

【Oracle 内に割当てられるリソース資源の使用制限の実施方法】 概要

リソースマネージャーを使用したリソースの使用制限機能

特定の SQL 文がリソースを独占してシステム全体に支障を与え、他の処理の実行のレスポンスが遅延するような現象があります

このような場合には、『リソースマネージャー』を使って、特定処理に割当てられる資源リソースの上限を指定して、他の処理にも資源リソースの割当てを残すようなデータベース全体での動作制限を行います

リソースマネージャーの使用制限が適用された場合のメッセージ

リソースマネージャーの使用制限が適用され、SQL 文の処理が途中で強制終了された場合には、エラーが発生し以下のようなメッセージが出力される

ORA-00040 : アクティブな時間制限を超えました - コールは異常終了しました
→ 対応としては、配分するリソースを増やして再実行する

リソースマネージャーによるリソースの使用制限を行う方法

まず、リソースプランを作成し、そのリソースプランを初期化パラメータ `RESOURCE_MANAGER_PLAN` に指定すれば、リソースプランの使用制限が適用されてデータベースが運用されることとなります

`RESOURCE_MANAGER_PLAN` 初期化パラメータの注意事項

- ・リソース使用制限のために有効にできるリソースプランは、1個だけである
- ・複数種類のリソースを使用制限のためには、1個のリソースプランに複数のリソースディレクティブ（制限内容）を紐付け、多数の種類の使用制限を設定する

`RESOURCE_MANAGER_PLAN` 初期化パラメータの設定方法

【手動設定方法】

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN =  
    '<リソースプラン名>' ;
```

【スケジュールを使用した自動時刻設定方法でのリソースプラン使用制限の適用】

```
sql> exec DBMS_SCHEDULER.CREATE_WINDOW(  
    window_name => '<スケジュール用ウィンドウ名>',  
    resource_plan => '<リソースプラン名>',  
    duration => interval '10' hour,  
    repeat_interval => 'FREQ = DAILY; BYDAY = MON, TUE, WED,  
        THU, FRI; BYHOUR = 22; BYMINUTE = 0; BYSECOND = 0',  
    window_priority => 'HIGH',  
    comments => 'COMMENT SENTENCE',  
);
```

【スケジュール自動時刻設定での変更指示が適用されない強制的な手動設定方法】

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN =  
    'FORCE:<リソースプラン名>' ;
```

- ・**FORCE** 指定でのリソースプラン適用は、リソースプランを適用した後、今後のスケジュール設定でのリソースプランの変更指示を無視して適用されないようになる
- ・**FORCE** 指定でのリソースプラン適用解除は、再度コマンドにて **FORCE** 指定無しでのリソースプラン適用を行うと、**FORCE** 指定が解除される

リソースマネージャー以外でのリソースの使用制限を行う方法

(インスタンス・ケージリング機能)

個々の **Oracle** インスタンスに対し、使用するCPUの個数を指定して、CPU割当てのリソース制御が行えます

このCPU割当てのリソース制御を行うためには、CPU 使用コア数指定の初期化パラメータである **CPU_COUNT** を使用する

リソースマネージャーが使用制限を行える対象の単位

- ・ユーザー単位
- ・セッション単位
- ・サービス単位

この制限単位とその対象者範囲を決めるために、リソースマネージャーでは『コンシューマ・グループ』を作成し、そこにリソース使用制限が行われる対象範囲を指定して定義します

