

日々は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。

2022年4月1日 金曜日

DBMS_CLOUD.EXPORT_DATAを使ったエクスポートとの速度比較

データベースの表データをCSV形式にして、オブジェクト・ストレージにエクスポートします。以下の2通りの方法で実施し、速度を比較してみました。

1. DBMS_CLOUD.EXPORT_DATAを使って、直接オブジェクト・ストレージにCSV形式でエクスポートする
2. APEX_DATA_EXPORT.EXPORTを使ってBLOBにCSVデータを書き出した後、DBMS_CLOUD.PUT_OBJECTを呼び出してアップロードする

最初の方法を実行するコードは以下です。

```
declare
  C_REGION constant    varchar2(20) := 'us-ashburn-1';
  C_NAMESPACE constant varchar2(20) := 'ネームスペースに置き換える';
  C_BUCKET constant    varchar2(20) := 'bricks';
  C_FILENAME constant  varchar2(20) := 'bricks';
  C_SQL constant varchar2(4000) := q'~select * from bricks where rownum < 10000001~';
  l_path      varchar2(400);
  l_operation_id number;
  l_status     varchar2(9);
  procedure set_checkpoint(ids varchar2)
  as
  begin
    dbms_output.put_line(systimestamp || ': ' || ids);
    dbms_session.set_identifier('csv_export-' || ids);
  end;
begin
  SET_CHECKPOINT('dbms_cloud.export_data');
  l_path := 'https://objectstorage.' || C_REGION || '.oraclecloud.com/n/' || C_NAMESPACE
    || '/b/' || C_BUCKET || '/o/' || C_FILENAME;
  dbms_cloud.export_data
  (
    credential_name => 'DEF_CRED'
    , file_uri_list => l_path
    , format => json_object(
      'type' value 'csv'
      , 'maxfilesize' value '214783648'
      , 'compression' value 'gzip'
```

```

    )
    , query => C_SQL
    , operation_id => l_operation_id
);
SET_CHECKPOINT('end');
for i in 1..10
loop
    select status into l_status from user_load_operations where id = l_operation_id;
    SET_CHECKPOINT(l_status);
    if l_status = 'COMPLETED' then
        exit;
    end if;
    dbms_session.sleep(1);
end loop;
end;
/

```

upload_obs_export_data.sql hosted with ❤️ by GitHub

[view raw](#)

2 番目の方法を実行するコードは以下です。

```

declare
    C_REGION constant    varchar2(20) := 'us-ashburn-1';
    C_NAMESPACE constant varchar2(20) := 'ネームスペースに置き換える';
    C_BUCKET constant    varchar2(20) := 'bricks';
    C_FILENAME constant  varchar2(20) := 'bricks';
    C_SQL constant        varchar2(4000) := q'~select * from bricks where rownum < 1000001~';
    l_context apex_exec.t_context;
    l_export apex_data_export.t_export;
    l_blob      blob;
    l_blob_src  blob;
    l_filename  varchar2(80);
    l_file      utl_file.file_type;
    l_buffer    raw(32767);
    l_amount    binary_integer := 32767;
    l_pos       integer := 1;
    l_blob_len  integer;
    l_path      varchar2(400);
    procedure set_checkpoint(ids varchar2)
    as
    begin
        dbms_output.put_line(systimestamp || ': ' || ids);
        dbms_session.set_identifier('csv_export-' || ids);
    end;
begin
    SET_CHECKPOINT('start');
    apex_session.create_session(

```

```

        p_app_id => 100
        , p_page_id => 1
        , p_username => 'APEXDEV'
    );
    SET_CHECKPOINT('apex_exec.open_query_context');
    l_context := apex_exec.open_query_context(
        p_location      => apex_exec.c_location_local_db
        , p_sql_query    => C_SQL
    );
$IF true $THEN
    SET_CHECKPOINT('apex_data_export.export');
    l_export := apex_data_export.export(
        p_context      => l_context
        , p_format      => apex_data_export.c_format_csv
        , p_file_name => C_FILENAME
    );
    l_blob_src := l_export.content_blob;
$ELSE
    SET_CHECKPOINT('apex_json');
    apex_json.initialize_clob_output;
    apex_json.open_object;
    apex_json.write_context(
        p_name => 'bricks'
        , p_context => l_context
    );
    apex_json.close_object;
    SET_CHECKPOINT('clob_to_blob');
    l_blob_src := wwv_flow_utilities.clob_to_blob(
        p_clob => apex_json.get_clob_output
        , p_charset => 'AL32UTF8'
        , p_include_bom => false
    );
$END
    SET_CHECKPOINT('utl_compress.lz_compress');
    l_blob := utl_compress.lz_compress(l_blob_src);
    SET_CHECKPOINT('utl_file');
    l_filename := C_FILENAME || '.csv.gz';
    l_blob_len := dbms_lob.getlength(l_blob);
    l_file := utl_file.fopen('DATA_PUMP_DIR',l_filename,'wb',32767);
    l_pos := 1;
    l_amount := 32767;
    while l_pos <= l_blob_len
    loop
        dbms_lob.read(l_blob, l_amount, l_pos, l_buffer);
        utl_file.put_raw(l_file, l_buffer, TRUE);
        l_pos := l_pos + l_amount;
    end loop;

