Architecture to Auto Refresh SAS® MOLAP Cubes

Gaurav K Agrawal, GCE Solutions Inc, USA

ABSTRACT

OLAP Cube is one of the very smart concepts of Business Intelligence environment to do fast business reporting with minimal load on servers. Because of the capabilities of OLAP Cubes, day by day Cubes counts are getting increased in all type of industries. SAS supports all three types of cubes like HOLAP, ROLAP and MOLAP. MOLAP is most commonly type of cube used in various industries. This paper is more focused around SAS MOLAP Cubes.

MOLAP cubes maintain its multi dimensional aggregated data separate from the main database. Hence to get proper reporting data it's must to get MOLAP Cubes refreshed properly and in timely manner. When we say timely then definitely it implies that as soon as input data is refreshed then Cube(s) also should be refreshed. Hence we should define a robust and reliable architecture in such a way that Cubes are being refreshed automatically and always ready for reporting on latest data.

SAS has provided options to refresh Cube(s) through SAS code and/or through Management Console. However in order to get it refreshed automatically there is requirement of smart architecture as per business need.

1. INTRODUCTION

This paper will discuss about different methods to refresh Cubes. In addition to that paper will discuss the challenges faced in Cube refresh with solutions around it. Paper will also be discussing a detailed architecture, which can be followed to achieve different business requirement around MOLAP Cube refresh. Following topics will be covered through this paper:

- ✓ Methods to Refresh SAS OLAP Cubes
- ✓ Challenges in SAS OLAP Cube Refresh
- ✓ Solution of challenges for SAS OLAP Cube Refresh
- ✓ Detailed Architecture for Auto Cube Refresh

2. METHODS TO REFRESH SAS OLAP CUBES:

There are two well known methods to refresh SAS OLAP Cubes. Below mentioned are those

- ➢ Through SAS OLAP Cube Studio
- ➢ Through SAS code

Note: As this paper is focused on auto refresh of the Cube, so we'll not discuss much about the steps of manual refresh of cube.

2.1 CUBE REFRESH BY OLAP CUBE STUDIO

After logging in SAS OLAP Cube studio, user will see a complete list of OLAP cubes created. Just right click on the cube and click “Create” in order to refresh the cube. By this method Cube will be refreshed manually but ensure there is no active connection with cube. If there is any active connection with OLAP cube then that should be disconnected using Management Console. So all these are manual steps need to be followed in order to refresh the cube.

Now if we have big number of cubes in our organization then it’s not easy to follow these manual steps in order to refresh the cubes. This creates a need of some type of automation process for the cube refresh.

2.2 CUBE REFRESH BY SAS CODE

In order to refresh SAS Cube it is always required to check the physical part of the Cube. If Physical data part exists then first delete and then go for the refresh of the code. Below mentioned code can be used to delete the physical part of the cube

```
PROC OLAP cube       =&cubename DELETE_PHYSICAL;
     METASVR host    =<HOSTName> port=8561 protocol=bridge
     Userid          =<METAUSER>
     Pw              =<METAPASS>
     Repository      =<Foundation>
     olap_schema     =<OLAP_schema>;
RUN;
```

After deleting the physical part, Cube can be refreshed using below mentioned code

```
PROC OLAP cube       =&cubename ;
     METASVR host    =<HOSTName> port=8561 protocol=bridge
     Userid          =<METAUSER>
     Pw              =<METAPASS>
     Repository      =<Foundation>
     olap_schema     =<OLAP_schema>;
RUN;
```

Parameters in Cube Refresh Code

S. No,	Parameter	Description
1.	HOSTNAME	Host Name of the machine where Metadata Server is running
2.	METAUSER	User name who has authority to refresh and deleted the physical part of Cube
3.	METAPASS	Encoded password of the User
4.	Foundation	Repository Name where OLAP Cube is residing
5.	OLAP_SCHEMA	Schema Name in which OLAP Cube is residing

In order to get Cube refreshed automatically we can use above mentioned code and schedule using Unix Cron facility or Windows scheduler or other scheduling utilities.

