

ORACLE®

ORACLE®

# Oracle Database 12c Release 1

## Global Data Services

日本オラクル株式会社

ORACLE®  
DATABASE 12<sup>c</sup>



Plug into the **Cloud**.

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。

# アジェンダ

1. Maximum Availability Architecture (MAA)
2. 複製データを活用している環境での問題点
3. Global Data Services と RAC
4. 想定されるユース・ケース
5. グローバル・サービス
6. ロード・バランシング
7. Global Service Manager(GSM)
8. 動作詳細、設定、必要なバージョン
9. Q&A

# Oracleが提供するテクノロジーとアーキテクチャ

## Oracle Maximum Availability Architecture (MAA)

### Production Site

### Active Replica

Oracle  
Real Application Clusters

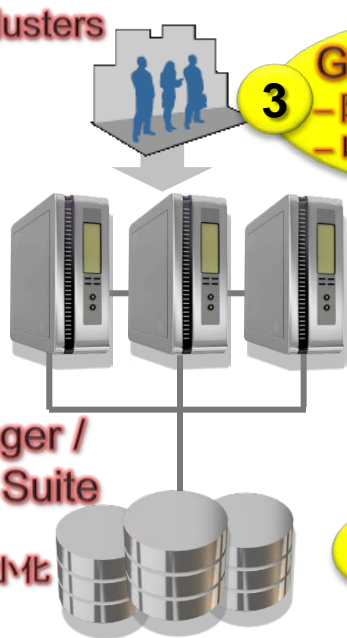
- 拡張性の確保
- 一点障害抑止

Flashback

- 人的ミスの迅速復旧

Enterprise Manager /  
Application Test Suite

- テスト作業の自動化
- 構築/移行コストの最小化



**3 Global Data Service**  
- 障害時の接続先切り替え  
- ロードバランシング

**ASM**  
- ボリューム管理  
- 一点障害抑止

**2 RMAN**  
- データ・バックアップ  
- 人的ミスの抑制

**1 Oracle Data Guard**  
- データの保護/災害対策  
- クエリ・オフロード

**GoldenGate**  
- アクティブ-アクティブ  
- 異種混在環境のサポート

**Oracle Secure Backup**  
- バックアップデータの保護  
- Tape、クラウドの活用



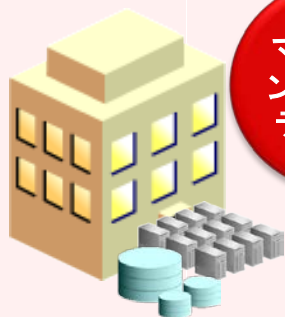
# アジェンダ

1. Maximum Availability Architecture (MAA)
2. 複製データを活用している環境での問題点
3. Global Data Services と RAC
4. 想定されるユース・ケース
5. グローバル・サービス
6. ロード・バランシング
7. Global Service Manager(GSM)
8. 動作詳細、設定、必要なバージョン
9. Q&A

# GDS 実装の背景

※ GDS ... Global Data Services

## コンソリデーション vs ディストリビューション



マルチテナ  
ント・アーキ  
テクチャー

### コンソリデーション(統合)の動機

- 管理の容易さ
- コスト削減

### ディストリビューションの動機

- DR ディザスタリカバリー
- 事業継続性
- 広い範囲で分散しているクライアントに高速にデータを提供
- 大量のクライアントアクセスを待たせることなく処理

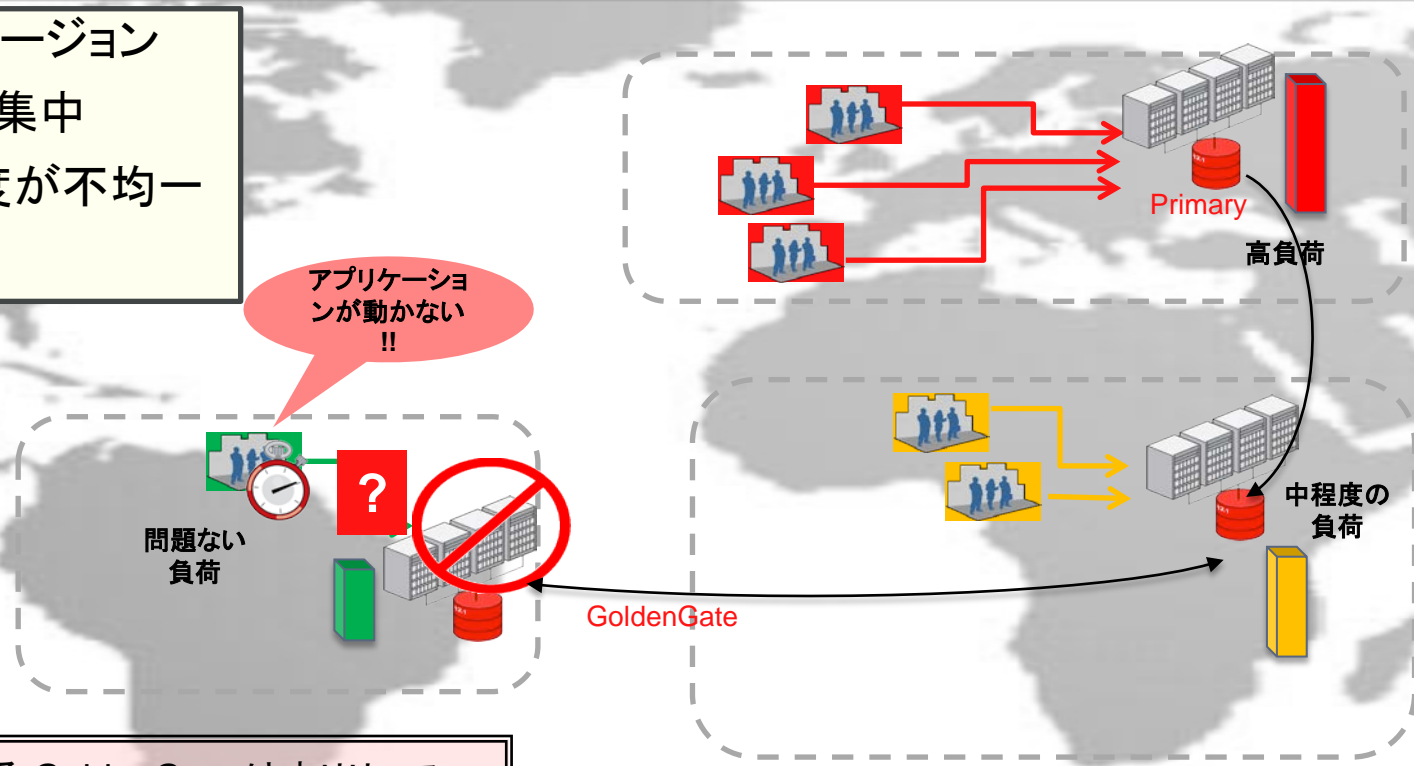
GDS

データの複製(コピー)を多様な業務アプリから生データ(live data)のようにアクセスする使用形態の増加

ORACLE

# 複製データを活用している環境での問題

- サイロ化されたリージョン
- 負荷がーか所に集中
- リソースの使用度が不均一
- 耐障害性がない



注: DB12c に対応する GoldenGate は未リリース

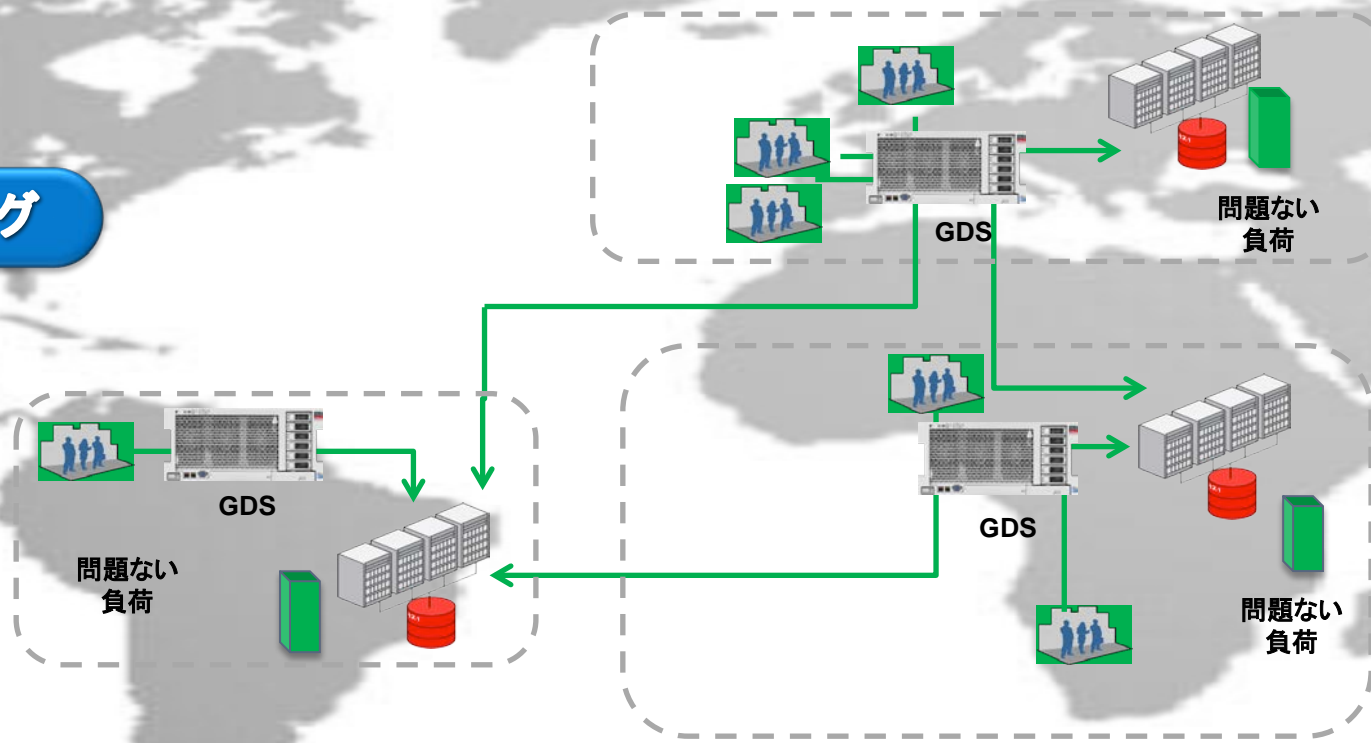


# アジェンダ

1. Maximum Availability Architecture (MAA)
2. 複製データを活用している環境での問題点
3. Global Data Services と RAC
4. 想定されるユース・ケース
5. グローバル・サービス
6. ロード・バランシング
7. Global Service Manager(GSM)
8. 動作詳細、設定、必要なバージョン
9. Q&A

# 複製データを活用している環境 – GDS 導入後

ロードバランシング



ORACLE

# 複製データを活用している環境 – GDS 導入後

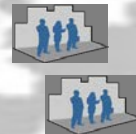
- 負荷を均等に分散
- リソース利用率の最適化
- 管理が楽に
- アプリケーションの耐障害性



