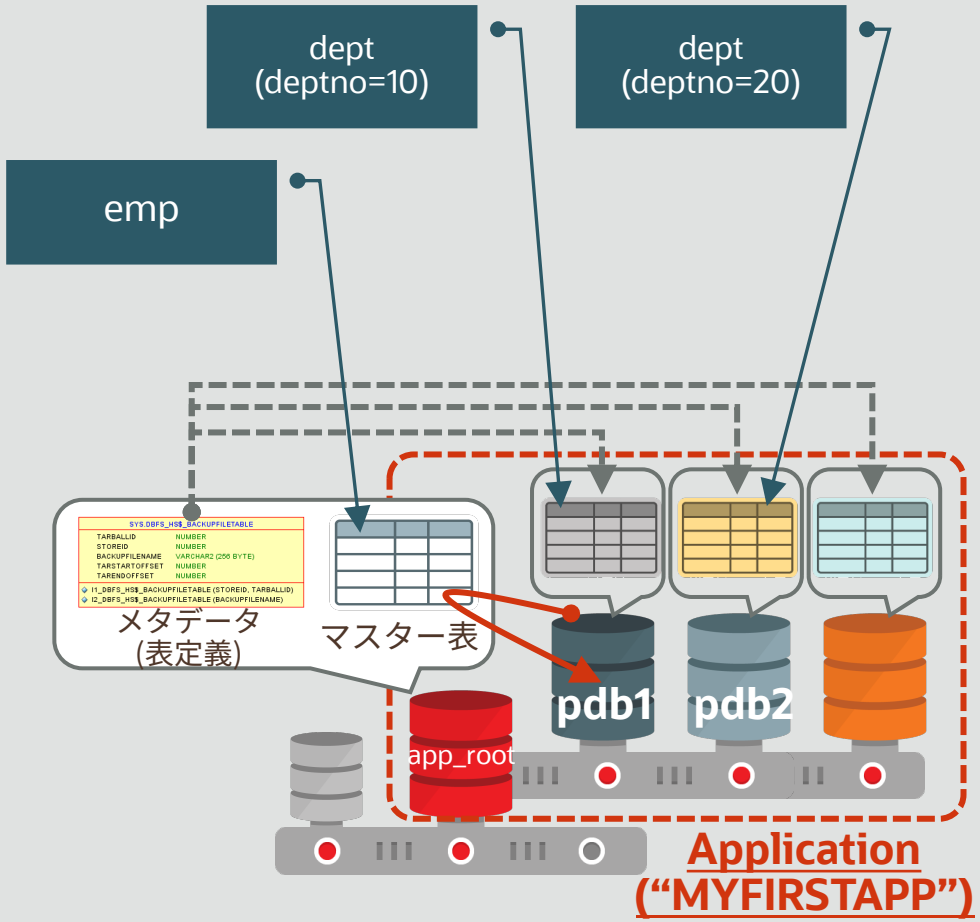
pdb1 [dept]

```
SQL> select * from dept;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
```

pdb2 [dept]

```
SQL> select * from dept;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        20 RESEARCH       DALLAS
```

アプリケーション・コンテナと実装例 - データリンク表・メタデータリンク表の結合(2/2)



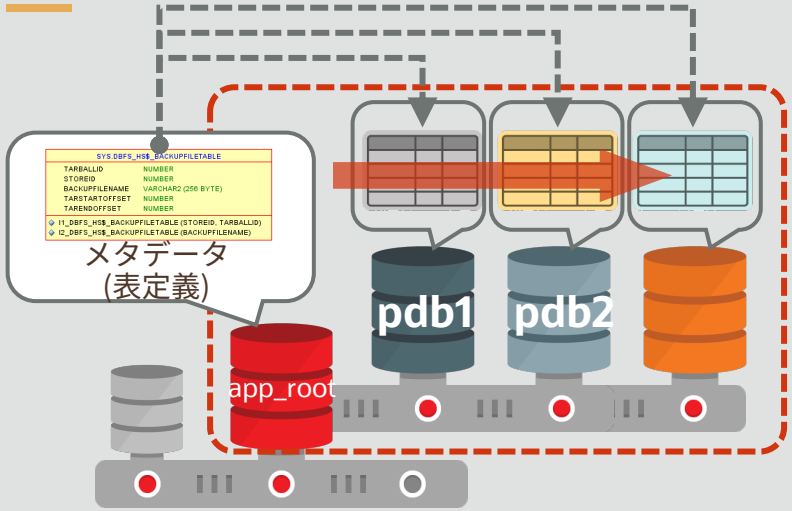
pdb1

```
SQL> select * from dept;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK

SQL> select * from emp,dept where emp.deptno=dept.deptno;
     EMPNO ENAME      JOB              MGR HIREDATE         SAL    …       LOC
---------- ---------- --------- ---------- -------- ---------- -------- -----
      7782 CLARK      MANAGER          7839 81-06-09       2450    …       NEW YORK
      7839 KING       PRESIDENT             81-11-17       5000    …       NEW YORK
      7934 MILLER     CLERK            7782 82-01-23       1300    …       NEW YORK
```

pdb2

```
SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        20 RESEARCH       DALLAS

SQL> select * from emp,dept where emp.deptno=dept.deptno;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL    …       LOC
---------- ---------- --------- ---------- -------- ---------- -------- -----
      7369 SMITH      CLERK            7902 80-12-17        800    …       DALLAS
      7566 JONES      MANAGER          7839 81-04-02       2975    …       DALLAS
      7788 SCOTT      ANALYST          7566 87-06-13       3000    …       DALLAS
      7876 ADAMS      CLERK            7788 87-06-13       1100    …       DALLAS
      7902 FORD       ANALYST          7566 81-12-03       3000    …       DALLAS
```

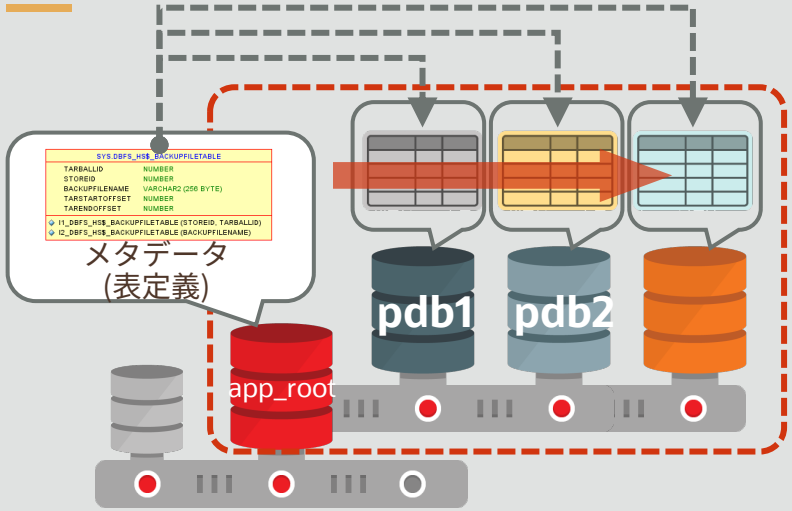
アプリケーション・コンテナと実装例 - Container 句を使用した一元的な集計例(1/2)