リソースマネージャーの構成要素

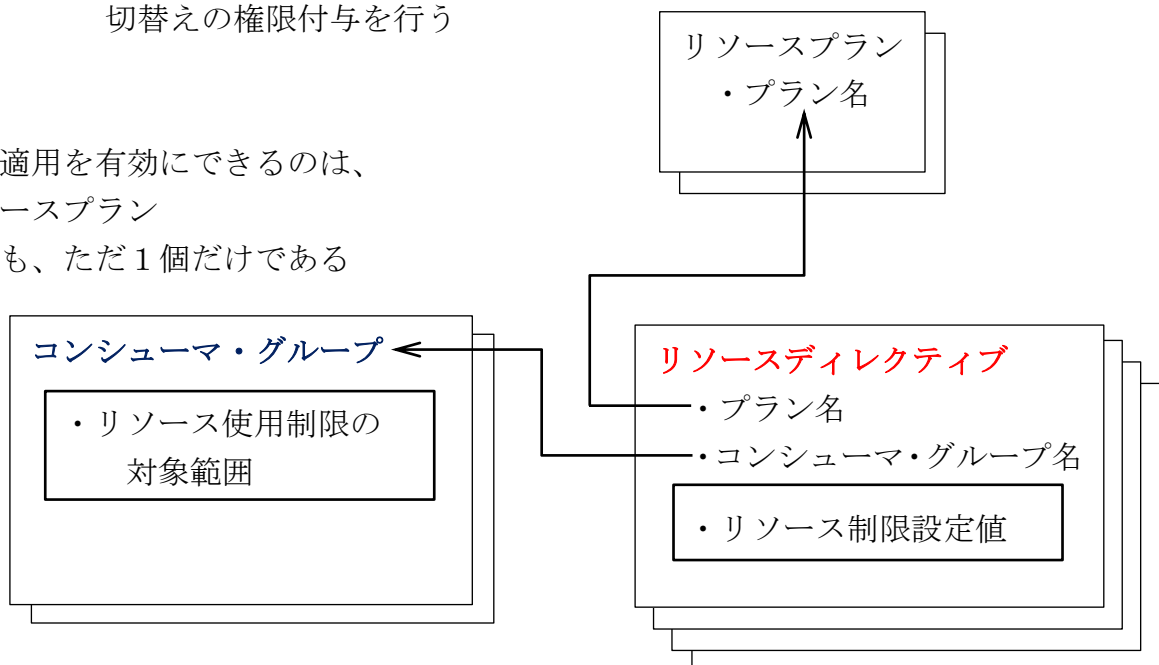
構成要素	要素の内容
コンシューマ・グループ	リソース管理対象とするグループ
リソースプラン	リソースプランの名称
リソースディレクティブ	リソース使用制限を行う具体的な設定の指定値情報
ペンディングエリア	作成中のリソースプラン情報を一時格納しておくエリア

※ ・リソースプランにその名称だけを決定して作成する。

~~このリソースプランにコンシューマ・グループ（リソース使用の制限の対象となる範囲）を紐付け、管理対象範囲を決める。~~

- ・コンシューマ・グループを単独で作成する
- ・リソース使用の制限値については、リソースディレクティブで具体的なリソース制限設定値を指定する。（リソースディレクティブ定義時）
- ・**リソースディレクティブで、リソースプランとコンシューマ・グループを指定して紐付ける**（リソースディレクティブ定義時）
- ・作成してあった**コンシューマ・グループへ、リソース使用の制限の対象となる範囲を設定する**
（ユーザー単位設定 or サービス単位設定 or セッション単位設定）
- ・コンシューマ・グループへ指定した対象範囲に対して、リソース使用制限切替えの権限付与を行う

制限適用を有効にできるのは、
リソースプラン
それも、ただ1個だけである



コンシューマ・グループで、『リソース使用制限の対象範囲』に指定できる種類とその種類を指定するための属性名（パラメータ名）

3) .コンシューマ・グループに『リソース使用制限の対象範囲』を設定
 DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
 (.

で使用

【表 4-11】

対象範囲指定の属性	説明
ORACLE_USER	接続 Oracle ユーザー名
SERVICE_NAME	クライアントが接続の確立に使用したサービス名
CLIENT_OS_USER	ログインしているクライアントの OS 側ユーザー名
CLIENT_PROGRAM	サーバーへのログインに使用されたクライアントのプログラム名
CLIENT_MACHINE	クライアントが接続に使用しているコンピュータ名

リソースディレクティブで、『リソース制限設定値』に指定する内容とその内容に対するパラメータ

4) .リソースプラン作成
 DBMS_RESOURCE_MANAGER.CREATE_PLAN(.
 で使用

【表 4-12】

リソース制限設定属性	説明
ACTIVE_SESS_POOL_P1	コンシューマ・グループ内で同時にアクティブにできるセッション数
SWITCH_GROUP	リソース制限設定値に達した時のアクション 'CANCEL_SQL': 処理中の SQL を強制終了させる (ロールバックが行われる) 'KILL_SESSION': セッションまで強制終了させる (ロールバックが行われる)
SWITCH_TIME	CPU 使用時間に対するリソース制限設定値 (秒)
UNDO_POOL	対象のコンシューマ・グループで使用するコミットされていないトランザクションの UNDO 合計使用量の制限値 (KB)
PARALLEL_TARGET_PERCENTAGE	対象のコンシューマ・グループで使用するパラレルサーバープールの最大値
MAX_UTILIZATION_LIMIT	対象のコンシューマ・グループで使用する最大 CPU 使用率の許可%値 CPU の合計使用率が 100%に満たない場合でも、動作制限適用