耐障害性



SALES\_REPORTING\_SRVC



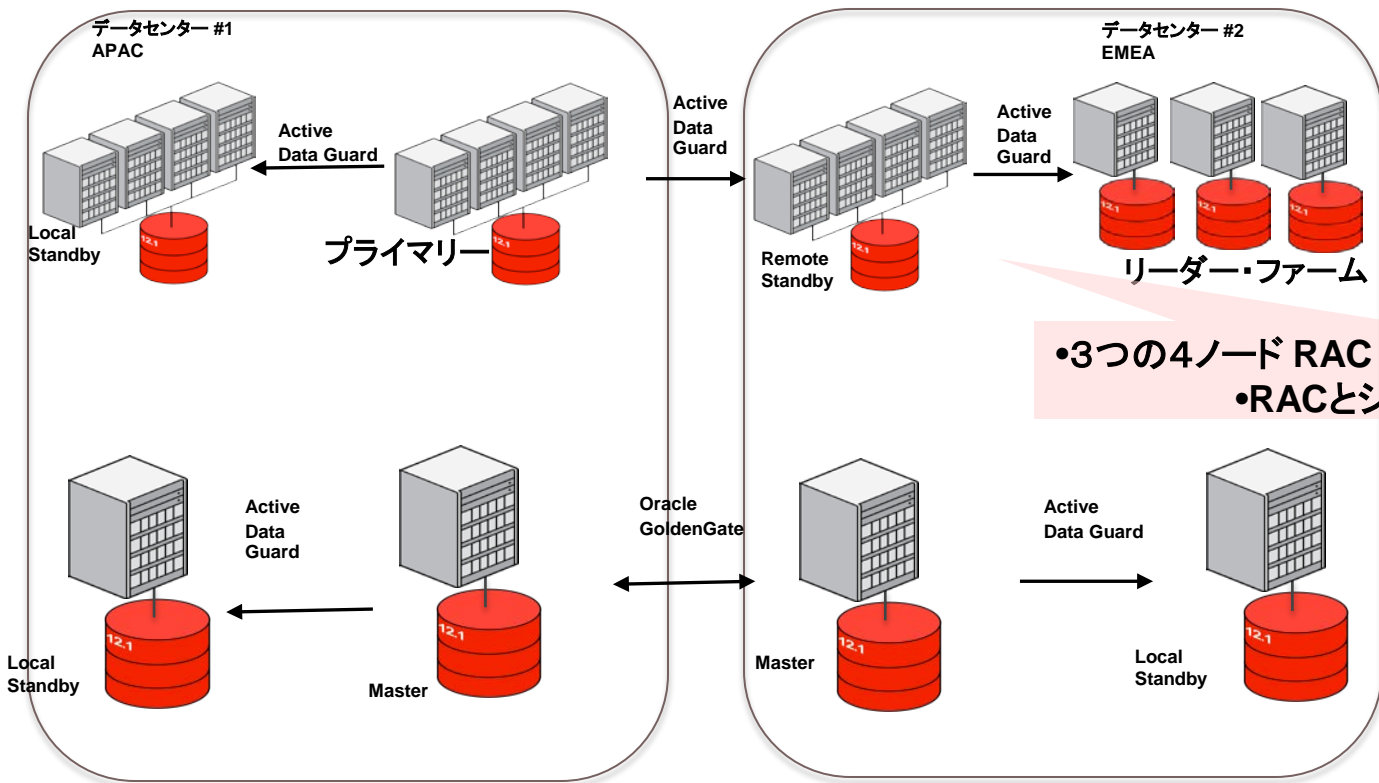
GDS



GDS

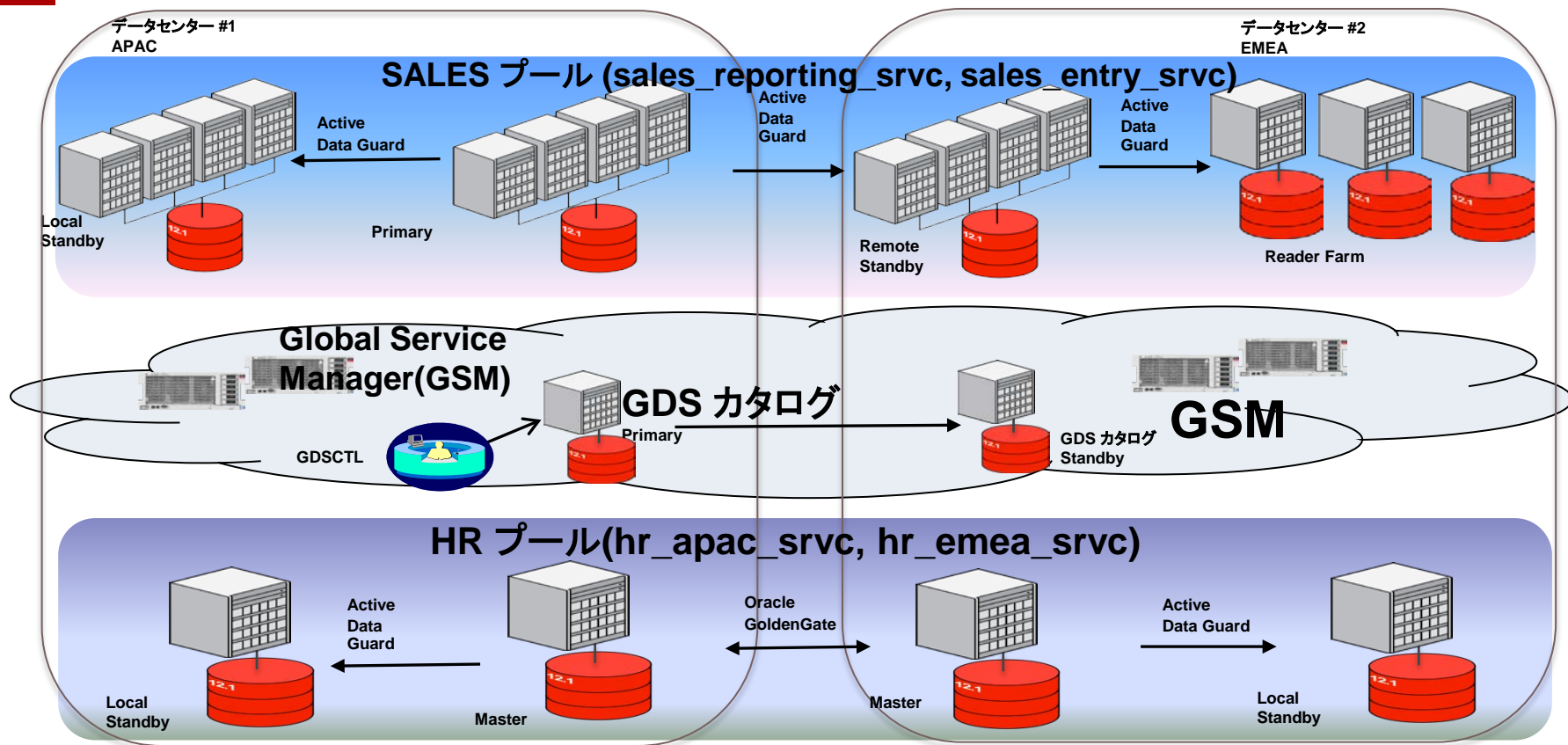


# GDS 導入前後の物理的なイメージ — 導入前



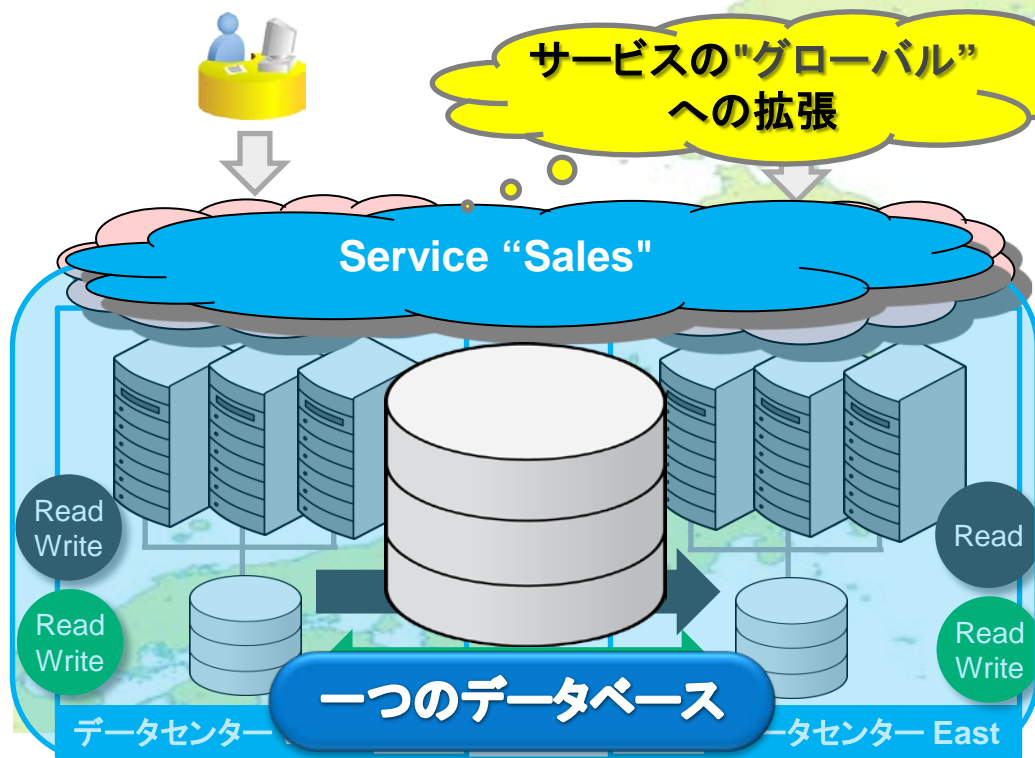
- データセンター 2カ所
  - 2組の独立した複製グループ
  - レプリケーション手段は問わない  
例) Streams, Advanced Repl, スクリプト, ストレージ・コピー
- 3つの4ノード RAC と 3 シングル・インスタンスDB  
• RACとシングルは混在可能

# GDS 導入前後の物理的なイメージ - 導入後



# Global Data Services – RAC クラスタ

独立したDB群を1つのRACクラスタのように使うことが可能に



## GDSの特徴

- 負荷分散  
DCを跨いだロードバランシングを実現
- 障害時フェイルオーバー  
自動でのフェイルオーバーが可能
- 集中管理  
単一(or少数)のサービスで一括管理

# Global Data Service と RAC の概念・用語対応表

RAC(Real Application Cluster)	GDS(Global Data Services)
RAC クラスター	GDS プール
インスタンス、または、ノード	データベース
SCAN リスナー	GSM エンドポイント(GSM リスナー)
	リージョン
ポリシー管理プール	
データベース・サービス	データベース・サービス
カーディナリティー	カーディナリティー
その他同じ用語、機能名が同じ意味で使われるもの多数	

# Global Data Services(GDS) と RAC 共通機能

- 1つのデータベース・“サービス” が複数のサーバーで起動
- ランタイム・ロード・バランシング
  - Oracle 製コネクションプール使用時
- 接続時サーバーサイド・ロード・バランシング
- 高速接続フェイル・オーバー(FCF, Fast Connection Failover)
- クライアントからの接続先を集約
  - RAC: SCAN (Single Client Access Name)
  - GDS: GSM (Global Service Manager) エンドポイント
- 一カ所での管理
  - RAC: Clusterware(CRS, OCR)
  - GDS カタログ



# GDS にあり RAC がない機能

- クライアントからDBまでのネットワーク・レイテンシー(Round Trip Time)を計算に入れたロードバランシング
- 複製元からの“データの遅れ”を監視しサービスの起動・停止
  - Active Data Guard
- クライアントの場所とDBの場所とのアフィニティー
- 多数データベース間でのロードバランス
- 処理能力に差があるデータベース間でのロードバランス
- “ロール”によるフェイル・オーバー
  - Data Guard

# GDS になく RAC にある機能

- すべてのノードで最新のデータ
- インターノード・パラレル・クエリー
- TAF(透過的接続フェイル・オーバー)
- アプリケーション・コンティニューイティ(Application Continuity)
- Oracle Clusterware (CRS, Grid Infrastructure) が提供する機能
  - インスタンス起動
  - OS 再起動
  - Voting, ハートビート,...