例えば拠点ごとの営業成績などが
PDB毎に格納されているケース

pdb1			pdb2		
2014 east	Q1	17908	2014 east	Q1	18095
2014 north	Q1	12254	2014 north	Q1	13842
2014 south	Q1	16321	2014 south	Q1	19689
2014 west	Q1	10103	2014 west	Q1	16966
2014 east	Q2	16430	2014 east	Q2	19736
2014 north	Q2	19698	2014 north	Q2	18638
2014 south	Q2	12076	2014 south	Q2	16886
2014 west	Q2	17561	2014 west	Q2	17747
2014 east	Q3	12356	2014 east	Q3	14784
2014 north	Q3	16189	2014 north	Q3	19705
2014 south	Q3	16066	2014 south	Q3	17561
2014 west	Q3	19543	2014 west	Q3	18574
2014 east	Q4	13837	2014 east	Q4	19122
2014 north	Q4	14549	2014 north	Q4	15414
2015 south	Q3	15900	2015 south	Q3	17705
	～			～	
2015 west	Q3	12837	2015 west	Q3	10126
2015 east	Q4	15720	2015 east	Q4	10959
2015 north	Q4	11986	2015 north	Q4	19306
2015 south	Q4	14754	2015 south	Q4	19026
2015 west	Q4	17219	2015 west	Q4	19379

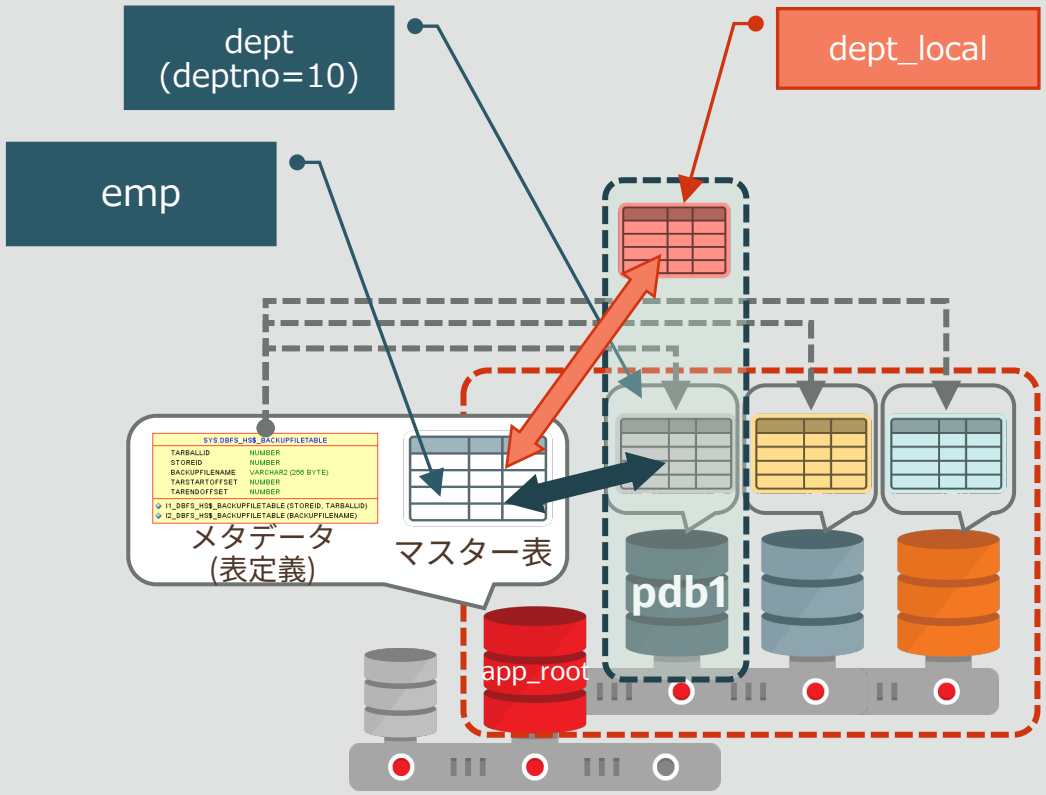
アプリケーション・コンテナと実装例 - Container 句を使用した一元的な集計例(2/2)




```
SQL> break on year skip 1
SQL> set pagesize 999
SQL> col name format a10
SQL> compute sum label "Yearly Revenue" of revenue on year
SQL> --
SQL> select sum(a.revenue) revenue, a.year, a.region, b.name
  2  from containers(sales_data) a, v$containers b
  3  where a.con_id = b.con_id
  4  group by a.year, a.region, b.name
  5  order by b.name,a.year,a.region
  6  /
```

```
   REVENUE       YEAR REGION     NAME
---------- ---------- ---------- ----------
     71737       2014 east       PDB2
     67599            north      PDB2
     70507            south      PDB2
     72882            west       PDB2
---------- **********
    282725 Yearly Rev

     54253       2015 east       PDB2
     64724            north      PDB2
     66807            south      PDB2
     50690            west       PDB2
---------- **********
    236474 Yearly Rev

     60531       2014 east       PDB1
     62690            north      PDB1
     56849            south      PDB1
     61169            west       PDB1
---------- **********
    241239 Yearly Rev

     49312       2015 east       PDB1
     53033            north      PDB1
     70349            south      PDB1
     64646            west       PDB1
---------- **********
    237340 Yearly Rev
```