3. CHALLENGES IN SAS OLAP CUBE REFRESH:

There are certain challenges in getting Cube refreshed by above mentioned methods. Below mentioned are those challenges

- If OLAP user leaves an active connection with OLAP Cube then cube refresh will fail
- What should be frequency of Cube refresh and what if cube refresh dependent on successful completion of other job

4. SOLUTION OF CHALLENGES FOR SAS OLAP CUBE REFRESH

4.1 ACTIVE CONNECTION WITH OLAP CUBE

Solution for the challenge can be to remove any existing connection on the cube before proceeding with cube refresh. In order to remove any active connection, most recommended approach is to write a Java code, which will interact with Metadata and will remove any active connection on the Cube. As soon as all active connections are removed, cube can be refreshed without any problem.

4.2 SCHEDULING OF CUBE REFRESH WHERE INPUT DATA DEPEND ON OTHER JOB

Now assume we have our Java code ready to remove any active connection and SAS code ready to refresh the cube. But next question arise that what should be the frequency of the Cube refresh. Is there any dependency of Cube refresh on any other job, etc?

Recommendation is to scheduling cube refresh either of the two ways

1) Schedule Cube Refresh - Schedule a Cube refresh at a particular defined time
2) Trigger file based – By this approach dependent job will create a flat file after successful execution. This trigger file will work as a flag to Cube refresh job that previous job has run successfully and data is ready to refresh cube. Already running job on system will check for the trigger file and will refresh the cube with new data using Java and SAS code.

5. DETAILED ARCHITECTURE FOR AUTO CUBE REFRESH

Below mentioned is the detailed architecture solution/suggestion for the auto cube refresh. Architecture will change as per the requirement of business. So it is just a proposed base for the architecture.

5.1 FLAT FILE WITH PARAMETERS

First define two parameters files in UNIX

a) As Global parameters file – This file will have all the required fields for the SAS environment and all general parameters required for UNIX scripting. Whatever parameter we can think of putting as global will be available in this file as central repository.

b) Cube Refresh file with below parameters – All parameters related to Cube information will reside in this file. All the information like as below

- Cube Name
- Is Trigger File Required
- Output Log file
- Input Library details
- Schema Name
- Frequency of Cube Refresh
- Etc. (Depend on Company/Business requirement)

5.2 SAS CODE

```
%macro Cube_Refresh;

/* The date and time when Cube refresh was started */

data _null_;
Job_Date_Time= datetime( ) ;
put "Cube Refresh Start Time " Job_Date_Time datetime16. ;
run;

%put &sysparm;

/* Initialize all parameter with values coming from the Unix
schell script as parameter */

%let param_libref        =    %scan(&sysparm,1,' ');
%let param_libname       =    %scan(&sysparm,2,' ');
%let param_cubename      =    %scan(&sysparm,3,' ');
%let param_cubelocation  =    %scan(&sysparm,4,' ');
%let param_isCubeExist   =    %scan(&sysparm,5,' ');
%let param_schema        =    %scan(&sysparm,6,' ');


LIBNAME &param_libref BASE "&param_libname" ;

%let cubename            =    &param_cubename;

/*Check if physical part exist then delete the physical part of
the cube. For your information parameter “param_isCubeExist” is
passed for Unix shell script confirming that physical part of
the cube exist or not*/

%if &param_isCubeExist   = 1 %then %do;

     PROC OLAP cube      =    &cubename DELETE_PHYSICAL;
          METASVR host="&HOSTID" port=8561 protocol=bridge
          Userid         =    "&METAUSER"
          Pw             =    "&METAPASS"
          Repository     =    "Foundation"
          olap_schema    =    "&param_schema";
     RUN;
%end;

/* Below mentioned Code will be used to refresh the Cube*/

PROC OLAP cube           =    &cubename ;
   METASVR host          =    "&HOSTID" port=8561
protocol=bridge
```

```
            Userid        =    "&METAUSER"
            pw            =    "&METAPASS"
            repository    =    "Foundation"
            olap_schema   =    "&param_schema";
RUN;

/* The date and time when Cube refresh was finished */

data _null_;
Job_Date_Time             = datetime( ) ;
put "Cube Refresh End Time " Job_Date_Time datetime16. ;
run;

%mend Cube_Refresh;

%Cube_Refresh;
```

5.3 JAVA CODE

There are Java APIs available to connect with metadata in order to check all active connection on cube. Using these API it is easy to list all active connection and disconnect any active connection. Reader can find these details of JAVA APIs in SAS book “SAS 9.1 Java Metadata Interface: User’s Guide”.

