

日々是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2021年5月31日月曜日

アプリケーションごとにリソース・マネージャーのコンシューマー・グループを切り替えてみる

Oracle DatabaseのEnterprise Edition限定ですが、リソース・マネージャーという機能があり、使用リソースの制御が可能になっています。Oracle Corporationが提供している無料で使える検証環境のapex.oracle.comでは、このリソース・マネージャーを使用して、リソースを有効活用しています。どのような設定を行なっているかは、[こちらの記事](#)が参考になります。

10年以上前の記事ですが、恐らく現在でも同様の制御を行なっていると思われます。

発行されたSQLは最初はコンシューマー・グループAPEX_HIGHで実行されます。CPUリソースの70%が使用可能ですが、CPU時間が10秒を過ぎると(つまり10秒以内に処理が終了しなければ)、コンシューマー・グループはAPEX_MEDIUMに切り替わります。

コンシューマー・グループAPEX_MEDIUMでは、CPUリソースの8%のみ使用可能で、CPU時間が120秒を過ぎると、コンシューマー・グループはAPEX_LOWに切り替わります。

コンシューマー・グループAPEX_LOWでは、CPUリソースの2%のみ使用可能で、CPU時間が1800秒(30分)を過ぎるとSQLはキャンセルされます。

その他、必須であるコンシューマー・グループOTHER_GROUPSやメンテナンス・タスクのための設定を含め、リソース・プランとしてAPEX_ORACLE_COM_PLANを作成して、初期化パラメーターresource_manager_planに設定しています。

データベース全体への設定は以上のようになりますが、今回はデータベース全体ではなく、アプリケーションごとに設定する方法を試してみます。

プラガブル・データベースPDB1にOracle APEXがインストールされていて、**ワークスペースAPEXDEV**にアプリケーションが作成されていることを前提とします。関連づけられているデータベースのスキーマはAPEXDEVです。

最初にリソース・プランMY_PDB_PLANを作成します。リソース・プランにはコンシューマー・グループRESTRICT_RUNAWAYを含みます。このコンシューマー・グループでは、SQLの経過時間が30秒に達すると、SQLをキャンセルします。

```
begin
  dbms_resource_manager.clear_pending_area();
  dbms_resource_manager.create_pending_area();

  dbms_resource_manager.create_consumer_group(
    consumer_group => 'RESTRICT_RUNAWAY');

  dbms_resource_manager.create_plan(
```

```

plan => 'MY_PDB_PLAN');

dbms_resource_manager.create_plan_directive(
  plan => 'MY_PDB_PLAN',
  group_or_subplan => 'RESTRICT_RUNAWAY',
  switch_group => 'CANCEL_SQL',
  switch_for_call => TRUE,
  switch_elapsed_time => 30);

dbms_resource_manager.create_plan_directive(
  plan => 'MY_PDB_PLAN',
  group_or_subplan => 'OTHER_GROUPS');

dbms_resource_manager.validate_pending_area();
dbms_resource_manager.submit_pending_area();
end;
/

```

PDB1にSYSで接続し、上記のスクリプトを実行します。

作成したプランはビューDBA_RSRC_PLANSから確認できます。

```
SQL> select * from dba_rsrc_plans where plan = 'MY_PDB_PLAN';
```

PLAN_ID	PLAN	NUM_PLAN_DIRECTIVES	CPU_METHOD	MGMT_METHOD	ACTIVE_SESS_POOL_MTH	
PARALLEL_DEGREE_LIMIT_MTH		QUEUEING_MTH	SUB_PLAN	COMMENTS	STATUS	MANDATORY
80044	MY_PDB_PLAN	2	EMPHASIS	EMPHASIS	ACTIVE_SESS_POOL_ABSOLUTE	
PARALLEL_DEGREE_LIMIT_ABSOLUTE		FIFO_TIMEOUT	NO			NO

```
SQL>
```

PDB1にリソース・プランを設定します。ALTER SYSTEMを発行します。

```
SQL> alter system set resource_manager_plan = 'MY_PDB_PLAN' scope=both;
```