アプリケーション・コンテナと実装例 – Application PDB のローカル表と結合



データリンク表とPDB内にある表の結合

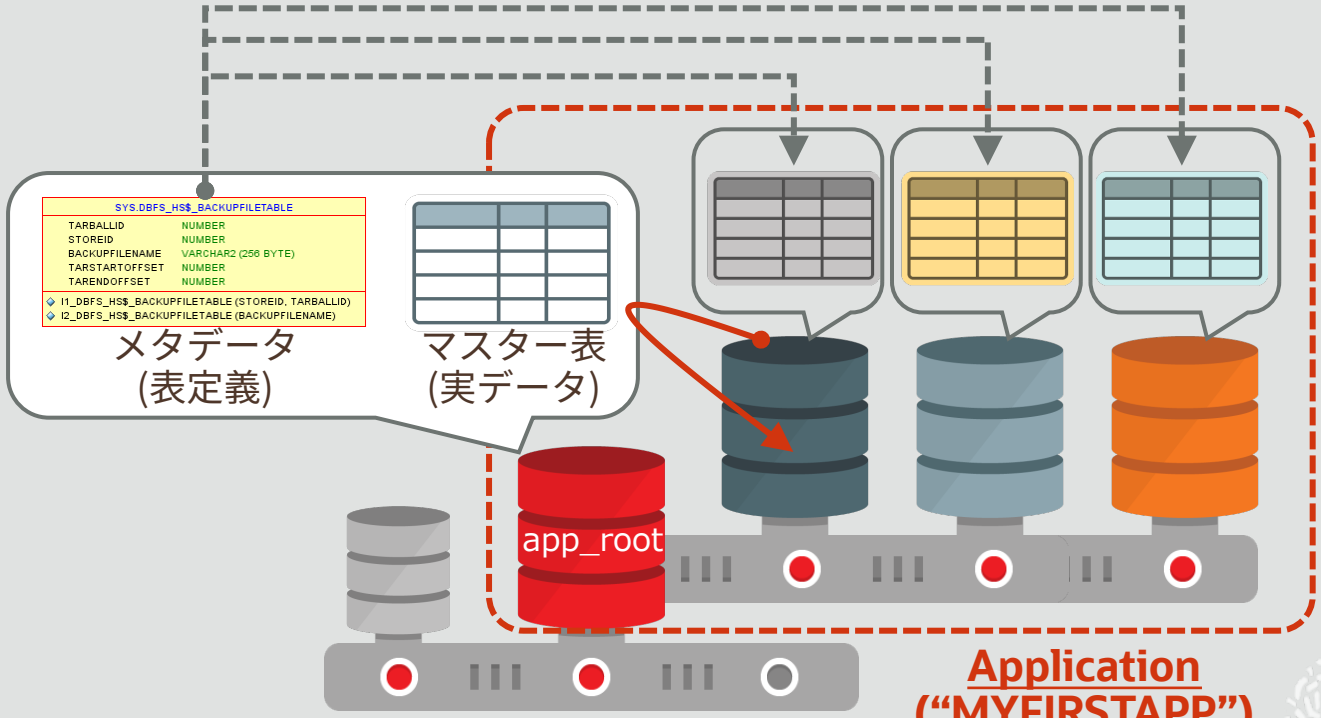
```
SQL> select * from dept_local;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     東京
        20 RESEARCH       大阪
        30 SALES          名古屋
        40 OPERATIONS     福岡
SQL> select * from emp,dept_local where emp.deptno=dept_local.deptno;
     EMPNO ENAME      JOB              MGR HIREDATE         SAL            LOC
---------- ---------- --------- ---------- -------- ---------- --------- -----
      7369 SMITH      CLERK           7902 80-12-17         800            大阪
      7499 ALLEN      SALESMAN        7698 81-02-20        1600            名古屋
      7521 WARD       SALESMAN        7698 81-02-22        1250            名古屋
      7566 JONES      MANAGER         7839 81-04-02        2975            大阪
     .....
```

データリンク表とメタデータ・リンク表の結合

```
SQL> select * from dept;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK

SQL> select * from emp,dept where emp.deptno=dept.deptno;
     EMPNO ENAME      JOB              MGR HIREDATE         SAL    …       LOC
---------- ---------- --------- ---------- -------- ---------- --------- -----
      7782 CLARK      MANAGER         7839 81-06-09        2450    …       NEW YORK
      7839 KING       PRESIDENT            81-11-17        5000    …       NEW YORK
      7934 MILLER     CLERK           7782 82-01-23        1300    …       NEW YORK
```

アプリケーション・コンテナと実装例

構築パターンと実装例



CDB の作成 (R12.2以上)

アプリケーション・コンテナの作成

アプリケーション “MYFIRSTAPP” 構成開始(ver. 1.0)

タイプ別・共有データを “app_root” に作成・構成変更

アプリケーション “MYFIRSTAPP” 構成完了(ver. 1.0)

アプリケーション PDB の作成と Sync

各PDBでデータ投入・参照・更新

アプリケーション・コンテナと実装例 – 留意点(1/3)

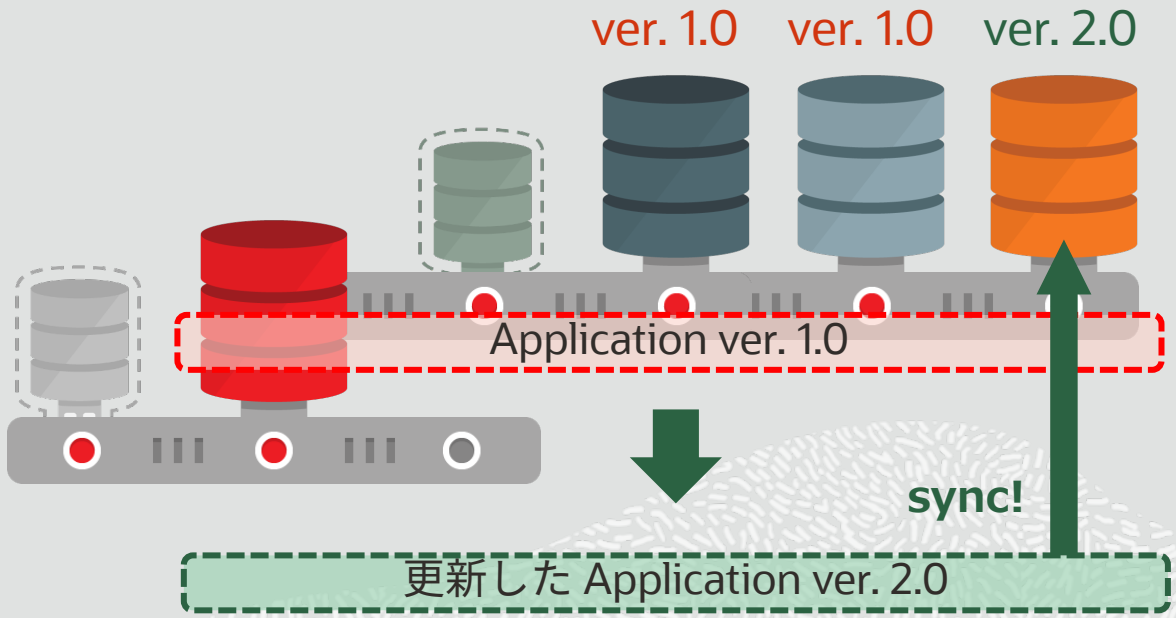
共有リンクオブジェクト変更は Alter … begin/end で行う

```
alter pluggable database
        application MYFIRSTAPP
                begin install '1.0';
        変更処理
        ～

alter pluggable database
        application MYFIRSTAPP
                end install '1.0';
```

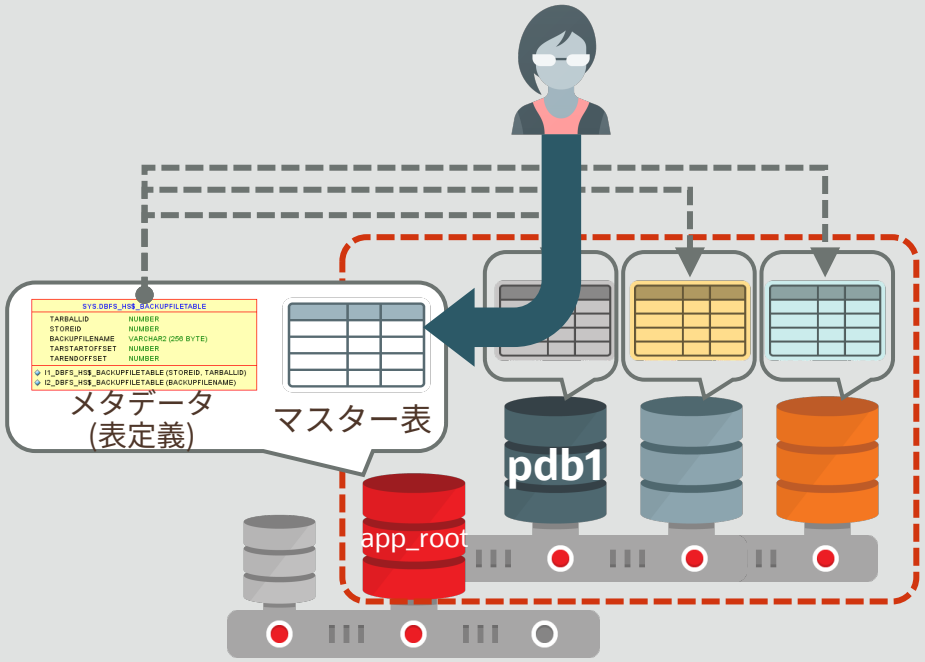
変更後、アプリケーションPDBで sync する

```
alter pluggable database xxx
        application MYFIRSTAPP sync;
```