5.4 GENERIC UNIX SHELL SCRIPT

UNIX file will be calling above two mentioned file. Structure of Unix scripting will be as below

- ✓ Check all parameters required for this script are properly initialized
- ✓ Read Parameter of Cube from the Parameter file of Cube
- ✓ Ensure all parameters are initialized/defined properly
- ✓ Check is this Cube need to be refreshed on this basis of trigger file and it just independent
- ✓ If trigger file exist or cube is independent of anything then we know that now cube need to be refreshed
- ✓ Take the backup of the cube. Never use “CP –R” command to take the backup of Cube (if it’s on file system) as it may lead to wrong backup. I would recommend to use “MV” command to take backup with some manipulation
- ✓ Check that is there physical data already exist of the Cube because if it exist that it will need to be deleted
- ✓ Build all the parameter need to be passed for SAS program as parameters
- ✓ Then Call Java Program to kill any existing session on the Cube
- ✓ Then call SAS Program to delete physical data and to refresh the Cube.
- ✓ Do not forget to remove trigger file if required.
- ✓ Run any other analysis on refreshed Cube and generate file
- ✓ If required merge all file and Mail status update to all stake holders.

5.6 RECOMMENDATION

Considering the fact that this architecture is manually created, hence we can gather lot of other facts to understand Cube behavior too. These facts can be like

- ✓ Generate a separate file which will maintain the history of the Cube with below information
    - Cube Name refreshed
    - Date of refresh
    - Size of Cube after refresh
    - Time taken by refresh
    - Number of aggregated records were created by Cube
    - Number of NWAY records
- ✓ After Cube refresh Run the code to know the cardinality of the Cube
- ✓ There can be other analysis, which can be done after Cube refresh (Depend on Company/Business requirement)

Above mentioned information will help in analysis of the growth pattern of cube and also to ensure that every month/week cube has been refreshed. This particular file can be send to business user and technical staff after refresh to let them know that Cube has been refreshed and this is the pattern after this month/week Cube refresh.

6. CONCLUSION

Following this architecture Cubes will be refreshed whenever they should and management/business user will always get the report which is based on most recent data. This architecture will remove the possibilities of reporting being done on old data.

7. REFERENCES

1) SAS Institute, Inc. (2006), SAS® Intelligence Platform Administration Guide v9.1.3, Fifth Edition. Cary, NC: SAS Institute Inc.
2) SAS Institute, Inc. (2006), SAS® Intelligence Platform Installation Guide v9.1.3, Fifth Edition. Cary, NC: SAS Institute Inc.
3) SAS Institute, Inc. (2006), SAS® Intelligence platform Overview v9.1.3, Second Edition. Cary, NC: SAS Institute Inc.
4) SAS Institute, Inc. (2006), SAS Institute Inc. 2004. SAS® Web Infrastructure Kit 1.0: Administrator's Guide. Cary, NC: SAS Institute Inc.
5) Work Experience on SAS® 9 Intelligence Platform

8. ACKNOWLEDGMENT

I would like to give many thanks to Monika Singhal for reviewing this document and for making it more useful and readable for the readers.

9. RECOMMENDED READING

http://support.sas.com/documentation/

10. CONTACT INFORMATION

You can contact author at:

Name	:	Gaurav Kumar Agrawal
Address	:	1717 R T Dunn Dr, Suite# 2011 Bloomington, IL 61701
Work Phone	:	7064053450
Fax	:	7062436415
Email	:	gaurav_agrawal@yahoo.com gaurav@gcesolutions.com