# 業務、アプリケーション の適性

全てのサーバーで最新データにアクセスできるとは限らないので制約がある

読み取りのみの処理  
と書き込みありの処理  
の分離

例)レポーティング系

データが最新でなくて  
も良いか

例)  
30秒前  
前日  
先月末

コネクション・プール  
使用

ORACLE製プール

# ユース・ケース – データベース側

代表的な例の大まかな分類

読み取り専用ファーム  
での負荷分散

(リーダー・ファーム、  
Reader Farm)  
Active Data Guard  
GoldenGate

フェイル・オーバー  
スイッチ・オーバー

Data Guard  
Active Data Guard  
コールド・スタンバイ

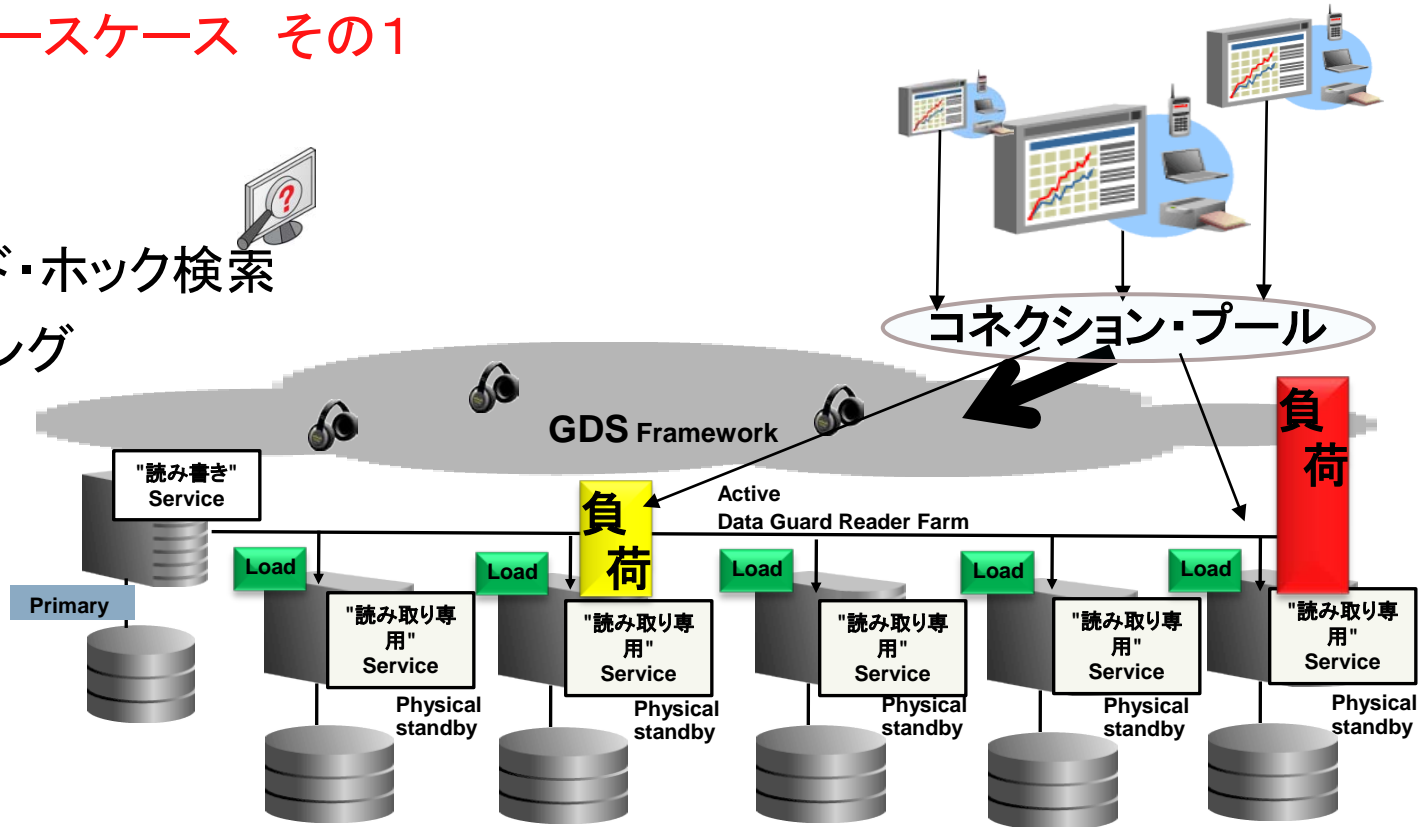
マルチ・マスター環境  
での負荷分散

GoldenGate  
Active-Active 構成

# リーダー・ファームの使用用途と負荷の偏り

## 想定されるユースケース その1

- バックアップ
- レポート作成
- 制限のないアド・ホック検索
- データ・マイニング



# アプリがリーダー・ファームを有効利用できるか？

Data Guard または GoldenGate 環境

100% 読み取り

リーダー・ファーム  
に接続

“Read Mostly” アプリ  
ケーション

データベース・リン  
クの利用  
または  
複数のデータソース

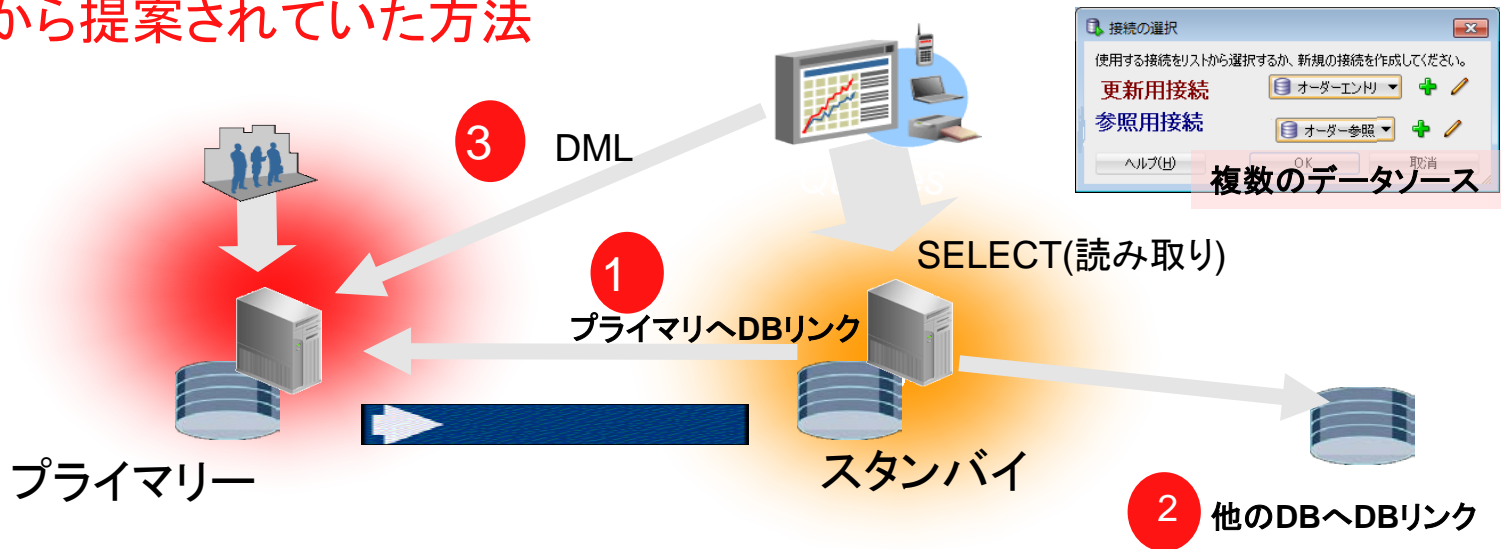
書き込みがある程度  
以上

プライマリー  
または  
マスターに接続

読み取り比率

# “Read-mostly” アプリケーションの3つの実現方法

11g 時代から提案されていた方法



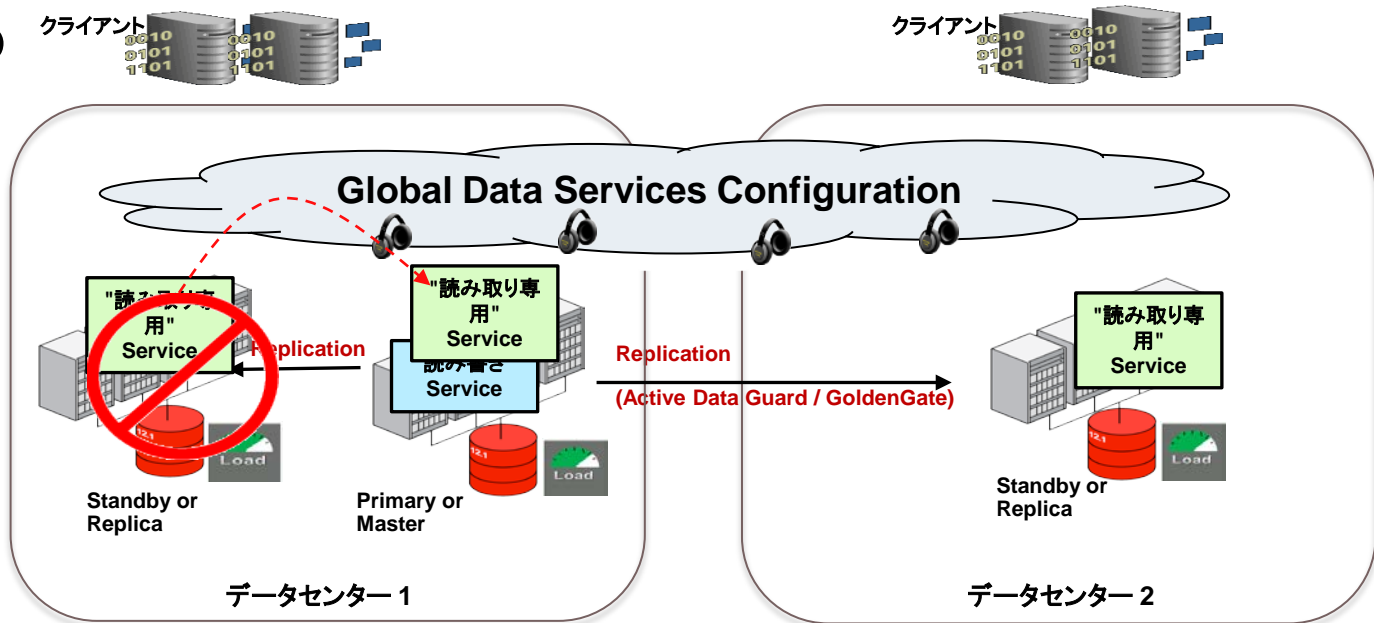
1. プライマリへのデータベース・リンクとシノニムを使用
2. プライマリでないDB(例: DG構成に含まれないローカルDB)
3. アプリケーションで振り分ける



# フェイル・オーバー — Data Guard, GoldenGate

## 想定されるユースケース その2

- “DBサービス”のフェイル・オーバー
- カーディナリティーを維持する



凡例:



Load Balancing



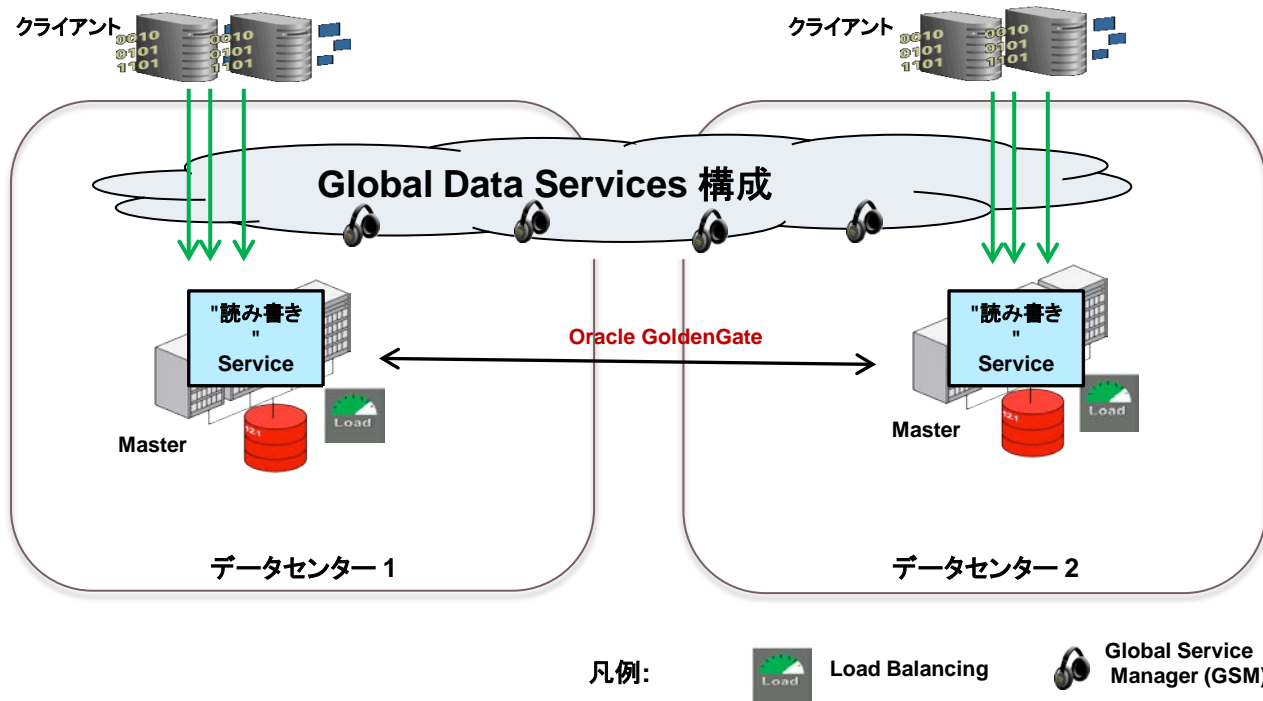
Global Service Manager (GSM)

ORACLE

# GoldenGate Active-Active でのロード・バランシング

## 想定されるユースケース その3

- データ更新を含むアプリケーションのロードバランシング
- アプリケーションの変更なしで利用できる
- データ競合(コンフリクト)の解決手段は GDS は提供しません
  - GoldenGate の機能などでの実装を想定



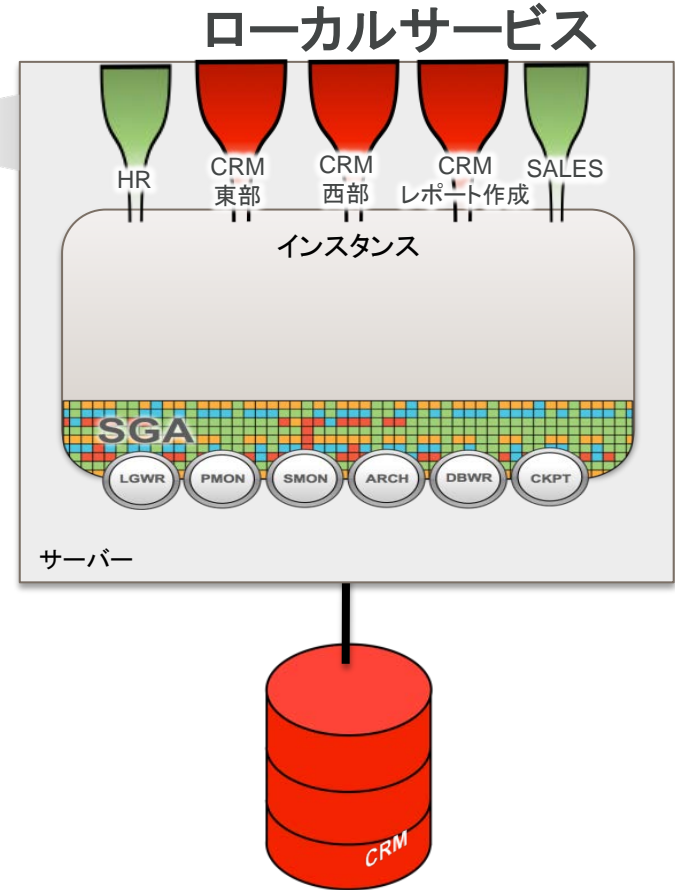
# アジェンダ

1. Maximum Availability Architecture (MAA)
2. 複製データを活用している環境での問題点
3. Global Data Services と RAC
4. 想定されるユース・ケース
5. グローバル・サービス
6. ロード・バランシング
7. Global Service Manager(GSM)
8. 動作詳細、設定、必要なバージョン
9. Q&A