※ sync しない AP-PDB は古い version で運用が継続される
->検証・開発などで効果的
->version 管理は重要!



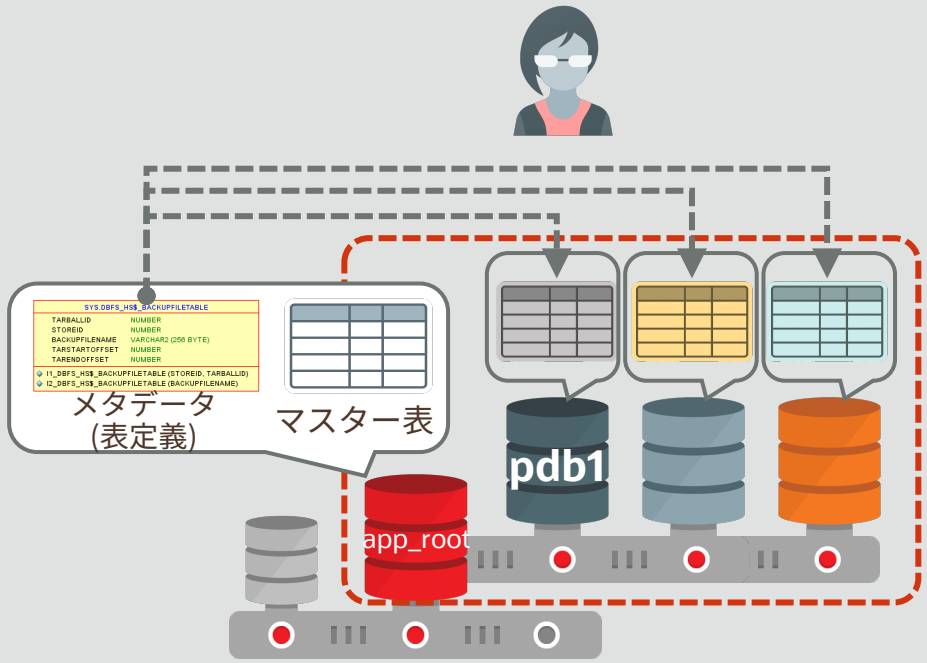
アプリケーション・コンテナと実装例 – 留意点(2/3)



pdb1: R18.3 / R19.4 で検証

```
SQL> connect app_owner/app_owner@ptvm10/pdb1.jp.oracle.com
接続されました。

SQL> select * from emp where empno=7369;
経過: 00:00:00.01

実行計画
----------------------------------------------------------
Plan hash value: 915754307
-------------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |     1 |    38 |     0   (0)| 00:00:01 |
|*  1 |  DATA LINK FULL   | EMP  |     1 |    38 |     0   (0)| 00:00:01 |
-------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("EMPNO"=7369)
```

アプリケーション・コンテナと実装例 – 留意点(3/3)



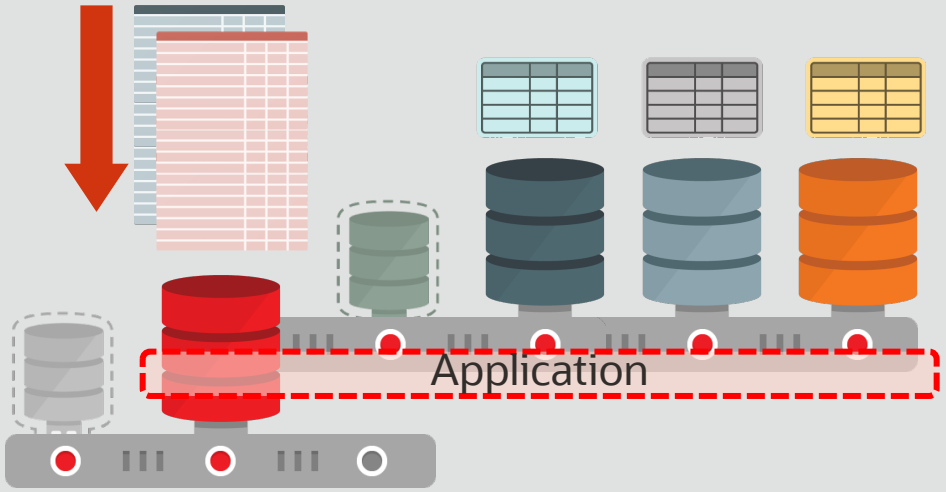
※ Metadata link 表では Index Scan

pdb1: R18.3 / R19.4 で検証

```
SQL> connect app_owner/app_owner@ptvm10/pdb1.jp.oracle.com
接続されました。

SQL> select * from emp,dept where emp.deptno=dept.deptno;
実行計画
----------------------------------------------------------

Plan hash value: 1737840670

-------------------------------------------------------------------------------
| Id  | Operation                    | Name    | Rows  | Bytes | Cost (%CPU)
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |         |     5 |   285 |     3  (34)
|   1 |  MERGE JOIN                  |         |     5 |   285 |     3  (34)
|   2 |   TABLE ACCESS BY INDEX ROWID| DEPT    |     1 |    19 |     2   (0)
|   3 |    INDEX FULL SCAN           | PK_DEPT |     1 |       |     1   (0)
|*  4 |   SORT JOIN                  |         |    14 |   532 |     1 (100)
|   5 |    DATA LINK FULL            | EMP     |    14 |   532 |     0   (0)
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   4 - access("EMP"."DEPTNO"="DEPT"."DEPTNO")
       filter("EMP"."DEPTNO"="DEPT"."DEPTNO")
```