System SETが変更されました。

```
SQL> show parameter resource_manager_plan
NAME                                TYPE    VALUE
-----
resource_manager_plan string  MY_PDB_PLAN
SQL>
```

ユーザーAPEXDEVがコンシューマー・グループRESTRICT_RUNAWAYを使用できる様に、権限を与えます。SYSで実行します。

```

begin
  dbms_resource_manager_privs.grant_switch_consumer_group (
    grantee_name => 'APEXDEV',
    consumer_group => 'RESTRICT_RUNAWAY',
    grant_option => FALSE );
end;
/

```

SQLがキャンセルされるかどうか、ユーザーAPEXDEVでPDB1に接続して確認します。

接続したのち、以下のスクリプトを実行して、セッションにコンシューマー・グループRESTRICT_RUNAWAYを設定します。

```
declare
  old_group varchar2(30);
begin
  dbms_session.switch_current_consumer_group('RESTRICT_RUNAWAY',
old_group, FALSE);
end;
/
```

確認のため、以下のSQLを実行します。

```
select count(*) from all_objects a, all_objects b;
```

30秒が経過するとORA-56735が発生し、SELECT文がキャンセルされます。

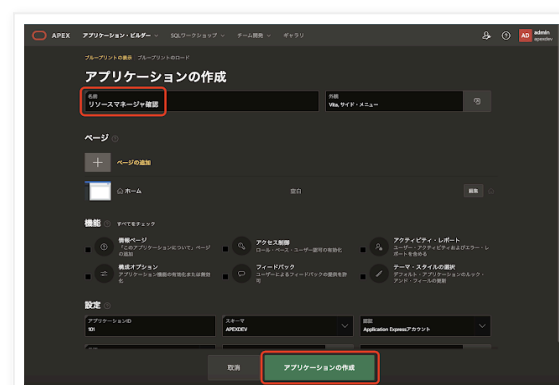
```
23:39:11 SQL> select count(*) from all_objects a, all_objects b;
select count(*) from all_objects a, all_objects b
                        *
ERROR at line 1:
ORA-56735: elapsed time limit exceeded - call aborted
```

23:39:47 SQL>

これで、リソース・プランMY_PDB_PLANとコンシューマー・グループRESTRICT_RUNAWAYが適切に設定されていることが確認できました。

では、コンシューマー・グループの設定をアプリケーションに適用してみます。

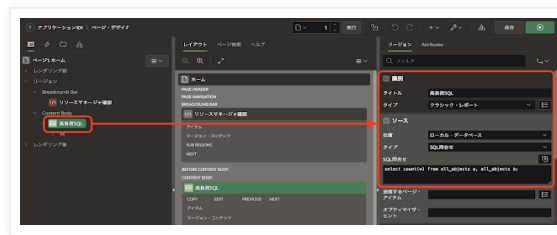
確認に使用するアプリケーションを作成します。アプリケーション作成ウィザードを起動し、空のアプリケーションを作成します。名前をリソースマネージャ確認とし、アプリケーションの作成を実行します。



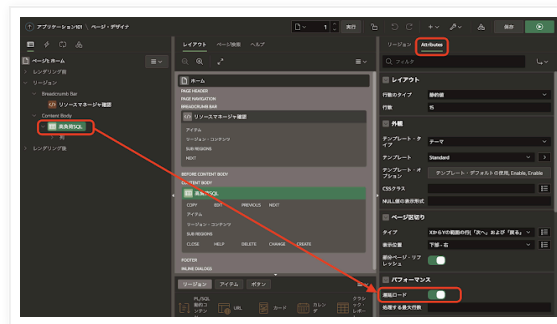
アプリケーションが作成されたら、ページ・デザイナーでホーム・ページ(ページ番号1)を開き、クラシック・レポートのリージョンを作成します。

リージョンの作成を実行します。識別のタイトルを高負荷SQLとし、タイプはクラシック・レポートとします。ソースの位置はローカル・データベース、タイプとしてSQL問合せを選択します。SQL問合せには、以下を指定します。