# データベース “サービス”

- ローカルな”サービス”は以前のバージョンからあった仕組みで主にワークロード管理のために使われていました。
- シングルインスタンスDBでも作成可能
- クラスタリソースとして作成可能
- Global Data Services でのグローバルサービスと対比させるため旧来のサービスが “ローカル” サービスと呼ばれることがあります。

サービス



# グローバル・サービス固有の 3 属性

ローカル・サービスには存在しない新しい属性

## カーディナリティー

- サービスを提供するDBの数
- リージョン毎に設定可能

## 遅延(ラグ、LAG)

- Active Data Guard 固有
- REDO適用遅延

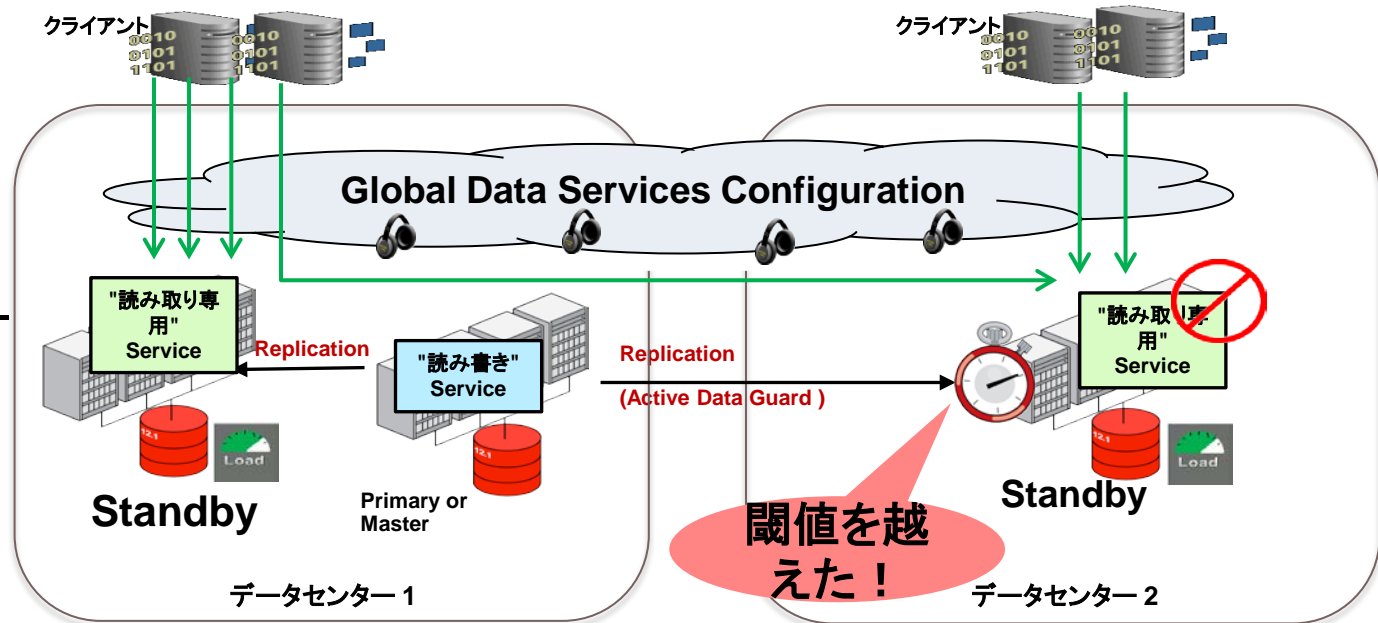
## リージョン・アフィニティ

遠隔データセンターのDBへの接続を許可するかどうか

# ラグ属性を持った”グローバル”サービスの動作

## Active Data Guard スタンバイでのREDO適用遅延

- 1種のSLA (Service Level Agreement)
- 右図はプライマリー1に対し、ローカルとリモートのDCにそれぞれスタンバイが1個



凡例:



Load Balancing

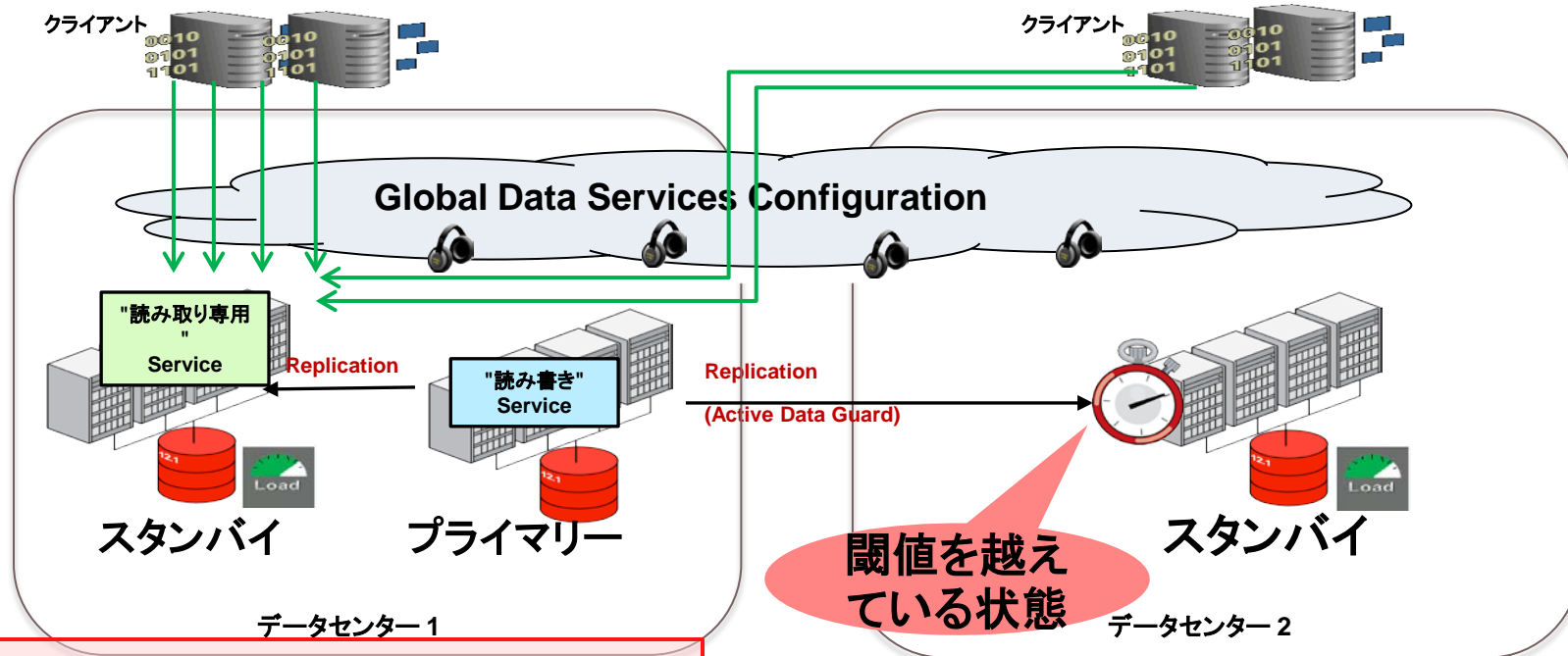


Global Service Manager (GSM)

ORACLE

# ラグ属性を持った”グローバル”サービスの動作

## Active Data Guard スタンバイでのREDO適用遅延



適用が追いついて再度閾値を下回ればサービスは再起動されます

凡例:

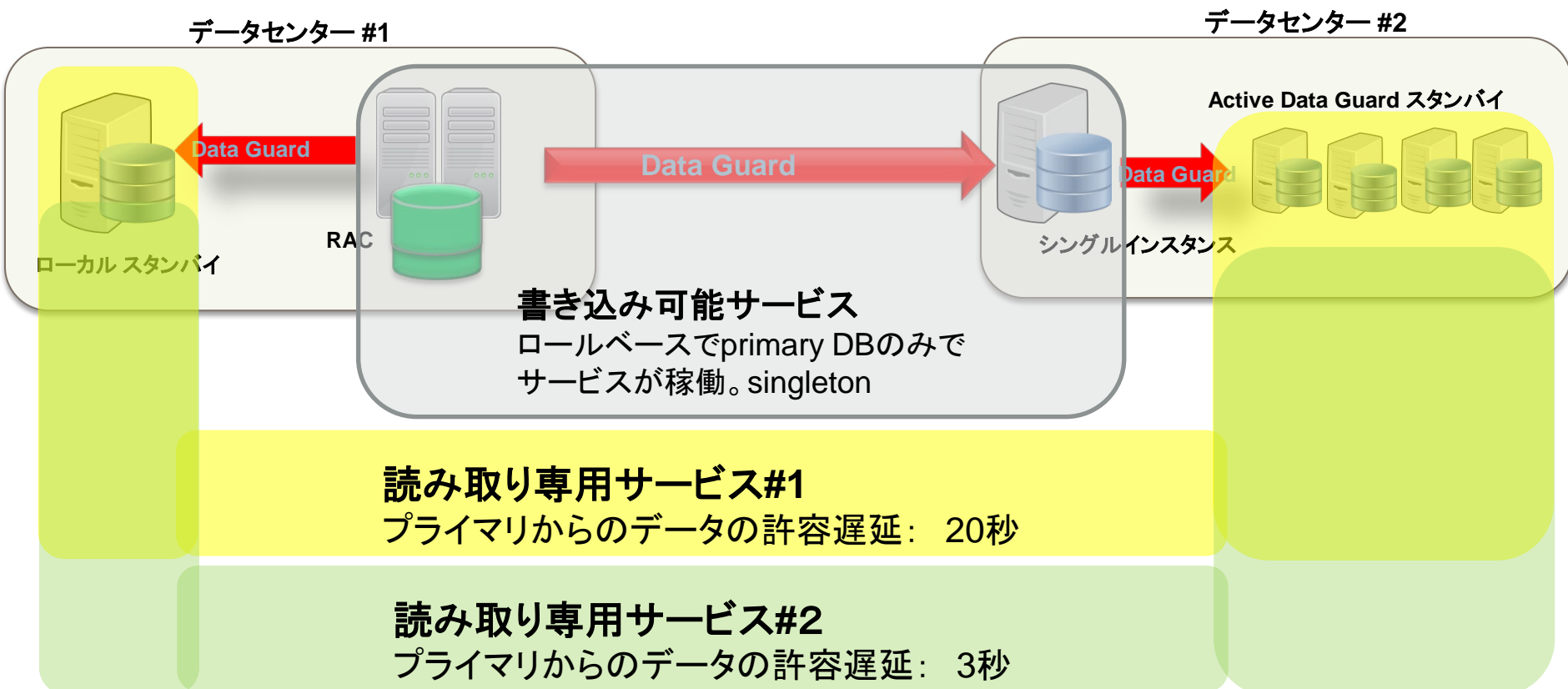


Load Balancing



Global Service Manager (GSM)

# サービス定義: Active Data Guard 環境での例





# GDS を使わない場合で似た機能を実現するには

## 11g からあった機能

- ログオントリガーでアプリケーションユーザのセッションが常に STANDBY\_MAX\_DATA\_DELAY を見るよう設定
  - ログオンはできてもSELECTを実行できない場合があります
- GDS のラグ属性機能と併用可能

```
SQL> connect sys/oracle@prod as sysdba
SQL> CREATE OR REPLACE TRIGGER hr_logon_set_SLA_trigger
2  AFTER LOGON ON hr.schema
3  BEGIN
4  IF (SYS_CONTEXT('USERENV','DATABASE_ROLE') IN ('PHYSICAL STANDBY'))
5  THEN
6    execute immediate 'ALTER SESSION SET STANDBY_MAX_DATA_DELAY=2; '
7  END IF;
8  END;
```

# アジェンダ

1. Maximum Availability Architecture (MAA)
2. 複製データを活用している環境での問題点
3. Global Data Services と RAC
4. 想定されるユース・ケース
5. グローバル・サービス
6. ロード・バランシング
7. Global Service Manager(GSM)
8. 動作詳細、設定、必要なバージョン
9. Q&A