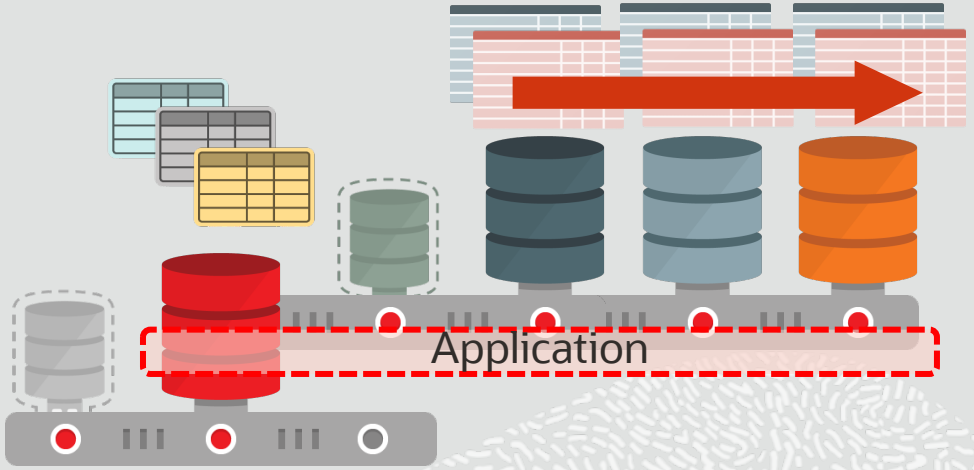
アプリケーション・コンテナのまとめ – ユース・ケースの例

ユース・ケース(1) – Application Root を軸



マスター管理や主機能をApplication Root軸に実装し、Application PDBからの処理は性能要件を低めに設定する
(データ更新は Application Root)

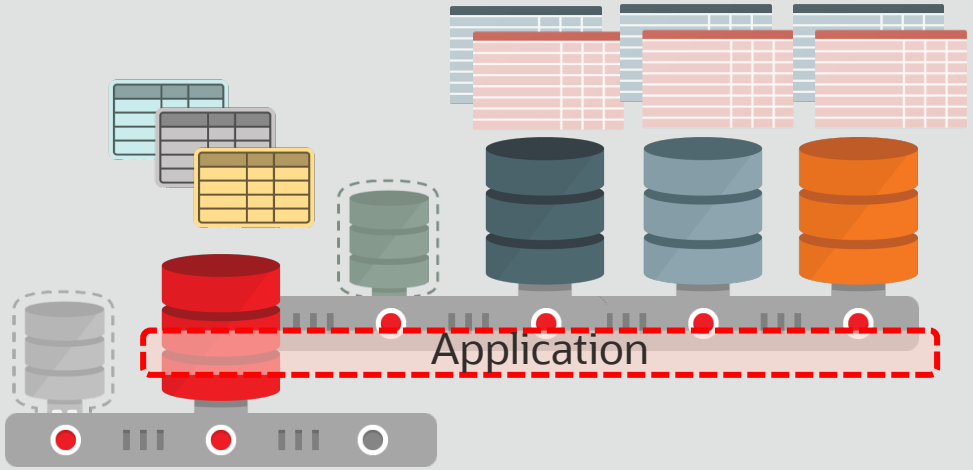
ユース・ケース(2) – Container Map を軸



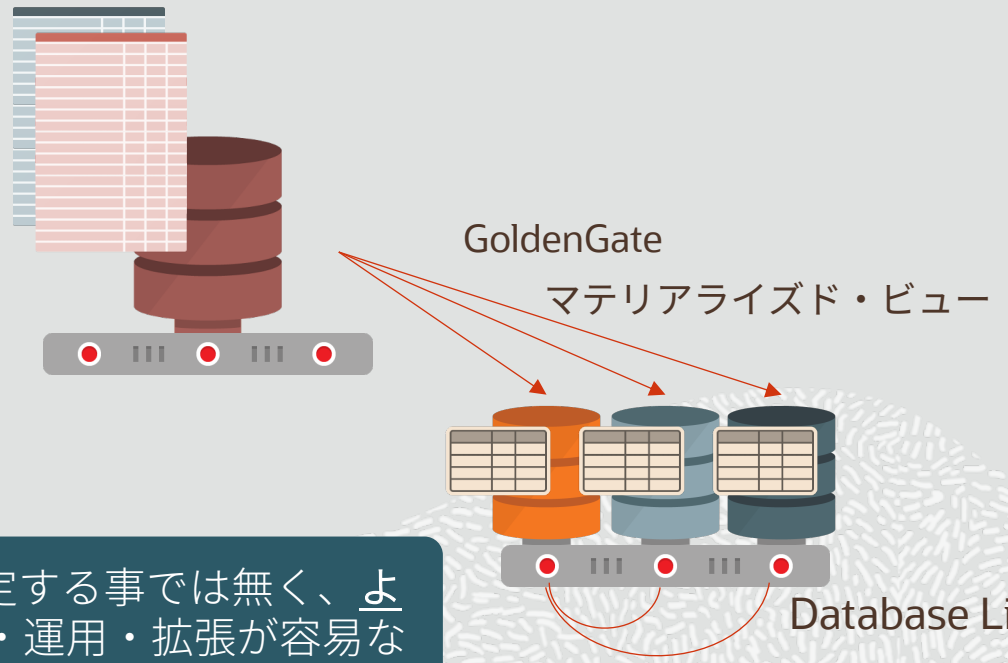
マスター構成や集計はApplication Rootを軸に実装するが、Container Map をベースとした分散データを Application Rootから見たマスターデータとして実装する
(データ更新は Application PDB単位)

アプリケーション・コンテナのまとめ – ユース・ケースの例

Application Container



データ・レプリケーションやリモート接続



どちらか一方を、肯定・否定する事では無く、<u>よりシンプル</u>に、かつ、管理・運用・拡張が容易なアーキテクチャ(・セット)を選択すべき

Program agenda

1. Oracle Database Technology Night #32 を振り返る
2. マルチテナント関連の新機能(18c / 19c)
3. 統合・集約におけるリソース分離性
4. アプリケーション・コンテナと実装例
5. まとめ

まとめ - 19c マルチテナント・アーキテクチャ応用編

- 18c / 19c の新機能とライセンス

対象項目	概要
CDB フリート管理	複数のCDB-ROOTを論理的に一元管理可能
PDB スナップショット・カルーセル	定期的にPDBのCloneを最大8つ取得(古いCloneは自動的にパージ)
リフレッシュ可能 PDB switchover	リフレッシュ可能PDBのロール変換が可能