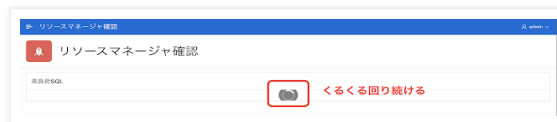
```
select count(*) from all_objects a, all_objects b;
```



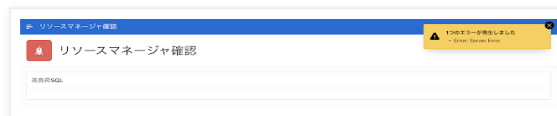
クラシック・レポートのリージョンのAttributesを開き、**遅延ロード**をONにします。クラシック・レポートの遅延ロードはOracle APEX 21.1で追加された新機能です。



この状態でアプリケーションを実行します。サインインの後、ホーム・ページが表示されますが、レポートの表示は延々終了しません。

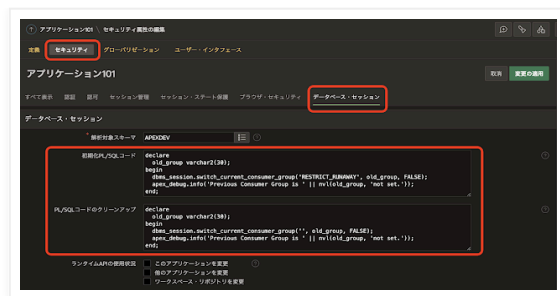


Oracle APEX 21.1以前で遅延ロードの設定ができない場合は、ホーム・ページの表示に延々と時間がかかり、表示されません。最終的にはエラーが発生します。



延々終了しないのも困るため、コンシューマー・グループRESTRICT_RUNAWAYで設定し、経過時間が30秒に達したらSQLの実行をキャンセルさせます。

コンシューマー・グループの設定は、**アプリケーション定義のセキュリティ**から行います。**データベース・セッション**のタブに含まれる設定から、コンシューマー・グループの切り替えを行います。



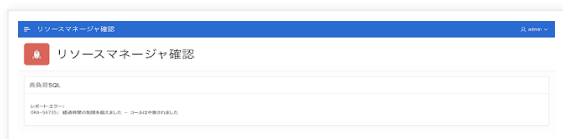
初期化PL/SQLコードとして、以下を記載します。dbms_session.switch_current_consumer_groupを呼び出して、コンシューマー・グループをRESTRICT_RUNAWAYに切り替えます。

```
declare
  old_group varchar2(30);
begin
  dbms_session.switch_current_consumer_group('RESTRICT_RUNAWAY', old_group, FALSE);
  apex_debug.info('Previous Consumer Group is ' || nvl(old_group, 'not set.'));
end;
```

PL/SQLコードのクリーンアップとして、以下を記載します。コンシューマー・グループの設定を外します。

```
declare
  old_group varchar2(30);
begin
  dbms_session.switch_current_consumer_group('', old_group, FALSE);
  apex_debug.info('Previous Consumer Group is ' || nvl(old_group, 'not set.'));
end;
```

上記の設定を行いアプリケーションを実行するとコンシューマー・グループによる制限が適用され、レポートの表示で「**ORA-56735: 経過時間の制限を超えました - コールは中断されました**」が発生します。



アプリケーションごとのコンシューマー・グループの切り替えは以上です。とはいえ、このままではページを初期化する時点でコンシューマー・グループが設定済みの場合に、クリーンアップのコードで元に戻すことができません。

初期化PL/SQLコードのヘルプ、PL/SQLコードのクリーンアップのヘルプに記載があるコンテキストを使用して、設定済みのコンシューマー・グループに戻す様にコードを拡張してみます。

PDB1にSYSで接続し、コンテキストの作成権限をユーザーAPEXDEVに与えます。

```
grant create any context to apexdev;
```

ユーザーAPEXDEVで接続し、コンテキストを作成します。この操作はOracle APEXのSQLワークショップのSQLコマンドからも実行できます。

```
create context ctx_consumer_group using ctx_consumer_group_pkg;
```