```

```

utl_file.fclose(l_file);
SET_CHECKPOINT('dbms_cloud.put_object');
l_path := 'https://objectstorage.' || C_REGION || '.oraclecloud.com/n/' || C_NAMESPACE
        || '/b/' || C_BUCKET || '/o/' || l_filename;
dbms_cloud.put_object(
    credential_name => 'DEF_CRED'
    , object_uri => l_path
    , directory_name => 'DATA_PUMP_DIR'
    , file_name => l_filename
);
SET_CHECKPOINT('end');
apex_exec.close( l_context );
dbms_cloud.delete_file('DATA_PUMP_DIR',l_filename);
exception
    when others then
        apex_exec.close( l_context );
        raise;
end;
/

```

upload_obs_apex_data_export.sql hosted with ❤ by GitHub

[view raw](#)

作業自体は表BRICKSの作成なども含め、[こちらの記事](#)で作成した環境にて実施しています。

1番目の方法 - 行数10,000 - 約1秒

```

22-04-01 02:29:51.498847000 +00:00: dbms_cloud.export_data
22-04-01 02:29:52.537668000 +00:00: end

```

2番目の方法 - 行数10,000 - 約2.5秒

```

22-04-01 02:30:02.391526000 +00:00: start
22-04-01 02:30:02.397568000 +00:00: apex_exec.open_query_context
22-04-01 02:30:02.399717000 +00:00: apex_data_export.export
22-04-01 02:30:04.462289000 +00:00: utl_compress.lz_compress
22-04-01 02:30:04.487900000 +00:00: utl_file
22-04-01 02:30:04.589217000 +00:00: dbms_cloud.put_object
22-04-01 02:30:04.713105000 +00:00: end

```

1番目の方法 - 行数100,000 - 約1.5秒

```

22-04-01 02:31:06.601669000 +00:00: dbms_cloud.export_data
22-04-01 02:31:07.966000000 +00:00: end

```

2番目の方法 - 行数100,000 - 約25秒

```

22-04-01 02:31:15.171480000 +00:00: start
22-04-01 02:31:15.176659000 +00:00: apex_exec.open_query_context
22-04-01 02:31:15.178690000 +00:00: apex_data_export.export
22-04-01 02:31:39.284485000 +00:00: utl_compress.lz_compress
22-04-01 02:31:39.469500000 +00:00: utl_file
22-04-01 02:31:39.595496000 +00:00: dbms_cloud.put_object
22-04-01 02:31:39.778184000 +00:00: end

```

1番目の方法 - 行数1,000,000 - 約3秒

```

22-04-01 02:54:53.030251000 +00:00: dbms_cloud.export_data
22-04-01 02:54:56.010861000 +00:00: end

```

2番目の方法 - 行数1,000,000 - 約4分

```
22-04-01 02:32:12.818698000 +00:00: start
22-04-01 02:32:12.829158000 +00:00: apex_exec.open_query_context
22-04-01 02:32:12.831126000 +00:00: apex_data_export.export
22-04-01 02:36:08.515317000 +00:00: utl_compress.lz_compress
22-04-01 02:36:10.202244000 +00:00: utl_file
22-04-01 02:36:10.474738000 +00:00: dbms_cloud.put_object
22-04-01 02:36:11.024944000 +00:00: end
```

1番目の方法 - 行数10,000,000 - 約21秒

```
22-04-01 03:14:09.332436000 +00:00: dbms_cloud.export_data
22-04-01 03:14:30.916486000 +00:00: end
```

パッケージAPEX_DATA_EXPORTはクラシック・レポート、対話モード・レポートや対話グリッドのダウンロードの機能で使用されています。画面から100万行のデータをダウンロードするといったユース・ケースはあまりないので気にしたことはなかったですが、DBMS_CLOUD.EXPORT_DATAと比べるとかなり重い処理になっているようです。

APEX_DATA_EXPORT.EXPORTがサポートしているファイル形式はXLSX、PDF、HTML、CSV、XMLおよびJSONで、人間が利用するを想定していることがわかります。大量のデータを交換する用途には向いていようです。

列数にも依存関係があるかもしれないので一概には言えないですが、100万行で苦しい感じで1千万行は実行できないでしょう。

Autonomous Databaseで大量のデータをCSVやJSON形式で交換するには、DBMS_CLOUD.EXPORT_DATAの使用を優先的に検討する必要があるでしょう。

完

Yuji N. 時刻: 12:58

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.