- リソースの共有と分離
- 新しいデータ共有の形態 – Application Container

まとめ - マルチテナント環境のキュメント・ロードマップ

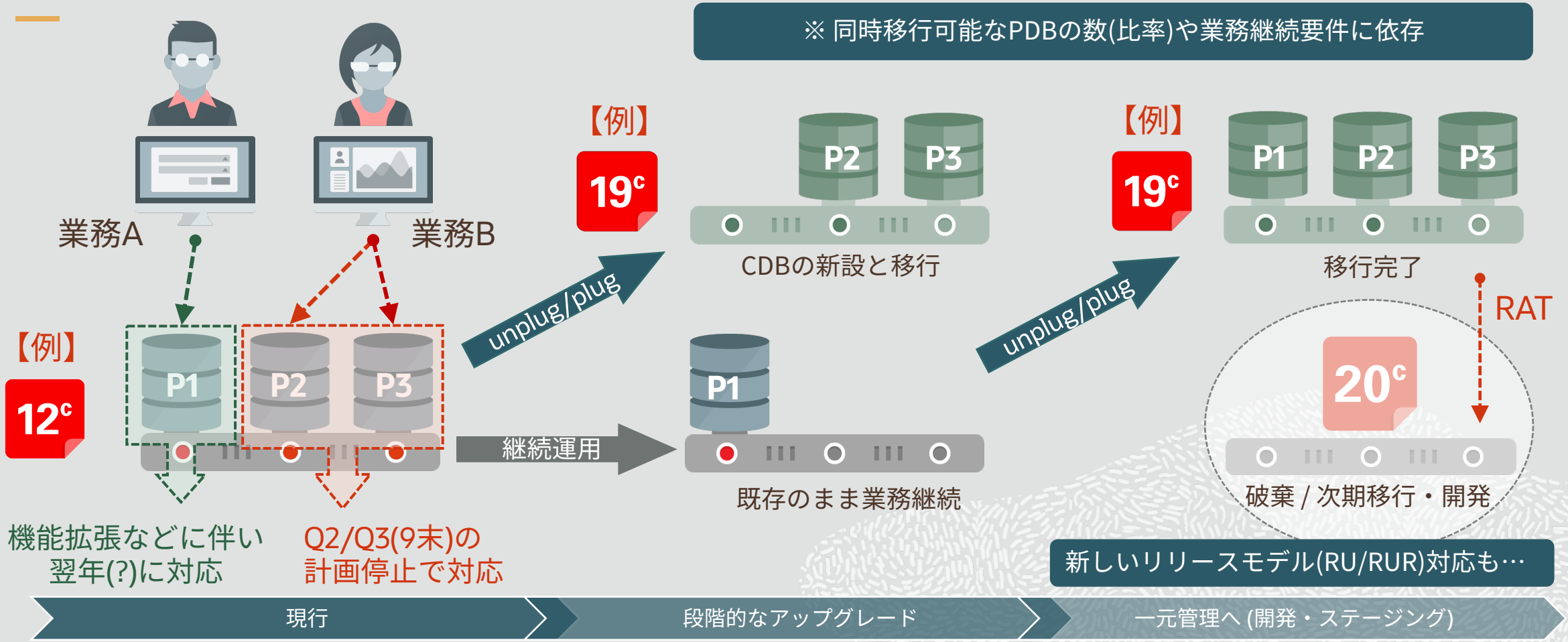
カテゴリ	トピック	ドキュメント
概要	CDBおよびPDBの概要	「マルチテナント・アーキテクチャの概要」
管理	CDBの作成および構成	「CDBの作成および構成」
管理	CDBの管理	「CDBの管理」
管理	PDBの作成および構成	「PDBおよびアプリケーション・コンテナの作成および削除」
管理	PDBの管理	「PDBの管理」
管理	アプリケーション・コンテナの作成および削除	「アプリケーション・コンテナの作成および削除」
管理	アプリケーション・コンテナの管理	「アプリケーション・コンテナの管理」
パフォーマンス	PDBのトラブルシューティング	*Oracle Databaseパフォーマンス・チューニング・ガイド*
監視	CDBおよびPDBに関する情報の表示	「CDBおよびPDB監視」
バックアップおよびリカバリ	CDBのバックアップおよびリカバリの実行	*『Oracle Databaseバックアップおよびリカバリ・アドバンスト・ユーザーズ・ガイド』*
セキュリティ	CDBの共通ユーザー、ロールおよび権限の管理	*『Oracle Databaseセキュリティ・ガイド』*
その他		

https://docs.oracle.com/cd/F19136_01/multi/introduction-to-the-multitenant-architecture.html#GUID-345CDAD9-3D12-475F-82EC-FED2790F504A

Appendix – 再・入門編より

No.	項目	解説とスライド
1	unplug / plug で EE Optionの差異はどうなる?	移行元が移行先と同一か、サブセットである必要があります 例 SE2 → EEなどは可能
2	リフレッシュ可能PDB Switch Overは Data Guard のかわりになるのか? (Active Data Guard との比較)	PDB単位での障害時のF/O(残存PDBの切離し)などは可能ですが、CDB-ROOTの障害時の対応や、ブロック破損・書込み欠損検出などはできません
3	PDBごとの段階的アップグレードのような方法/ベストプラクティスなどあるか？	計画停止などの要件などにも依存しますが、一括アップグレードが困難な場合は複数のCDB環境を準備して unplug&plug がお勧めできるパターンの1つです
4	異なるリリース間でのホットクローンは可能か	可能です。 ただしCDB-ROOTとPDBに差異がある状態ではOpenできますが、制限付きになるので通常のオペレーションはできません。
5	アップグレードに関して 除外リストを使用してCDB-ROOTと異なるリリースのPDBを混在する事は可能か	可能です。 ただしCDB-ROOTとPDBに差異がある状態ではOpenできますが、制限付きになるので通常のオペレーションはできません。
6	アラート・ログなどはPDB単位で参照できる?	v$diag_alert_ext を検索する事で可能です。(サンプル P56)