コンテキストの操作はパッケージCTX_CONSUMER_GROUP_PKGに含まれるプロシージャから行われます。DBMS_SESSION.SET_CONTEXTやCLEAR_CONTEXTを直接呼び出すことはできません。

パッケージCTX_CONSUMER_GROUP_PKGを作成します。SETとCLEARのふたつのプロシージャを定義します。

```
create or replace package ctx_consumer_group_pkg is
procedure set
(
    p_group in varchar2
);
procedure clear;
end;
/
```

パッケージ本体を作成します。

```
create or replace package body ctx_consumer_group_pkg is
procedure set(
    p_group in varchar2
)
is
    l_old_group varchar2(30);
begin
    apex_debug.info('consumer group is set to ' || p_group);
    dbms_session.switch_current_consumer_group(p_group, l_old_group, FALSE);
    apex_debug.info('preserve old consumer group ' || l_old_group);
    dbms_session.set_context('ctx_consumer_group','name',l_old_group);
end set;
procedure clear
is
    l_group varchar2(30);
    l_old_group varchar2(30);
begin
    l_group := sys_context('ctx_consumer_group','name');
    apex_debug.info('recover old consumer group ' || l_group);
    dbms_session.switch_current_consumer_group(l_group, l_old_group, FALSE);
    apex_debug.info('previous consumer group ' || l_old_group);
    dbms_session.clear_context('ctx_consumer_group');
end clear;
end ctx_consumer_group_pkg;
/
```

プロシージャ**SET**では指定されたコンシューマー・グループに切り替え、

```
dbms_session.switch_current_consumer_group(p_group, l_old_group, FALSE);
```

以前に設定されていたコンシューマー・グループをコンテキストctx_consumer_groupにnameとして保存します。

```
dbms_session.set_context('ctx_consumer_group','name',l_old_group);
```

プロシージャ**CLEAR**では、コンテキストctx_consumer_groupにnameとして保存されている値を取り出し、

```
l_group := sys_context('ctx_consumer_group', 'name');
```

コンシューマー・グループを以前のグループに戻し、

```
dbms_session.switch_current_consumer_group(l_group, l_old_group, FALSE);
```

コンテキストctx_consumer_groupをクリアしています。

```
dbms_session.clear_context('ctx_consumer_group');
```

アプリケーション定義のデータベース・セッションでは、それぞれ1行の記載に変更します。



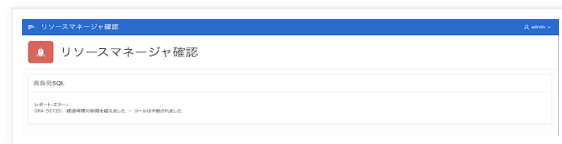
初期化PL/SQLコードでは、以下を呼び出します。

```
ctx_consumer_group_pkg.set('RESTRICT_RUNAWAY');
```

PL/SQLコードのクリーンアップでは、以下を呼び出します。

```
ctx_consumer_group_pkg.clear;
```

以上で設定の改変は完了です。アプリケーションを実行してみます。



コンシューマー・グループの設定は変わっていないため、ORA-56735が発生します。

今回はコンシューマー・グループの設定にコンテキストを使用しました。

実際のところコンテキストは、仮想プライベート・データベース - Virtual Private Database(VPD) - とともに利用することが想定されている機能です。Oracle APEXのアプリケーションで仮想プライベート・データベースを活用する方法については、機会を改めて紹介したいと思います。

今回作成したアプリケーションのエクスポートを以下に置きました。

<https://github.com/ujnak/apexapps/blob/master/exports/resource-manager.sql>

単にアプリケーションだけです。アプリケーションを動かすには、SYSで必要なコマンド(grant文など)を実行しておきます。

以上になります。Oracle APEXのアプリケーション作成の参考になれば幸いです。

完

Yuji N. 時刻: 16:24

共有

◀

ホーム

▶

[ウェブ バージョンを表示](#)

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

[詳細プロフィールを表示](#)

Powered by Blogger.