# サービスのロード・バランシング属性

```
GDSCTL>config service -service sales_read_only
```

名前: sales\_read\_only

ネットワーク名: sales\_read\_only.hr.oradbcloud

プール: hr

起動済: はい

すべて優先: はい

ローカリティ: ANYWHERE

リージョン・フェイルオーバー: いいえ

ロール: PHYSICAL\_STANDBY

プライマリ・フェイルオーバー: はい

ラグ: ANY

ランタイム・バランス SERVICE\_TIME

接続バランス LONG

通知: はい

TAFポリシー: NONE

ポリシー: AUTOMATIC

DTP: いいえ

フェイルオーバー・メソッド: NONE

フェイルオーバー・タイプ: NONE

フェイルオーバー再試行:

フェイルオーバー遅延:

エディション:

PDB:

コミット結果:

保存タイムアウト:

再生開始タイムアウト:

セッション状態一貫性: DYNAMIC

SQL翻訳プロファイル:

SERVICE\_TIME または THROUGHPUT または NONE

LONG  
または  
SHORT

データベース

-----

データベース

優先

ステータス

-----

--

-----

dbvm3

はい

有効

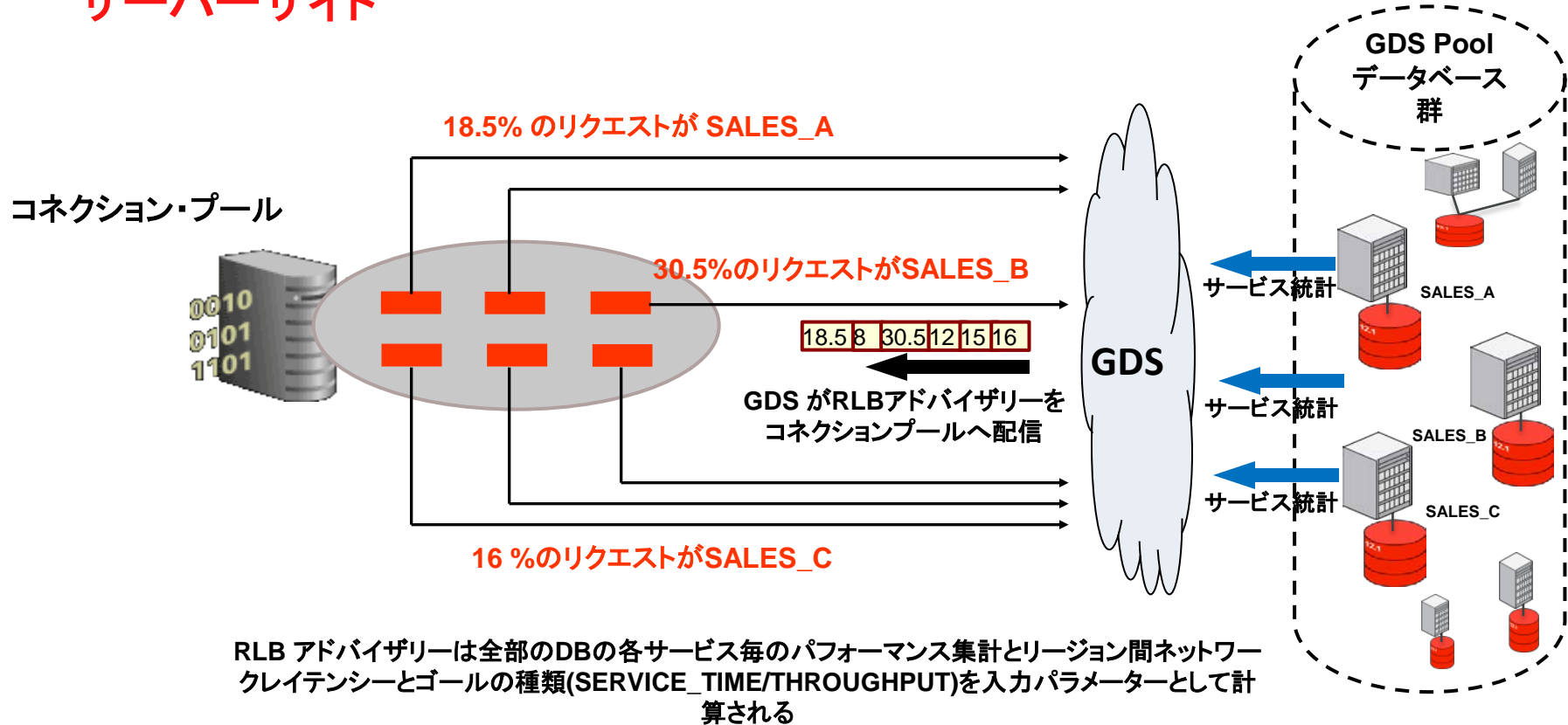
dbvm4

はい

有効

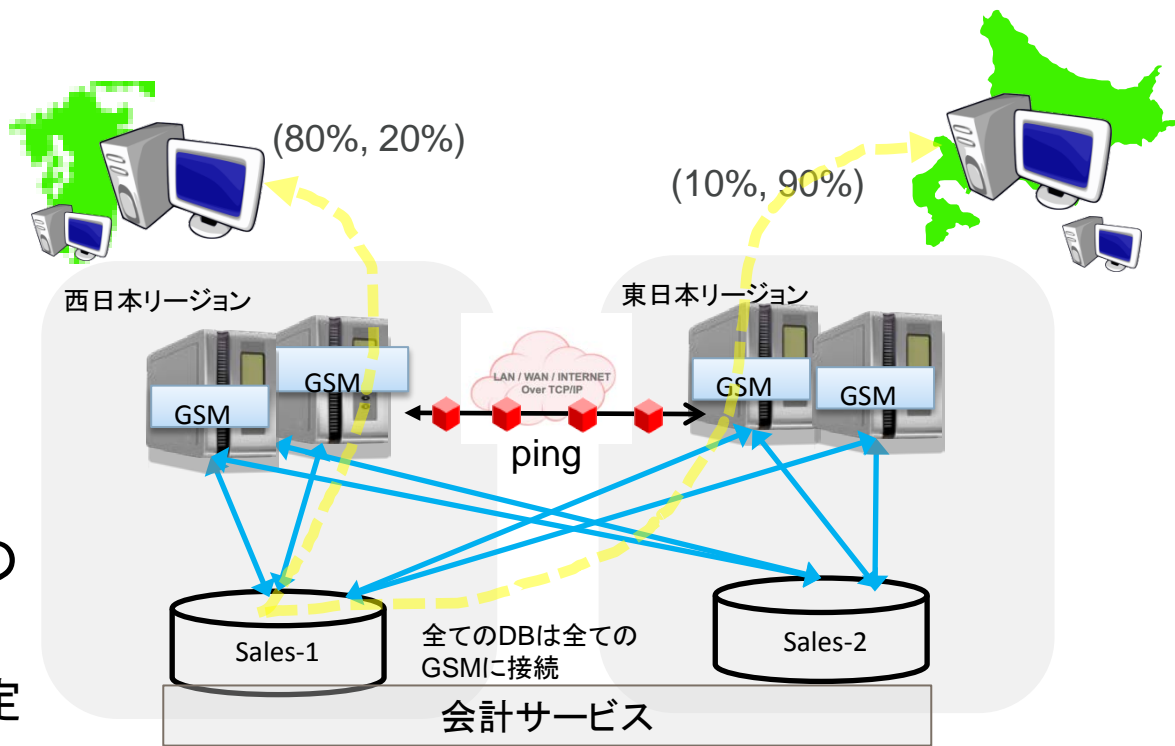
# グローバル・ランタイム・ロード・バランシング (RLB)

## サーバーサイド



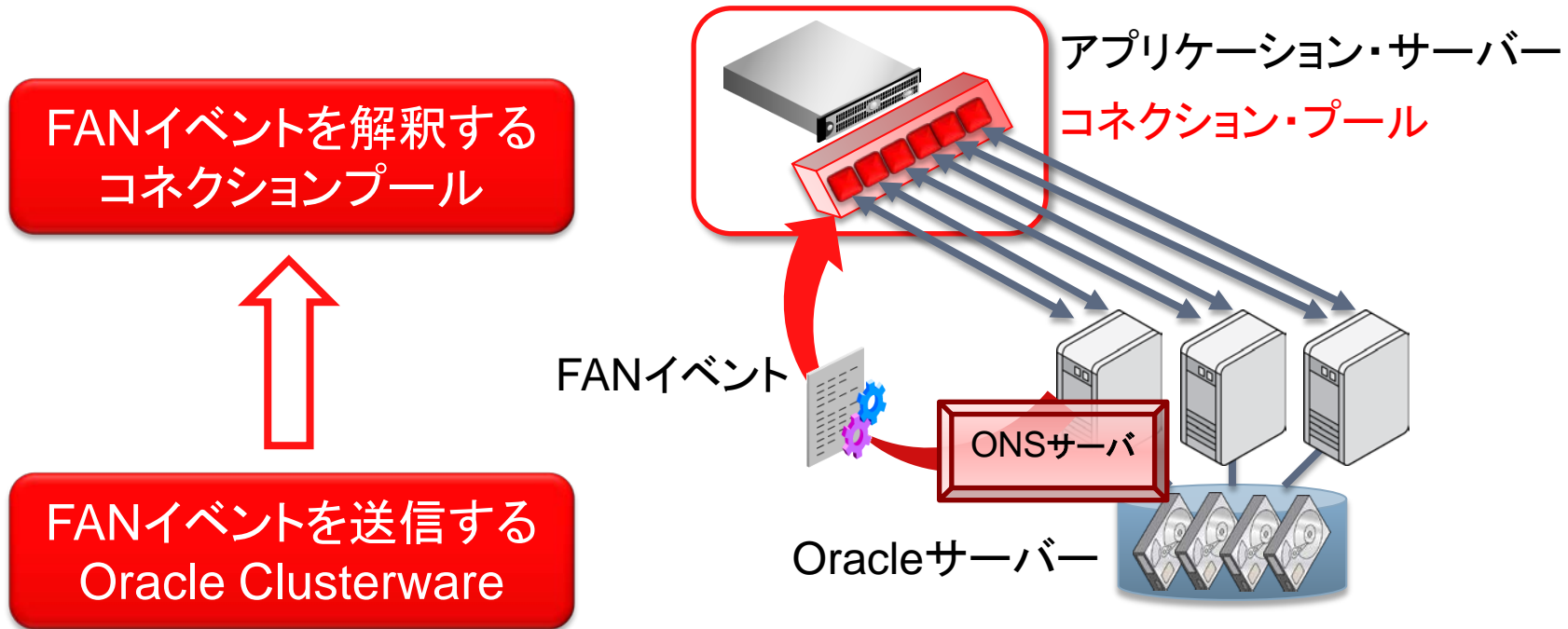
# DB12c RAC でのロードバランシングとの比較

- ロードバランシングの設定や計算方法は 10gR2,11g,12c RAC と 12c の GDS は似ています。
- 異なる点: GSMは各DBから取ってくる負荷情報に REGION 間のネットワーク的な距離を加味した値をクライアントにガイドします
- RLB\_GOAL が THROUGHPUT の時はDBの負荷情報のみでガイド
  - (DB時間) ≫ network と仮定



# Fast Application Notification: FAN

RACノードからOracleクライアントにイベントを伝える仕組み



# ONSからの通知情報: DB11g との違い

## LBA: ロードバランシングアドバイザー

- DB 11g, 2ノード RAC

```
VERSION=1.0      database=orcl      service=srv1_rclb {  
  {instance=orcl1 percent=57 flag=GOOD aff=FALSE}  
  {instance=orcl2 percent=43 flag=GOOD aff=FALSE}  
} timestamp=2012-05-15 20:38:23
```

```
VERSION=1.0      database=orcl      service=srv1_rclb {  
  {instance=orcl1 percent=63 flag=GOOD aff=FALSE} {instance=orcl2 percent=37 flag=GOOD aff=FALSE}  
} timestamp=2012-05-15 20:39:23
```

- 12c GDS, 4つのDB

```
VERSION=1.0      database=pool1.oradbcloud      service=west%sales.pool1.oradbcloud {  
  { instance=pool1%11 percent=20   percentf=20.0000  flag = GOOD aff=TRUE }  
  { instance=pool1%2  percent=30   percentf=30.0000  flag = GOOD aff=TRUE }  
  { instance=pool1%12 percent=33   percentf=33.3333  flag = GOOD aff=TRUE }  
  { instance=pool1%1  percent=17   percentf=16.6666  flag = GOOD aff=TRUE }  
} timestamp=2012-08-11 03:05:31  timezone=-07:00
```

# ONSからの通知情報: DB11g との違い: 続き

- database=pool1.oradbcloud
  - GDS では “プール” があたかも一つのDBのように見えます
- percentf=33.3333
  - 11g RAC では整数値
  - GDS ではサービスを提供する DB が数十個に及ぶこともあるため、精度が上げられています(percent float)
- GDSでは配分比率にはリージョン間のネットワーク Round Trip Time(RTT, レイテンシー)により重みづけされた値が配信される
  - 同じサービスでもリージョン毎に違う配分比率配列が配信される