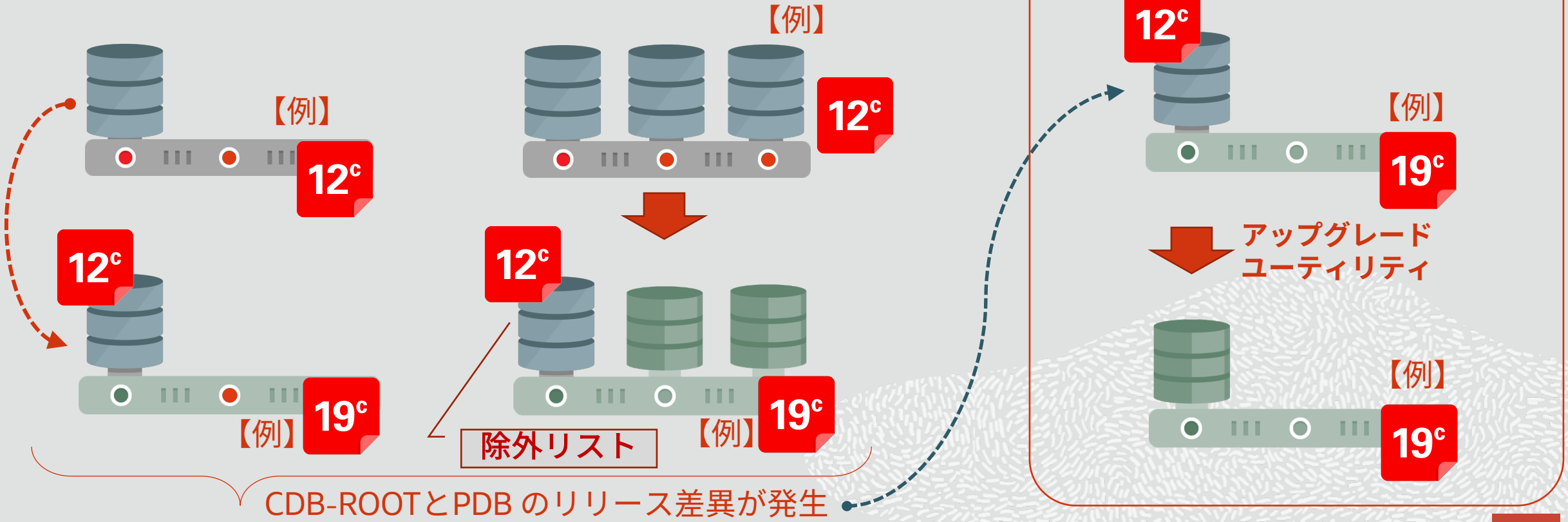
Appendix - 項番3 / 段階的なアップグレード案について



Appendix - 項番4および5についての補足

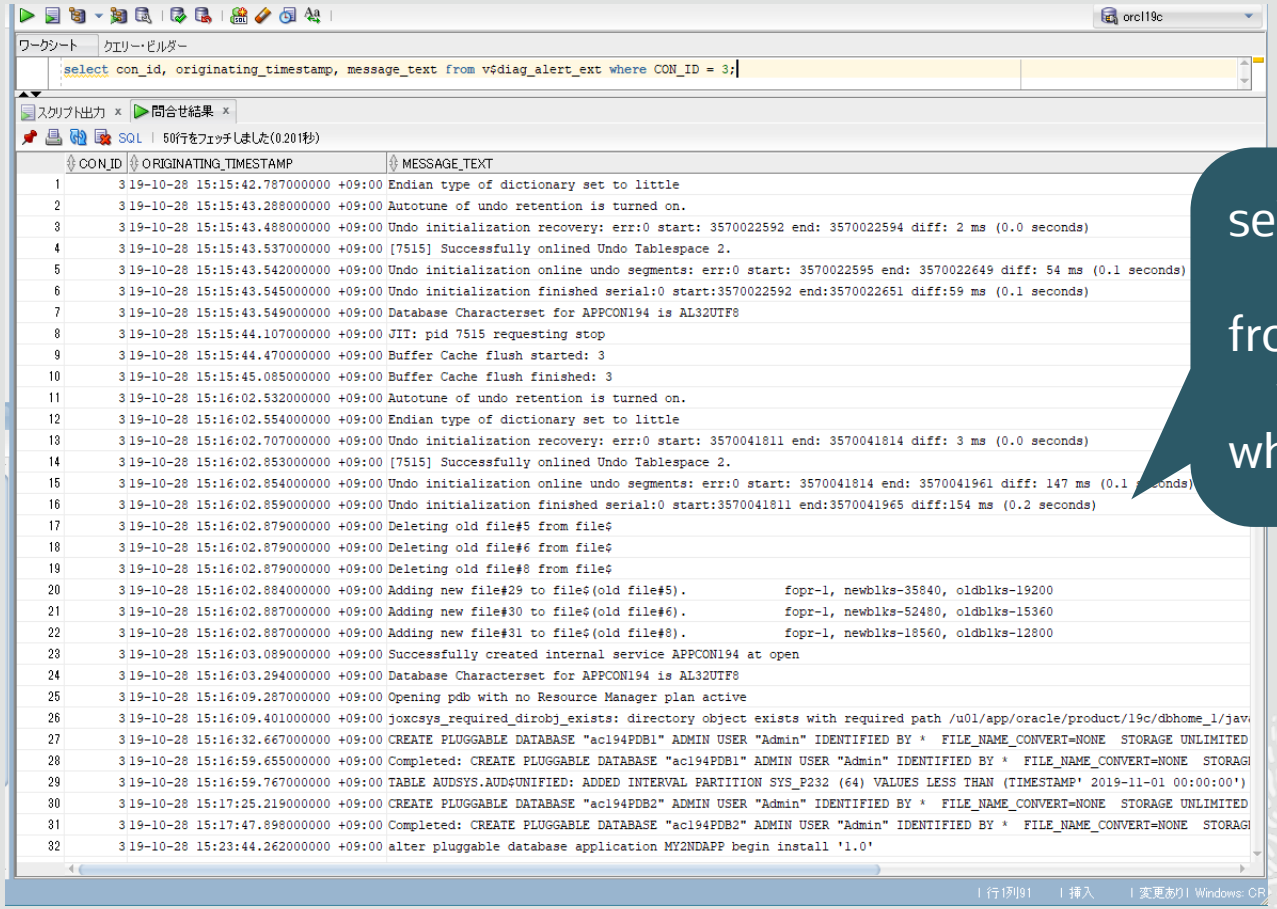
A) 異なるリリース間のクローン作成 (オンライン/ unplug & plug)

B) 部分(PDB)適用または除外アップグレード



Appendix -アラート・ログなどはPDB単位で参照できるか

PDBごとのアラート　v$diag_alert_extビュー



SQL Developer v19

```
select
    con_id,originating_timestamp,message_text
from
    v$diag_alert_ext
where con_id = 3;
```

19.1.3 CDBのビュー
https://docs.oracle.com/cd/F19136_01/multi/viewing-information-about-cdbs-and-pdbs-with-sql-plus.html#GUID-7EE6B341-9A36-4189-BE46-1FDCCCE8B475

