Estimating Greenplum Database 4.2.x to 4.3.x Migration Time

Rev: A01

Updated: March, 2015

Overview

Append-optimized tables are introduced in Greenplum Database 4.3.0. If your Greenplum Database 4.2.x databases contains append-only tables, the migration process from Greenplum Database 4.2.x to 4.3.x converts append-only tables to append-optimized tables. For information about append-optimized tables, see the release notes for Greenplum Database 4.3.0.

For a Greenplum Database 4.2.x database that contains a large number of append-only tables, the conversion to append-optimized tables might take a considerable amount of time. You can install the `estimate_42_to_43_migrate_time()` user-defined function in a Greenplum Database 4.2.x database that contains append-only tables to estimate the time required to migrate the database to a 4.3.x database. The estimate includes the estimated time required to convert append-only tables to append-optimized tables.

The function returns an estimated time for the migration for planning purposes based on the number of append only objects in the databases being migrated, the number of heap objects in the database, as well as the number of segments in the configuration. The size of data in each object does not effect this calculation, only the number of objects. Other factors might impact the overall migration time.

Installing and Running estimate_42_to_43_migrate_time

Follow these steps to estimate the migration time for a Greenplum Database 4.2.x installation:

1. Download the zip file that contains the user defined function from the Greenplum Database Documentation site. *estimate_42_to_43_migrate_time.zip*
2. Extract the SQL script.
3. Create the function in each Greenplum Database 4.2.x database.
4. Connect to each database and execute the function.
5. Add the time estimate for each database for an estimated total migration time.

Example

This example creates the function `estimate_42_to_43_migrate_time()` in the database, `mytest`:

```
$ psql -d mytest -f estimate_42_to_43_migrate_time.sql
```

This example runs the `estimate_42_to_43_migrate_time()` function. The example assumes that the function was installed in the database, `mytest`:

```
$ psql -d mytest -c "select estimate_42_to_43_migrate_time();"
                    estimate_42_to_43_migrate_time
-------------------------------------------------------------------------
estimate_42_to_43_migrate_time() version: 0.3 run at 2014-11-17 11:40:11
GPDB version: PostgreSQL 8.2.15 (Greenplum Database 4.2.8.2 build 1)
  on x86_64-unknown-linux-gnu,
  compiled by GCC gcc (GCC) 4.4.2 compiled on Oct 8 2014 14:55:11
Database: mytest
Number of primary segments: 96
Num of AO objects: 11100
Num of heap objects: 3005
Estimated migrate time: 00:57:56
(7 rows)
```

Notes

The `estimate_42_to_43_migrate_time()` function must be created in each database and executed for each database. The function uses the *pg_class* table and each database has its own copy of *pg_class*. The function does not need to run against the `template0`, `template1`, and `postgres` databases unless user created objects exist in the databases.

The formulas in this function have been created based on testing with the EMC Greenplum DCA products. Tests were performed using both the V1 and V2 EMC Greenplum DCA systems.

Monitoring the append-only Object Conversion Progress

The Greenplum Database migration process output does not provide the status or the specific number of append-only objects that have been converted to append-optimized objects. The process only reports the databases that have been migrated. During the migration, the migration utility displays the `Performing catalog transformation` step for the migration of each database. The catalog transformation step converts append-only objects and will most likely be the longest step in the migration. This example migration output is from a small test Greenplum Database installation:

```
20141103:09:07:14:024320 gpstart:dca6-mst1:gpadmin-[INFO]:-Check status of
 database with gpstate utility
20141103:09:07:14:011329 gpmigrator_mirror:dca6-mst1:gpadmin-[INFO]:-
Performing catalog transformation
20141103:09:07:14:011329 gpmigrator_mirror:dca6-mst1:gpadmin-[INFO]:-...
 template1
20141103:09:07:16:011329 gpmigrator_mirror:dca6-mst1:gpadmin-[INFO]:-...
 template0
20141103:09:07:19:011329 gpmigrator_mirror:dca6-mst1:gpadmin-[INFO]:-...
 postgres
20141103:09:07:21:011329 gpmigrator_mirror:dca6-mst1:gpadmin-[INFO]:-...
 mytest
```

```
20141103:09:38:22:011329 gpmigrator_mirror:dca6-mst1:gpadmin-[INFO]:-
Shutting down new Greenplum postmaster
```

To monitor the progress of the catalog transformation step of the migration process, you can run a simple `tail` command on the current Greenplum Database log file in the directory `$MASTER_DATA_DIRECTORY/pg_log`. This example `tail -f` command shows the progress of a migration by displaying the messages that Greenplum Database migration utility appends to the Greenplum Database log file to the command prompt.

```
$ tail -f gpdb-2014-12-17_143740.csv
```

Start the `tail` command before you start the Greenplum Database migration utility and substitute the current log file name for `gpdb-2014-12-17_143740.csv`.

During the migration process, you can use the `grep` and `wc` utilities to count number of append-only objects that have been converted to append-optimized objects. This example command returns the number of append-only objects that have been converted by searching for the string `pg_catalog.upgrade_appendonly_aux`:

```
$ grep pg_catalog.upgrade_appendonly_aux gpdb-2014-12-17_143740.csv | wc -l
```

When you run the command, substitute the current log file name for `gpdb-2014-12-17_143740.csv`. When you run the command multiple times during the migration process, the number increases until all the append-only objects that have been converted. The count is for all databases in the Greenplum Database installation and does not separate counts for each database.

For more information about the `grep`, `wc`, and `tail` commands, see your operating system documentation. For more information about the Greenplum Database log file, see "Viewing the Database Server Log Files" in the *Greenplum Database Administrator Guide*.

> **Note:** On most systems, a new log file is created for each day due to automatic log file rotation. If the upgrade process runs over midnight, a new log file might be created. If that is the case, the `grep` command must be run on each log file to find the total number of append-only objects that were converted.