Making plans with Oracle 11g

and not leaving them behind

Author: Jed Walker is the Manager of Database Operations for the Comcast Media Center in Centennial Colorado. He has been working with Oracle Database since 1997 and is an Oracle Certified Professional for 9i, 10g, and 11g Database.

Table of Contents

Introduction

Starting with Oracle 11g the database now has a component called the SQL Management Base (SMB). The SQL Management Base contains SQL Plan baselines. SQL Plan Baselines are a known plan for a given SQL statement. Oracle stores these plans and then uses them to ensure optimal query performance. SQL Plan Baselines often significantly improve query performance without making any changes to your schema or database configuration. Once you've got these performance improving plans stored in the SMB you don't want to leave them behind. The purpose of this paper is to point out some situations when you would want to migrate your SMB and how to do it.

Basics of SQL Plan Baselines

In case you aren't familiar with SQL Plan Baselines and the SQL Management Base, I'll cover some basics. As queries are requested from the database, Oracle will use the Cost Based Optimizer (CBO) to determine an execution plan and cost and then store it in the SMB. If the cost matches an accepted plan in the SMB then it will use that plan, otherwise it will compare the cost with the costs of other plan baselines. If it finds a baseline with a lower cost it will use that baseline for the query. This helps ensure that your queries are performing optimally and that you don't experience performance degradation. Over time Oracle can continue to improve upon the performance of the queries, and also make sure that if a new plan would degrade performance that an acceptable plan is used instead.

In order for Oracle to provide this, you must have a few parameters set correctly. Plans can be loaded into the SMB manually or automatically. You can load them manually a few different ways and you can find that information in the manual if you want to do it. If you want them to be loaded automatically you must set the parameter optimizer_capture_sql_plan_baselines = true. For Oracle to use the SQL Plan Baselines (as described above) you must set optimizer_use_sql_plan_baselines = true.

I can't count how many times I've seen a poorly performing query in Oracle 11g that was fixed by a quick trip to OEM to implement a new SQL Plan. I'm sure there were many more that were automatically implemented during maintenance windows. I definitely recommend trying out this feature if you haven't.

SMB Space Usage

In case you haven't tried this feature yet, you should be aware that the SMB is stored in the SYSAUX tablespace, so be prepared to have it grow. I'm not aware of a neat way to determine how much space you'll need to plan for because I believe it depends on many factors. I've seen some databases that barely use any extra space at all, and some that have had SYSAUX grow like mad. We had a system with a poor storage system that experienced rapid growth - I believe because it was simply trying hard to find a good plan. That database rapidly chewed through 15+ GB of SYSAUX space when on local storage. Once we built the database on our permanent storage solution the total SYSAUX used space was only about 3GB. So, regardless of the size of the database, be aware that you might need to allocate a little, or a lot, of extra space. You can manage some of this by using the DBMS_SPM.CONFIGURE command to change the space usage and retention policies for SQL Plans.

```
-- don't allow the SMB to use more than 25% of SYSAUX

execute dbms_spm.configure('space_budget_percent',25);

-- remove unused SQL Plans after 12 weeks

execute dbms_spm.configure('plan_retention_weeks',12);
```

Going on a trip?

When you're taking your data on a trip you'll want to pay attention and not leave your SMB behind. There are many ways of moving data around and you should be aware of your SMB when doing it. Depending on how you go about moving your data you could have some great SQL Plan Baselines, but leave them behind. If you move your schemas, or create schemas from scripts in a new database these SQL plan baselines will have to be created and then evolved in a maintenance window (or by you) and thus won't be available when you first start your database. If you're going to production you'll not want to leave those performance improvements behind.

If you're using RMAN duplication then you should be set because you're getting an exact copy of the physical structure including your SYSAUX tablespace and your SMB. I always like to use RMAN if I can because it is a sure-fire way to get exactly the same database. If you don't have that luxury, then just migrate your SMB.

If you use scripts, Data Pump, SQL Loader, or other database movement tools to move data from one system to another, you need to be aware that your SQL Plan Baselines do not come along for the ride. If you export your schemas to move them to test and then production, or to move production data back to test or development, you'll want to migrate your SMB. If you don't, then all the work to create those has to be done again, possibly meaning initial performance may suffer.