# GDS ロードバランシング新機能 CPU, I/O 閾値

- GDSプールのDBのインスタンスレベルロード情報、OS レベルロード情報
  - Disk I/O 閾値
    - インスタンスレベル
    - シングル・ブロック・Read 時間
    - Default 20ms
  - CPU 閾値
    - OSレベル
    - Default 75%
- GSM はこの2つを全DBから受信
  - 閾値に近づいているDBがある場合、LBA(ロードバランシングアドバイザー)の配分比率を調整

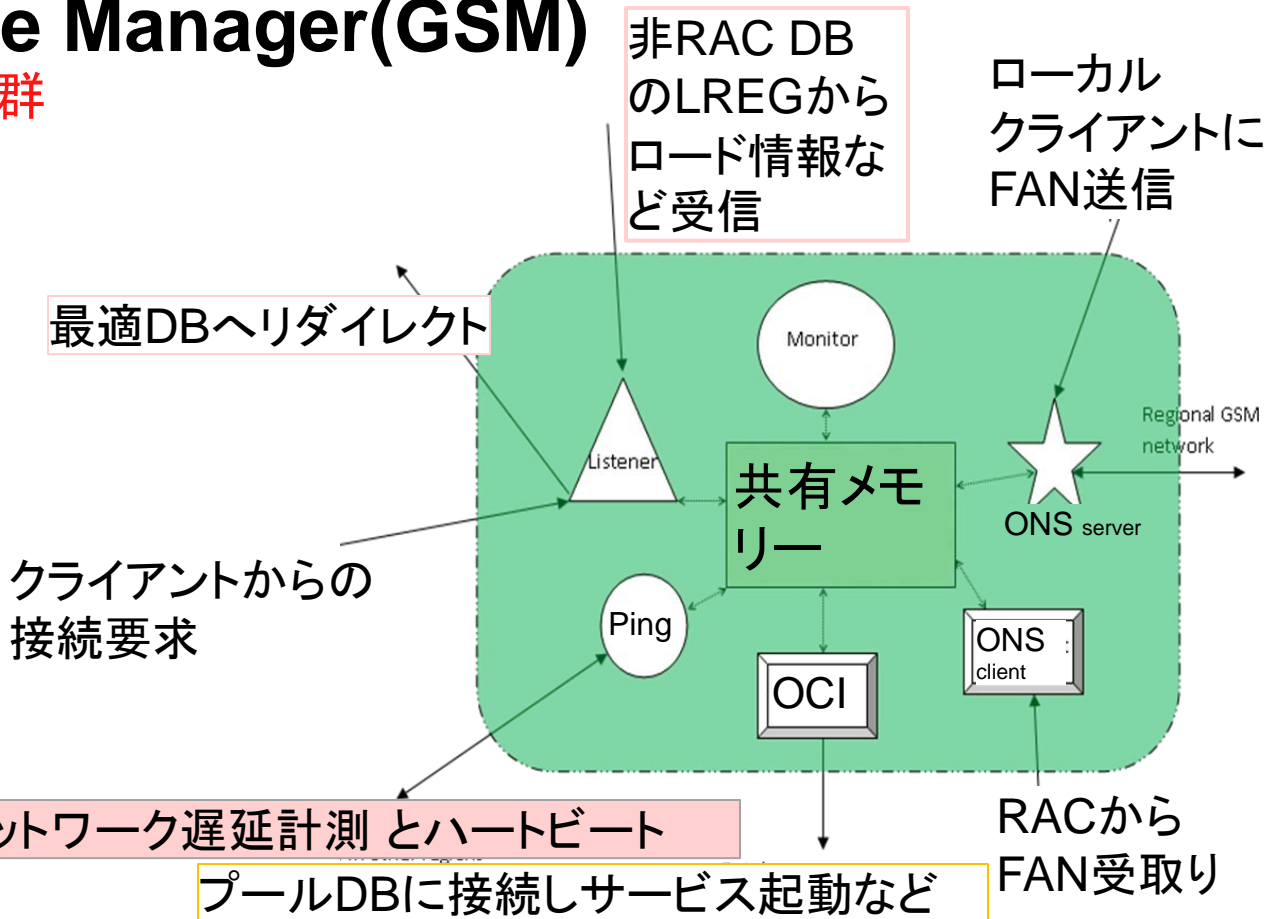
# アジェンダ

1. Maximum Availability Architecture (MAA)
2. 複製データを活用している環境での問題点
3. Global Data Services と RAC
4. 想定されるユース・ケース
5. グローバル・サービス
6. ロード・バランシング
7. Global Service Manager(GSM)
8. 動作詳細、設定、必要なバージョン
9. Q&A

# Global Service Manager(GSM)

## 共有メモリとプロセス群

- 比較的軽量
- 以下のようなプロセス
  - Monitor
  - Listener
  - Pinger
  - OCI Process
  - ONS Server
  - ONS Client



# GSM の実体

リージョン名

```
$ ps -o etime,args | sort -r
```

```
18:42:41 /gsm/bin/gsmmon west_gsm1 -inherit
```

```
18:42:40 /gsm/bin/gsmoci ifile=/gsm/network/admin/gsm.ora west_gsm1
```

```
18:42:40 oraclegdscat (LOCAL=NO)
```

```
18:42:40 oraclegdscat (LOCAL=NO)
```

GDSカタログDBへの接続

```
18:42:39 /gsm/bin/tnslsnr ifile=/gsm/network/admin/gsm.ora west_gsm1 -mode gsm
```

```
18:42:39 /gsm/bin/gsm ping ifile=/gsm/network/admin/gsm.ora west_gsm1
```

```
18:42:39 /gsm/bin/gsmopxy ifile=/gsm/network/admin/gsm.ora west_gsm1
```

```
18:42:39 /gsm/bin/gsm onsc ifile=/gsm/network/admin/gsm.ora west_gsm1
```

```
18:42:39 oraclegdscat (LOCAL=NO)
```

```
$ ipcs -m
```

```
----- Shared Memory Segments -----
```

key	shmid	owner	perms	bytes	nattch	status
0x0c1d0000	44105750	gsm	640	8438224	6	

共有メモリに  
6プロセス

# アジェンダ

1. Maximum Availability Architecture (MAA)
2. 複製データを活用している環境での問題点
3. Global Data Services と RAC
4. 想定されるユース・ケース
5. グローバル・サービス
6. ロード・バランシング
7. Global Service Manager(GSM)
8. 動作詳細、設定、必要なバージョン
9. Q&A

# 論理的な構成物(コンポーネント)



GDS configuration(構成)  
デフォルト名: oradbcloud

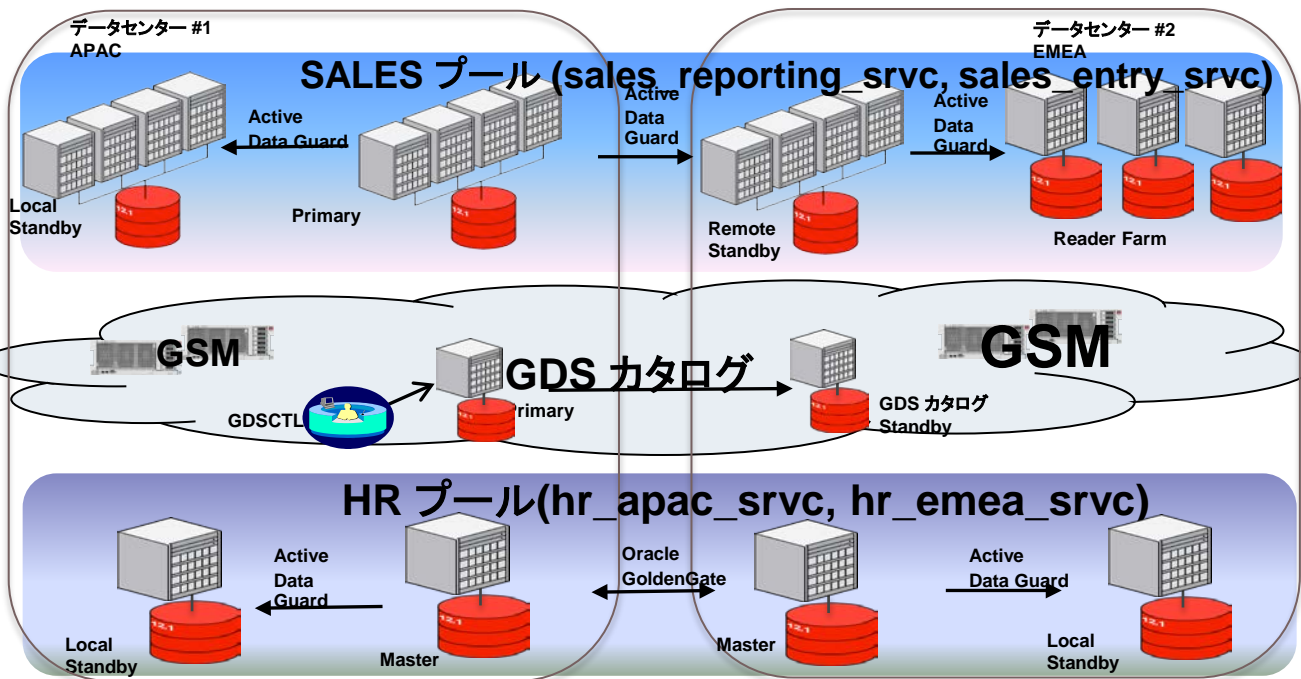
リージョン  
デフォルト名: regionora

リージョン#2

GDSプール  
デフォルト名: dbpoolora

GDSプール#2

# Global Data Services 全体の規模感

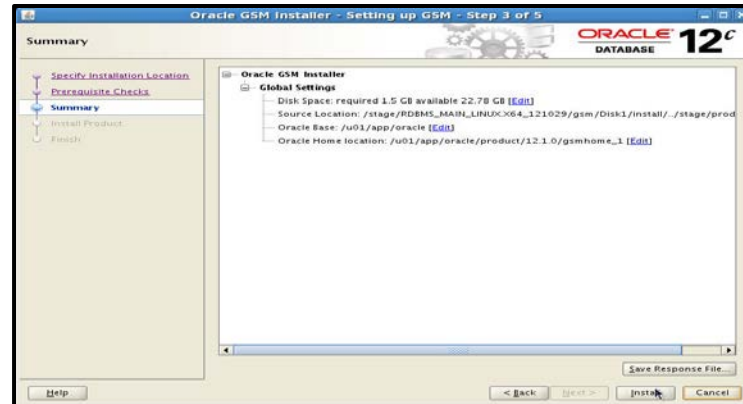
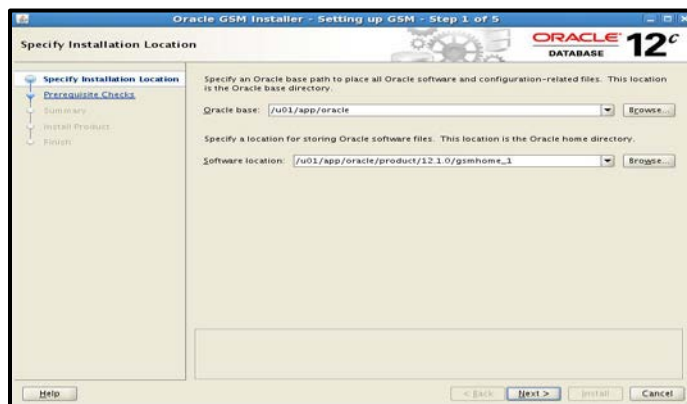


- 300 DB インスタンス
- 1000 Global Services
- 20 GDS プール
- 10 GDS リージョン
- リージョンあたり5つの GSM サーバー
- 100 connection pool 中間層

# Global Service Manager(GSM) インストール

GSM はデータファイルを持たずインストールも高速

- 別メディアからのインストール
- 各リージョンで最低1個
- 複数筐体で稼働させることが推奨
  - その場合もインストールは複数回必要





# CLI(コマンドラインインターフェース)での設定の流れ

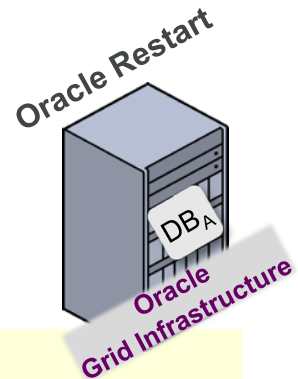
EM12 Cloud Control は DB Plug-in 12.1.0.5 で対応予定

- gdsctl コマンドラインツールを使った初期設定例
  - srvctl に操作感は近いです

```
GDSCTL> create catalog -database dbc
GDSCTL> ADD GSM -gsm west_gsm1 -catalog dbc
GDSCTL> set gsm -gsm west_gsm1
GDSCTL> start GSM // ローカルで実行する必要があります
GDSCTL> add region -region west,east
GDSCTL> add gdspool -gdspool myfarm
GDSCTL> modify gsm -gsm west_gsm1 -region west
GDSCTL> add database -database dbvm3 -connect server_a:1521:dbvm3 -region west -gdspool myfarm
GDSCTL> add service -gdspool myfarm -service sales.example.com -locality local_only -preferred
DB1,DB2 -available db3 // カーディナリティ(cardinality): 2
GDSCTL> start service -gdspool myfarm -service sales.example.com
GDSCTL> add database -database dbvx4 -connect server_b:1521:dbvx4 -region west -gdspool myfarm
```

