

# 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。

2023年4月12日 水曜日

## DB 23cのLock Free Reservationを使ってみる

DB 23cよりLock Free Reservationと呼ばれる機能が追加されました。これは以下の操作を行います。

- 数値列の操作を加算と減算に限定します。
- 演算結果を更新する代わりに、差分をジャーナルに保存します。
- 値が書かれているブロックは変更しないため、ロックを取得しません。値自体は変更されたように見えます。
- 別のトランザクションが同じ行の同じ列を更新する場合も、ロックを取得せずに差分をジャーナルに保存するので待機が発生しません。
- トランザクションのコミット時に差分を適用します。
- SAGAが開始していて（DBMS\_SAGA.BEGIN\_SAGAが呼び出された後）、SAGAのロールバックが発生した（DBMS\_SAGA.ROLLBACK\_SAGAが呼びされる）場合は、自動的に補償トランザクションが実行されます。

この機能を実装してみて、実際に効果を確認してみます。SAGAの実装は大変なので、それは除きます。また、仕組みの詳細について解説することは、目的としていません。

効果の確認に、Oracle REST Data ServicesのRESTサービスを使います。

最初に表CALL\_COUNTERを作成します。

```
create table call_counter(  
  id number primary key,  
  count number  
);
```

あらかじめ1行データを投入します。主キーIDの値は1です。UPDATEを実行するときは、条件として主キーの値を指定する必要があります。

```
insert into call_counter(id, count) values(1, 0);
```

Lock Free Reservationを有効にする場合は、以下のように**reservable**を指定します。

```
create table call_counter(  
  id number primary key,  
  count number reservable  
);
```

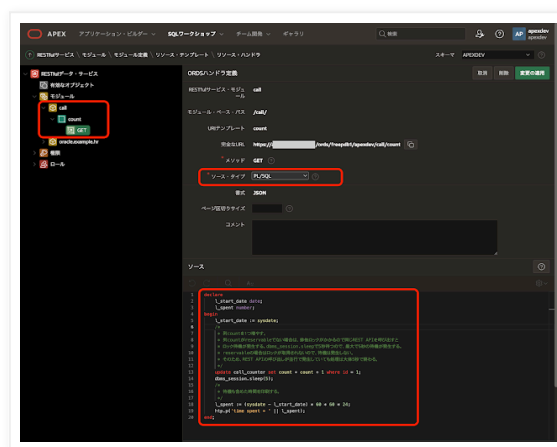
最初に、Lock Free Reservationなしで作業を進めます。以下のコードがGETハンドラの処理になります。

```
declare
    l_start_date date;
    l_spent number;
    l_count number;
begin
    l_start_date := sysdate;
    /*
    * 列countを1つ増やす。
    * 列countがreservableでない場合は、排他ロックがかかるので同じREST APIを呼び出すと
    * ロック待機が発生する。dbms_session.sleepで5秒待つので、最大で5秒の待機が発生する。
    * reservableの場合はロックが取得されないなので、待機は発生しない。
    * そのため、REST APIの呼び出しが並行で発生していても処理は大体5秒で終わる。
    */
    update call_counter set count = count + 1 where id = 1;
    select count into l_count from call_counter where id = 1;
    dbms_session.sleep(5);
    /*
    * 待機も含めた時間を印刷する。
    */
    l_spent := (sysdate - l_start_date) * 60 * 60 * 24;
    http.p('count = ' || l_count || ',time spent = ' || l_spent);
end;
```

lock-free.sql hosted with ❤ by GitHub

[view raw](#)

RESTfulサービスのモジュールとしてcall、テンプレートとしてcountを作成し、GETハンドラを作成します。ソース・タイプはPL/SQLを選択します。



テストの準備ができたので、同時にRESTサービス呼び出してみます。RESTサービスの待機時間を含む処理時間は、5、9、12、16秒と増えていきます。countは52、53、54、55となっています。



```
alter table call_counter modify (count reservable);
```

ほとんどのリクエストが5秒で終了します。ロック待機が発生していないことが確認できます。また、逐次処理にはなっていないため、`count`は全部同じ値（以下では55）です。次にRESTサービスを呼び出すと、`count`は4が加わり59になります。



reservableとなっている列がある表をドロップしようとする、ORA-55764が発生します。そのため、表をドロップする前に列をnon reservableに変更する必要があります。

```
SQL> drop table call_counter;
drop table call_counter
*
```

```
ERROR at line 1:
ORA-55764: Cannot DROP or MOVE tables with reservable columns. First run "ALTER
TABLE <table_name> MODIFY (<reservable_column_name> NOT RESERVABLE)" and then
DROP or MOVE the table.
```

SQL&gt;

以上で、Lock Free Reservationの説明は終了です。

ここまで極端なホットスポットはあまり無いと思います。とはいえ加算と減算に操作を限定できる数値列、例えばページの訪問数やいいねをクリックした数のようなデータがある場合は、適用を検討する価値はあるでしょう。

完

[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](#).

---