アプリケーション・コンテナと実装例 – Appendix

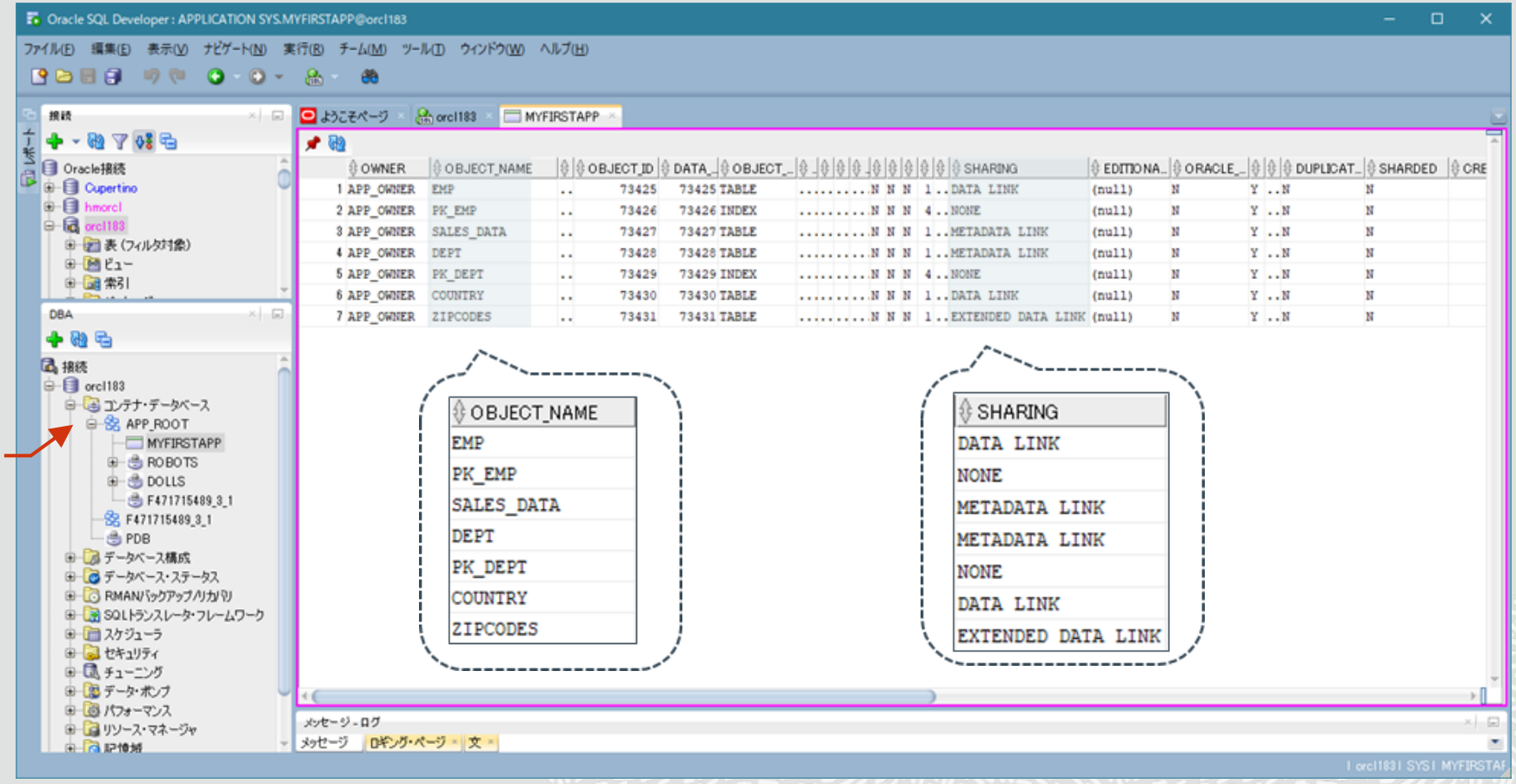
※サンプルイメージ

本資料の作成にあたり、実際に構築した環境をベースに SQL Developer イメージをキャプチャ

Oracle R18.3.0
SQL Developer 18.4.0.376.1900

DBAペインからCDB-rootを参照

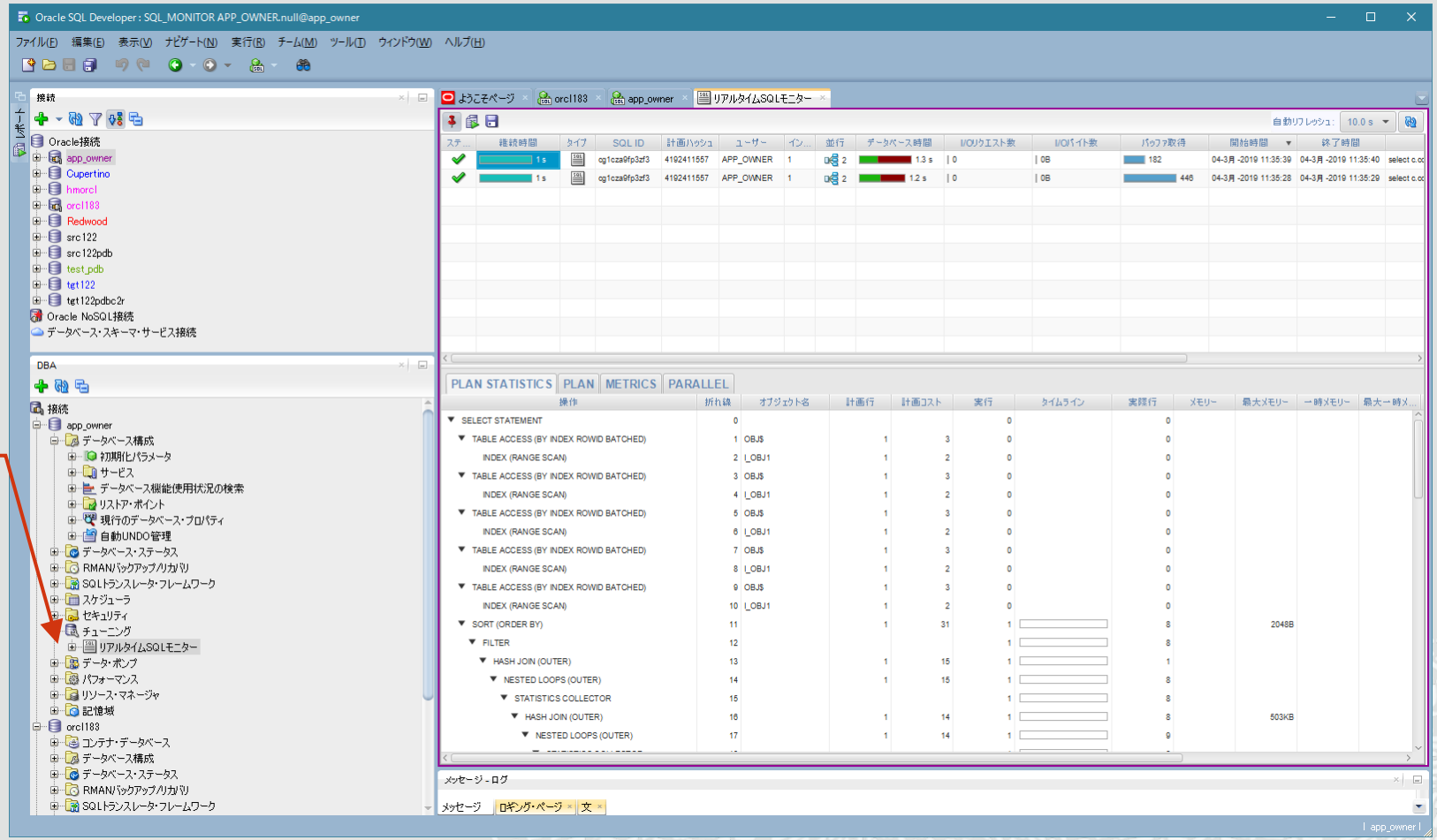
作成したアプリケーション・ルート内に構成した各データリンクオブジェクトと、そのSharing情報を参照したところ



アプリケーション・コンテナと実装例 – Appendix

app_rootに接続したところ

特に共通マスタを保持するため、
性能問題が発生すると全体に波及
する事からもモニタは重要な要素



Thank you

テック・ナイトアーカイブ資料と お役立ち情報

各回テック・ナイトセッション資料
ダウンロードサイト

oracle technight

しばちょう先生の
試して納得！DBA
への道

津島博士の
パフォーマンス講
座

もしも
みなみんが
DBをクラウドで
動かしてみたら

基本からわかる！
高性能×高可用性
データベースシステム
の作り方

～ みなさまの投稿をお待ちしております ～

Twitter

#OracleTechNight

こんな時、かけこむ会社が増えています。

Oracle Digitalは、オラクル製品の導入をご検討いただく際の総合窓口。
電話とインターネットによるダイレクトなコニュニケーションで、どんなお問い合わせにもすばやく対応します。
もちろん、無償。どんなことでも、ご相談ください。

Oracle Digital

お問い合わせは電話またはWebフォーム

☎ 0120-155-096
受付時間　月～金　9:00-12:00 / 13:00-17:00
（祝日および年末年始休業日を除きます）

http://www.oracle.com/jp/contact-us