# 11g Data Guard: ロールベース・サービス

- 11gR2 からはクラスタウェア(Grid Infrastructure)で可能
  - シングル・インスタンスでは Oracle Restart が必要



```
srvctl add service -d <db_unique_name> -s <service_name> [-l  
[PRIMARY][,PHYSICAL_STANDBY][,LOGICAL_STANDBY][,SNAPSHOT_STANDBY]]  
[-y {AUTOMATIC | MANUAL}][-q {true|false}] [-j {SHORT|LONG}] [-B  
{NONE|SERVICE_TIME|THROUGHPUT}][-e {NONE|SESSION|SELECT}] [-m {NONE|BASIC}][[-z  
<failover_retries>] [-w <failover_delay>] [-t <edition>] [-f]
```

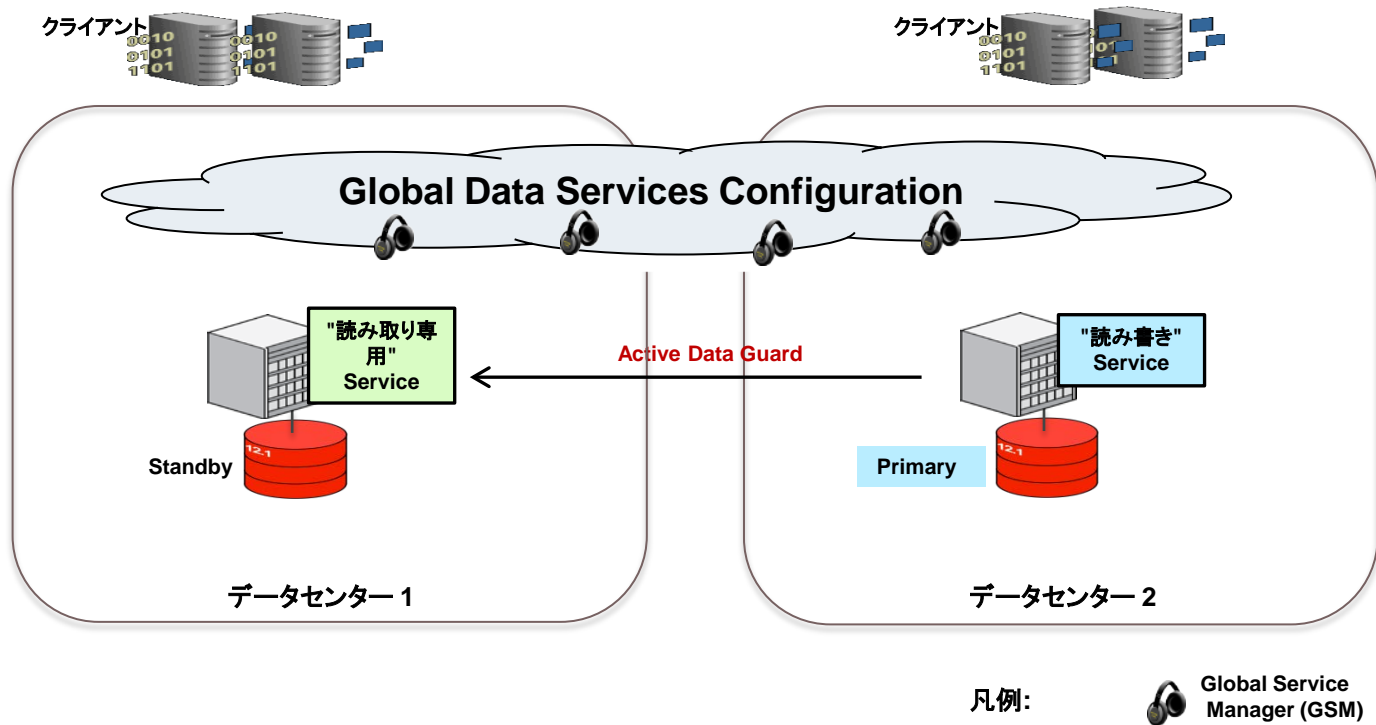
- 11gR1 まではシステムイベントによるトリガーで実現

```
create trigger set_rc_svc after DB_ROLE_CHANGE on database  
[...]  
select database_role into role from v$database ;  
if role = 'PRIMARY' then  
  DBMS_SERVICE.START_SERVICE('adg_ac') ;
```

# DB12c: Global Data Services (GDS)

## Data Guard ロール変換

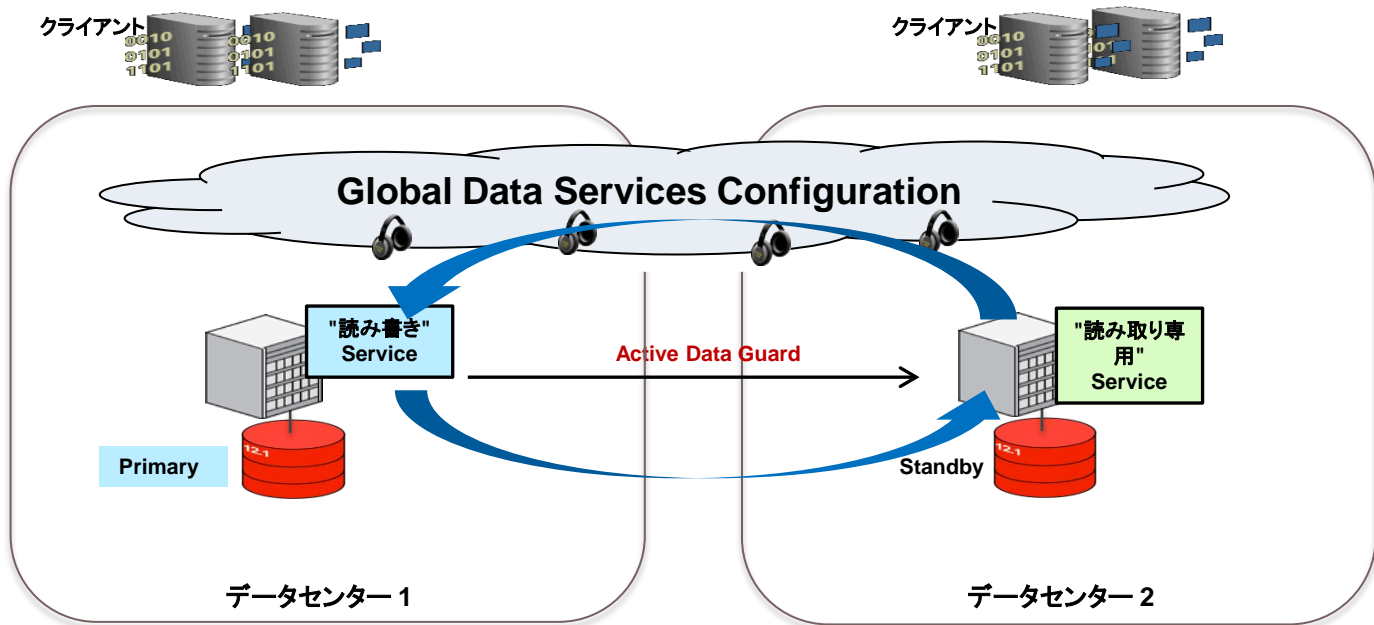
- “読み書き” サービスはプライマリで動作するようにサービス属性で設定されている
- “読み取り専用” サービスは物理スタンバイで動作するようにサービス属性で設定されている



# DB12c: Global Data Services (GDS)

## Data Guard ロール変換

- Fast Start Fail Over (FSFO) やスイッチ・オーバーによるロール変換が起こると Data Guard Broker が GSM に通知しグローバル・DB・サービスも入れ替わる。
- Oracle Clusterware (CRS, Grid Infrastructure) は不要
  - シングル・インスタンスでは Oracle Restart 不要



凡例:  Global Service Manager (GSM)

ORACLE

# GDS を使わないロールベース・サービスとの違い

1種類多い

- 11g R2 は4種類

```
srvctl add service -d <db_unique_name> -s <service_name> [-l  
[PRIMARY][,PHYSICAL_STANDBY][,LOGICAL_STANDBY][,SNAPSHOT_STANDBY]]
```

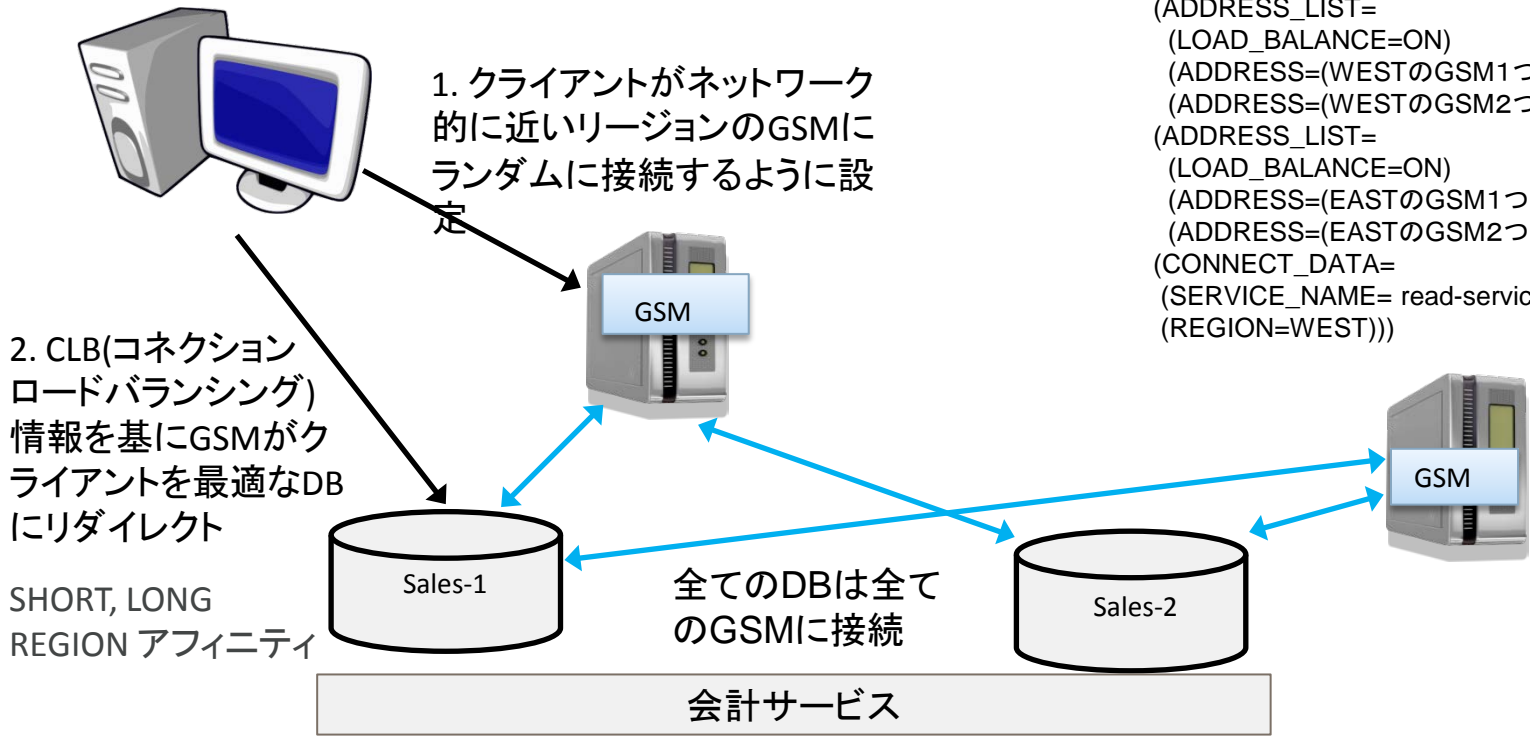
- Global Data Services では 5 種類

```
GDSCTL> add service [-gdspool gdspool_name] -service service_name  
{-preferred_all | -preferred dbname_list [-available dbname_list]}  
[-locality {ANYWHERE | LOCAL_ONLY [-region_failover]}]  
[-role {PRIMARY | PHYSICAL_STANDBY [-failover_primary] | LOGICAL_STANDBY | SNAPSHOT_STANDBY}]
```