The following demonstrates that DataPump won't bring your SMB with the schema.

We have a schema, MY_USER, with 3 enabled and 37 accepted SQL Plan baselines

```
SQL> select creator, enabled, accepted, fixed, count(1)

from dba_sql_plan_baselines

where creator = 'MY_USER'

group by creator,  enabled, accepted, fixed

 /

CREATOR      ENA ACC FIX   COUNT(1)

------------ --- --- --- ----------

MY_USER    YES NO  NO           3

MY_USER    YES YES NO          37
```

We export the schema using Data Pump

```
->expdp dumpfile=it.dmp logfile=it_exp.dmp schemas=MY_USER
```

```
Export: Release 11.2.0.2.0 - Production on Wed Mar 2 15:05:12 2011

…

Job "SYSTEM"."SYS_EXPORT_SCHEMA_01" successfully completed at 15:07:06
```

We import the schema into another database using Data Pump

```
->impdp dumpfile=it.dmp logfile=it_imp.log remap_tablespace=adn_data:users

Import: Release 11.2.0.2.0 - Production on Wed Mar 2 15:27:51 2011

…

Job "SYSTEM"."SYS_IMPORT_FULL_01" successfully completed at 15:17:41
```

We then check for SQL Plan Baselines in the new database

```
SQL> select creator, enabled, accepted, fixed, count(1)

from dba_sql_plan_baselines

where creator = 'MY_USER'

group by creator,  enabled, accepted, fixed

 /

no rows selected
```

So, while we have our schema loaded, the SQL Plan Baselines are not there. That means risking poor performance until plans have been recreated and evolved all over again, or …

Take the SMB with you

Follow these steps to take all, or some of, your SQL Plan Baselines to another database.

First, you need to create a staging table.

```
begin

dbms_spm.create_stgtab_baseline(table_owner=>'MY_USER',

                                table_name=>'SMB_TRIP');

end;

/

PL/SQL procedure successfully completed.
```

Now you can start loading plans from the SMB for any user you plan on taking with you. When loading your plans you need to specify one of the parameters for enabled, accepted, and fixed depending on what baselines you want to take with you. Also, note that SQL Plan Baselines are stored by the user who ran the query, not the schema in which the objects belong (which could be multiple). If you have application users you will probably

want to pack their plans also. All of these can be packed together in the same table, so run the pack_stgtab_baseline function for each user you want to bring baselines for. I also like to run a query after loading to make sure the plans I wanted are packed.

```
declare
  my_plans number;
begin
  my_plans:=dbms_spm.pack_stgtab_baseline(table_owner=>'MY_USER',
                                          table_name=>'SMB_TRIP',
                                          creator=>'MY_USER',
                                          enabled=>'YES');
end;
/
PL/SQL procedure successfully completed.


select creator, count(1) from my_user.smb_trip group by creator
/
CREATOR                        COUNT(1)
------------------------------ ----------
MY_USER                                40
```

Now that you have these loaded, you should use Data Pump to export the table and load it into the database you are moving to. Once the table is imported you can unload all of the SQL Plan Baselines from the staging table at once. The unpack function will retrieve the baselines for all users that you packed in the table.

```
prompt Unpack Staging Table into SMB
declare
  my_plans number;
begin
  my_plans:=dbms_spm.unpack_stgtab_baseline(table_owner=>'MY_USER',
                                            table_name=>'SMB_TRIP',
                                            enabled=>'YES');
end;
/
```

```
PL/SQL procedure successfully completed.


select creator, enabled, accepted, fixed, count(1) from dba_sql_plan_baselines
group by creator, enabled, accepted, fixed order by 1

/

CREATOR                        ENA ACC FIX   COUNT(1)

------------------------------ --- --- --- ----------

MY_USER                       YES NO  NO             3

MY_USER                       YES YES NO            37

…
```

Now you have the plans that you and Oracle worked so hard to create. So, next time you pack for a trip, pack your SMB and take it with you!