mgmt_p1 mgmt_p8	～	対象のコンシューマ・グループで使用する最大 CPU 使用率の許可%値 一度 CPU の合計使用率が 100%になってから制限動作が行われる。 それまでは、制限なし
--------------------	---	---

リソース制限設定値についての説明

設定されるのは、最大使用率である

よって、グループ別の設定値が、100%以上でも問題ない

設定される値は、複数のコンシューマ・グループが存在する場合に、CPU 使用が 100%になった時に配分に使われる割合ではない

どれか1つのコンシューマ・グループだけに配分が高いような場合もある

(ただし、指定した最大 CPU 使用率を超えて CPU を割り当てられることはない)

例) グループ	指定値	ある時点での実際の CPU 使用値
A	70	68
B	50	30
C	50	2
		合計 100

このような場合もある

リソース使用制限を行うための『リソースマネージャー』の設定手順概要

1) .ペンディングエリア作成

```
DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();  
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

2) .コンシューマ・グループ作成

```
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP( . . .
```

3) .コンシューマ・グループに『リソース使用制限の対象範囲』を設定

```
DBMS_RESOURCE_MANAGER.SET_INITIAL_CONSUMER_GROUP  
( . . . .
```

もしくは、

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING  
( . . . .
```

この使い分けは、コンシューマ・グループに行う『リソース使用制限の対象範囲』設定の種類によって決まってくる

この設定の操作は、手順9)の後で行っても構わない

4) .リソースプラン作成

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN( . . . . .
```

5) .リソースディレクティブ作成 (リソース使用制限の対象範囲の指定)

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE( . . . . .
```

6) .リソースプラン検証 (設定の有効性の確認)

```
DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

7) .リソースプラン有効化 (ペンディングエリアのクローズ)

```
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

8) .コンシューマ・グループへのリソース使用制限切替えの権限付与

```
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP( . . . . .
```