# クライアントの”グローバル・サービス”への接続

## CLB – 接続時サーバーサイドロードバランシング

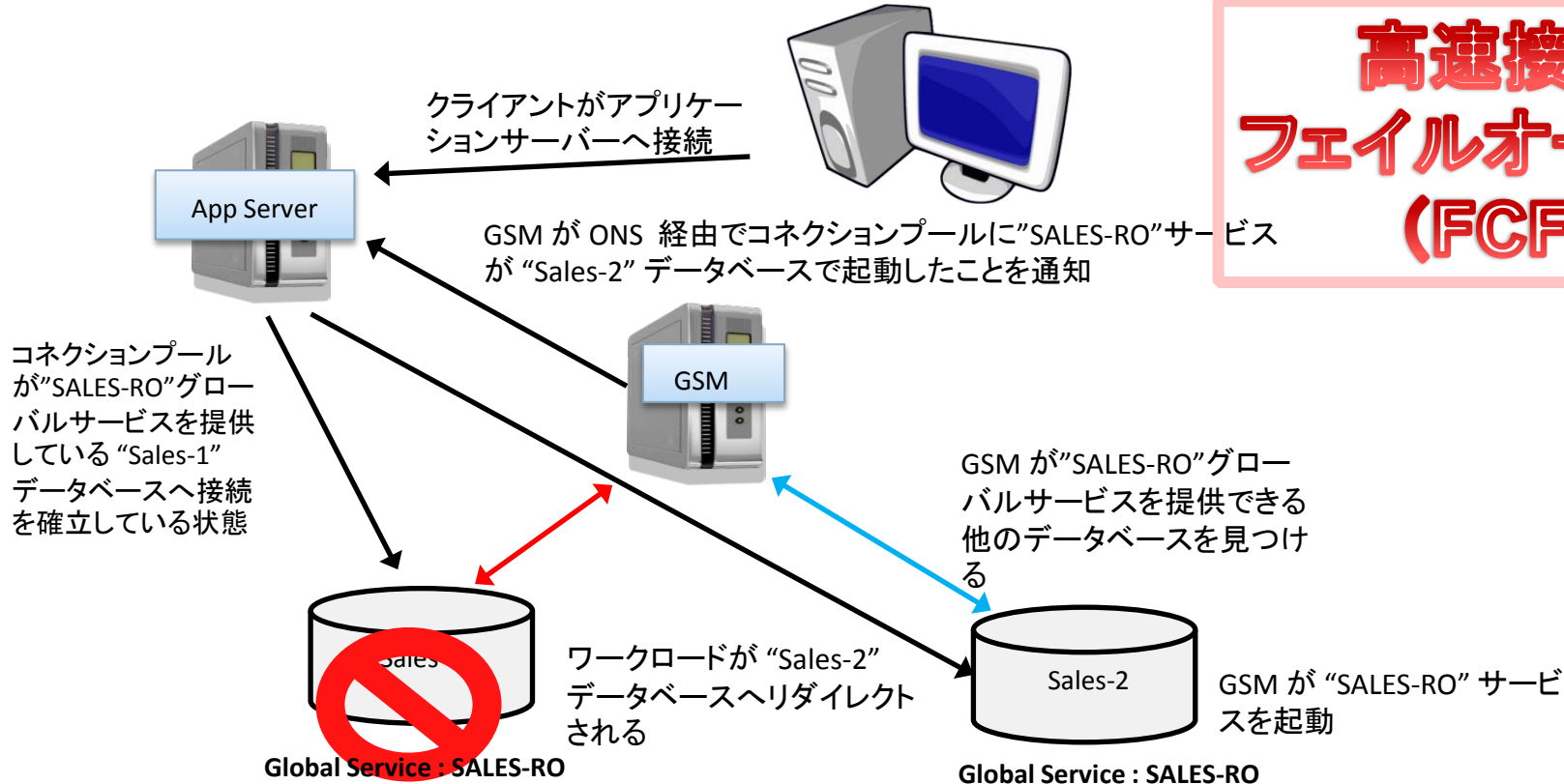
```
TNSNAMES.ORA  
(DESCRIPTION=  
(FAILOVER=on)  
(ADDRESS_LIST=  
(LOAD_BALANCE=ON)  
(ADDRESS=(WESTのGSM1つ目))  
(ADDRESS=(WESTのGSM2つ目)))  
(ADDRESS_LIST=  
(LOAD_BALANCE=ON)  
(ADDRESS=(EASTのGSM1つ目))  
(ADDRESS=(EASTのGSM2つ目)))  
(CONNECT_DATA=  
(SERVICE_NAME= read-service.salespool.oradbccloud)  
(REGION=WEST)))
```



SHORT, LONG  
REGION アフィニティ

# “グローバル・サービス”のフェイルオーバー

高速接続  
フェイルオーバー  
(FCF)



# “リージョン・アフィニティ” サービス属性

“グローバル”サービス<->クライアント、および、サービス<->DB 間

## 全リージョン

- デフォルト
- ロード・バランシングの際はネットワーク・レイテンシーが計算に入る

## ローカル・リージョン

- クライアントは自分自身が属しているリージョンで稼働しているサービスにのみ接続可能
- サービス・カーディナリティーはリージョン毎に保たれる

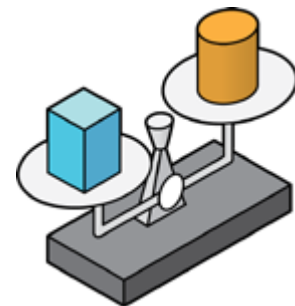
## Local with Inter-Regional Fail-Over

ローカルに該当サービスが一個も起動してない場合のみ他リージョンを探す



# Any Region – 全てのリージョンが選択対象

- デフォルト
- プール内各DBの負荷とそのDBが属するリージョンとクライアントリージョンのネットワークラウンドトリップタイム(RTT)を比較
- 全てのDBがほぼ同程度の負荷の場合、ローカルリージョンから選択される



# GDS の互換性、必要なバージョン

## サーバーなど

- GDS Pool に属する DB: 12.1 以上
  - いわゆる \_ (アンダースコア) init.ora パラメーターを使用するため
  - SPFILE 使用が必須
- カタログ DB: 12.1 以上
- 11gR2 以前の OCI, ODP.net は未サポート
  - 従来通り ローカル・リスナー を宛先に指定する必要あり

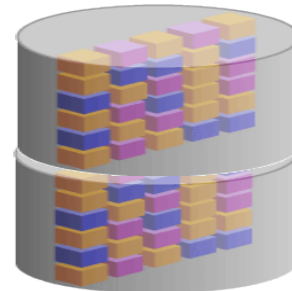
# GDS の互換性、必要なバージョン

## クライアント系

- 全ての 12c クライアントは Connection Load Balancing 利用可能
- Runtime Load Balancing をサポートするコネクションプール
  - OCI – 12.1
  - ODP.NET – 12.1
  - Thin JDBC – 11.2
    - ONS サーバーへの自動サブスクライブされないのでメソッド呼び出し追加が必要
  - Thin JDBC – 12.1
  - Weblogic Server – 12.1.2      Active GridLink データソース
- 11g OCI, ODP.net アプリはコード変更、再コンパイル、再リンクとも不要な場合があります

# GDS カタログ データベース

- DB12c が必要
- 1 つのGDS configuration(“oradbpool”, GDS構成)につき1つ必要
- GDS configuration に含める必要はない
- このDB自体も RAC, Data Guard, GoldenGate などで保護されることが推奨
- サイズは小さいので既存DB上に作成されることも想定されている
  - 例: EM12 Cloud Control OMS DB とスキーマを分け同居
  - 例: Recovery Manager カタログとスキーマを分け同居



# RAC DB 固有事項



- 12cR1 リリースではポリシーベース管理のみがサポートされる
- GDSプールに追加する際などの管理目的のために SCAN 設定が必要
- ユーザ・アプリケーションが GSM から接続要求リダイレクトを受けて接続する際は SCAN はバイパスされローカル・リスナーに接続される
- RLBアドバイザリーやノード、サービス Up/Down イベントは以前からと同じく ONS から FAN として GSM に定期的送信される
- gdsctl コマンド
  - GDSプール内に複数セットの RAC クラスターがあってもそれぞれ個別の設定が可能

# Data Guard 固有事項

- Broker が構成されている Data Guard のみ “フル” の機能がサポートされます。
  - ラグ属性
  - ロールベースサービス
- 1 つの “Broker構成” が 1 GDSプール になります
  - カスケード 構成にも対応
- Broker 構成の場合のスタンバイDBの追加・削除手順
  - 1. Data Guard BrokerのCLI(コマンドラインインターフェース): dgmgrl
  - 2. GDSCTL> sync brokerconfig // DG 専用コマンド

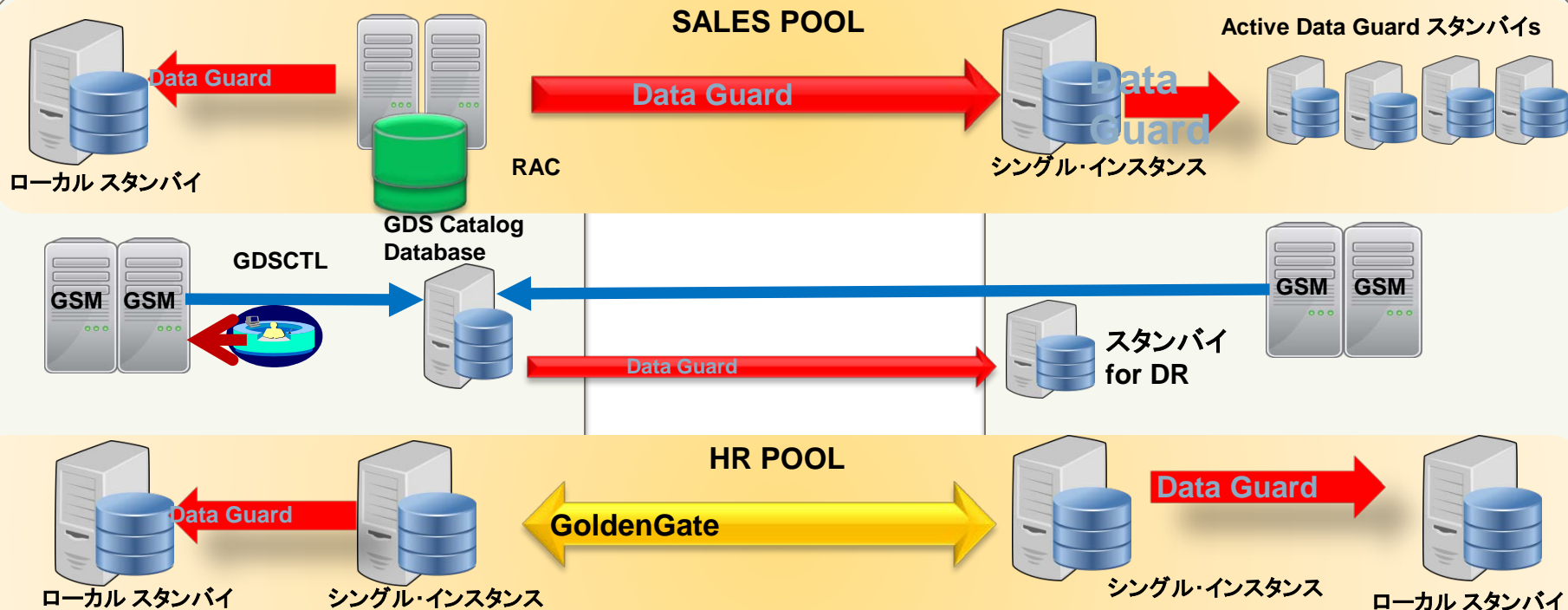
# DB12c 側での変更点: Global Serviceを提供するDB

- ユーザ/スキーマ “GSMUSER” などがデフォルトで作成されます
- GDS 環境に追加されるDBにはGDS中全てのGSMのアドレス(サーバ名、ポート名)が登録されます
  - gdsctl add database ...
  - LREG(Listener Registration)バックグラウンドプロセスがこれらのアドレスへ負荷(ロード)情報を送信

# Global Data Service/GSM 自体の高可用性を意識した配置

データセンター #1

データセンター #2



ORACLE



# Global Data Services 自体の高可用性

- リージョンにつき複数 GSM サーバーを推奨
- リージョンの中のある一つのGSMが“マスター”になる
- マスター GSM はクライアントへのFANイベントの配信を引き受け、ONSサーバーからイベントをパブリッシュする
- マスター GSM がダウンした場合、同じリージョン中の別のGSMが引き継ぐ
- リージョンの全ての GSM がダウンした場合、別リージョンの GSM が引き継ぐ
- 全リージョンの全GSMがダウンした場合、クライアントは旧来の“ローカル・リスナー”に直接つなぐ方式でDB接続が必要
- GDS カタログDB自体を HA/DR 目的のため Data Guard 構成を推奨
- GDSカタログDBがダウンしていてもGSMに対する多くの操作は可能
  - 例) クライアントからの接続を最適DBへリダイレクト

# アジェンダ

1. Maximum Availability Architecture (MAA)
2. 複製データを活用している環境での問題点
3. Global Data Services と RAC
4. 想定されるユース・ケース
5. グローバル・サービス
6. ロード・バランシング
7. Global Service Manager(GSM)
8. 動作詳細、設定、必要なバージョン
9. Q&A



**Hardware and Software**

**ORACLE®**

**Engineered to Work Together**

ORACLE®