9) .完成したリソースプランの運用のための適用

【手動適用】

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN  
= '<リソースプラン名>' ;
```

もしくは、

【スケジュールを使用した自動時刻での適用】

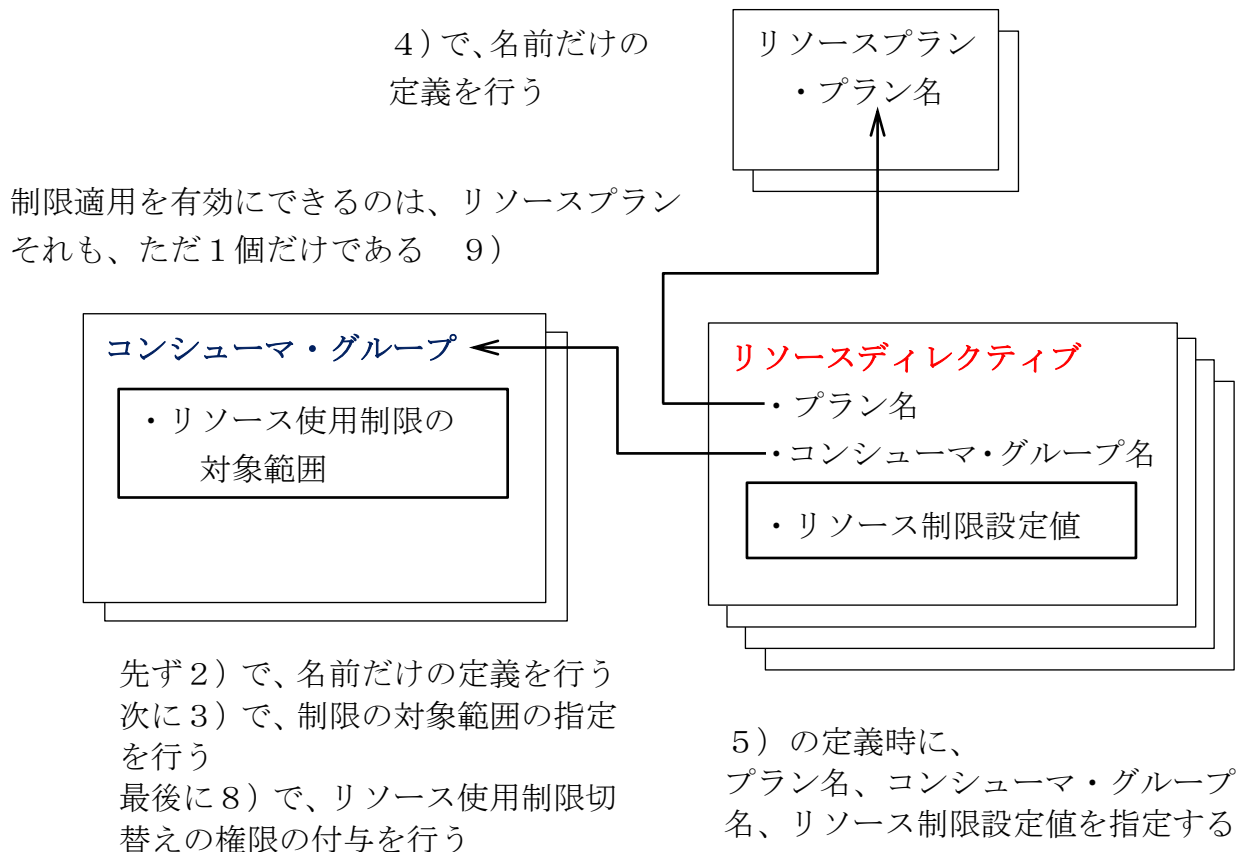
```
sql> exec DBMS_SCHEDULER.CREATE_WINDOW( . . . .
```

リソース使用制限のために有効にできるリソースプランは、1個だけ

10) .リソースマネージャーの設定内容確認

『リソース使用制限の種類』によって確認方法は、異なってくる

『リソースマネージャー』のコマンドの指定パラメータ説明



2) .コンシューマ・グループ作成

```
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(
  consumer_group => '<コンシューマ・グループ名>',
  comment => 'コメント'
);
```

3) .コンシューマ・グループに『リソース使用制限の対象範囲』を設定

```
DBMS_RESOURCE_MANAGER.SET_INITIAL_CONSUMER_GROUP(
  user => '<制限対象のユーザー名>',
  consumer_group => '<コンシューマ・グループ名>'
);
```

もしくは、

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(
  attribute => DBMS_RESOURCE_MANAGER.XXXXXX_XXXX,
  value => '<XXXXXX_XXXX に対する指定値>',
  consumer_group => '<コンシューマ・グループ名>'
);
```

XXXXXX_XXXX は、【表 4-11】の対象範囲指定の属性名

この使い分けは、コンシューマ・グループに行う『リソース使用制限の対象範囲』設定の種類によって決まってくる

4) .リソースプラン作成

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN(  
  plan => '<リソースプラン名>', ←  
  comment => 'コメント'  
);
```

5) .リソースディレクティブ作成 (リソース使用制限の対象範囲の指定)

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(  
  plan => '<リソースプラン名>', _____  
  group_or_subplan => '<コンシューマ・グループ名>', _____  
  switch_group => '<switch_group の選択値>', ←  
  switch_time => <値>, ←  
  switch_estimate => FALSE,  
  comment => 'コメント'  
);
```

ここで使用するパラメータは、
【表 4-12】にて説明

8) . コンシューマ・グループへのリソース使用制限切替えの権限付与

```
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUM  
ER_GROUP(  
  grantee_name => '<3) のコンシューマ・グループへの『リソース使  
  用制限の対象範囲』で USER に指定した値>',  
  consumer_group => '<コンシューマ・グループ名>', _____  
  grant_option => FALSE  
);
```

もしくは、

```
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUM  
ER_GROUP(  
  '<3) のコンシューマ・グループのマッピングで指定した VALUE 値>',  
  '<コンシューマ・グループ名>', _____  
  TRUE  
);
```

この使い分けは、3) でのコンシューマ・グループに行う『リソース使用制限の
対象範囲』設定の種類によって決まってくる

9) .完成したリソースプランの運用のための適用

1、2 ページ目に記述