

Randomly download data from DB2 database on mainframe to SAS data set on RISC/6000 via SAS

WEN LIN, Shanghai BaoSteel Computer System Engineering Co., Ltd
Shanghai, China

Abstract

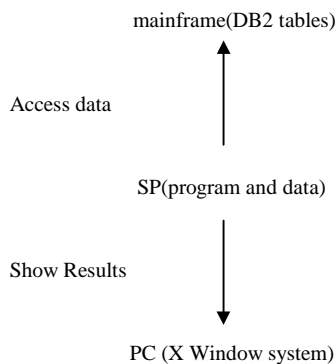
In the Production Management Computer System of BaoSteel, all the production data are stored in DB2 database on mainframe. As a subsystem, the Quality Analyzing System on RISC/6000(SP), which is developed by SAS/AIX, accesses DB2 table and then analyzes these data to gain the correct statistics. So as to develop the application system, the all-important is how to access DB2 table on mainframe from SP reasonably and efficiently.

This paper simply addresses why adopting the method which is randomly download data from DB2 database to SAS data set but not directly access the data on DB2 database, and at the same time emphasizes the process of the achievements with SAS/CONNECT and SAS/ACCESS.

Keywords OLTP, data view, access descriptor

1.0 Introduction of the system architecture

Because the end users to the Quality Analyzing System are decentralized in BaoSteel, in order to ensure the maintenance of the program easily and accordance of the data, both the program and data are decided to be stored on SP. The application system runs on SP, then the results are showed on the clients where X Window system is installed. The system architecture is as followed:



2.0 The key to access DB2 table

There are two keys to access DB2 tables from SP:

1. directly read the SAS view to the DB2 table on mainframe which is created with SAS/ACCESS;
2. download the tables from mainframe to SP, then transfer the data to SAS data sets either regularly or randomly.

Because the application system is not required to be built in an OLTP environment, the real-time character of the data is not necessary. At the same time, as a result of the busyness of the communication with the mainframe and the security mechanism of large database, if the first method is adopted, the end users would often wait for a long time to access the data, thereby the response time of the application system is decreased.

Secondly, the end users wish they would be able to access and analyze the relatively “new” data whenever they want, and reflect the current statistical information utmost. So adopted randomly download DB2 tables to SP.

3.0 Design

3.0.1 Create SAS data view on mainframe

During the process of realization, SAS/CONNECT and SAS/CONNECT are mostly used.

SAS/ACCESS is a software module in SAS system, it provides the ability for the users to access other database files pellucidly, such as DB2 tables.

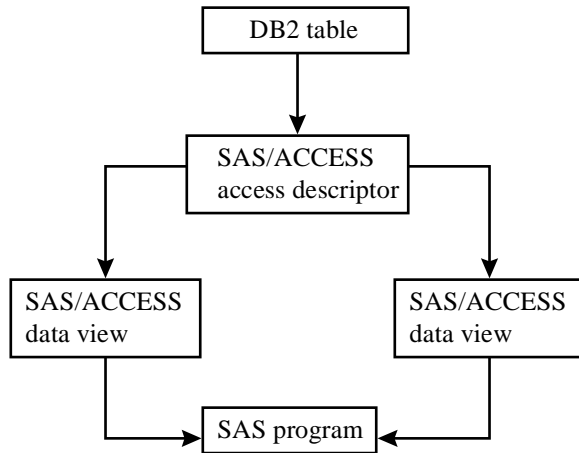
Before download the tables to SP, the SAS data view to the DB2 tables must be created on mainframe with SAS/MVS so that other SAS procedures on SP can access these data.

There are two phases in creating SAS view:

Table 3.1

phase	mission	created result
1	describing DB2 table for SAS	SAS/ACCESS access

		descriptor
2	created one or more SAS view from access descriptor	SAS/VIEW data view



The maximum length of the DB2 table's column name is 18 characters, thus that of the SAS data set's column name is restricted to 8, therefore during the course of the SAS data view's creation, the column name of DB2 table is automatically cut to 8 characters. If some duplicate column names still exist, they must be modified manually. At the same time, the data type of the variables also automatically change to either character type or numerical one according to the DB2 data type. In SAS system, each numerical variable is stored in 8 bytes, the length is indicated as 8, no matter how long the length of the variable is specified in the DB2 database. Following is one of the DB2 table before being transferred:

Table 3.2

Variable name	type	length
HOT_STRIP_NUM	C	10
ORDER_NUM	C	10
MATERIAL_THICK	SI	4
MATERIAL_WIDTH	SI	5
MATERIAL_LENGTH	I	5
SORTING_GRADE	C	1
CUST_ORD_QUALITY	C	16
STEELGRADE	C	8
TEMPER	C	8
COATING_WEIGHT_TOP	SI	3
CTNG_WEIGHT_BOTTOM	SI	3
MATERIAL_WEIGHT	I	5

DETERMINE_RESULT	C	20
SEAL_CAUSE	C	14
SEAL_PLANT	C	10
SEAL_UNIT	C	12
WITHHOLDING_DATE	C	8
SEAL_OPERATOR	C	4
DATE_PROC	C	8
PROCESSOR	C	4
ACCREDIT_PLANT	C	10
ACCREDIT_UNIT	C	12

Table 3.3 the program which is to create SAS data view to DB2 table:

```

submit continue; /*submit*/
%let uid=*****;
%let pwd=*****;
%let rm=190.2.68.2;
OPTIONS REMOTE= rm;
OPTIONS COMAMID= tcp;
FILENAME RLINK "/home2/dev/lzzjrp/connbтчh.scr";
signon;
endsubmit;

submit continue remote; /*remote submit*/
OPTIONS TRANTAB=(,,,,,,CHINESE);
PROC ACCESS DBMS=DB2;
CREATE LZZJRP.DTIC0020.ACCESS;
TABLE=IC.DTIC0020;
ssid=dbpa;
CREATE LZZJRP.DTIC0020.VIEW;
SELECT ALL;
RENAME ACCREDIT_PLANT = AEDT_PLT
ACCREDIT_UNIT = AEDT_UNT
COATING_WEIGHT_TOP = COATING_
CTNG_WEIGHT_BOTTOM = CTNG_WEI
CUST_ORD_QUALITY = CUST_ORD
DATE_PROC = DATE_PRO
DETERMINE_RESULT = DETERMIN
SORTING_GRADE = GRADE
HOT_STRIP_NUM = HOT_STRI
MATERIAL_LENGTH = LENGTH
MATERIAL_WEIGHT = MATERIAL
ORDER_NUM = ORDER_NU
PROCESSOR = PROCESSO
SEAL_CAUSE = SEAL_CAU
SEAL_OPERATOR = SEAL_OPE
SEAL_PLANT = SEAL_PLA
SEAL_UNIT = SEAL_UNI
STEELGRADE = STEELGRA
TEMPER = TEMPER
MATERIAL_THICK = THICK
MATERIAL_WIDTH = WIDTH
WITHHOLDING_DATE = WITHHOLD;
RUN;
endsubmit;
    
```

The process to create SAS data view is realized by logging on to mainframe and submitting program remotely from SP. It can be also realized with SAS/MVS on mainframe.

Table 3.4 the SAS data set after transfer:

Variable name	Type and length
HOT_STRI	\$10
ORDER_NU	\$10
THICK	8
WIDTH	8
LENGTH	8
GRADE	\$1
CUST_ORD	\$16
STEELGRA	\$8
TEMPER	\$8
COATING_	8
CTNG_WEI	8
MATERIAL	8
DETERMIN	\$20
SEAL_CAU	\$14
SEAL_PLA	\$10
SEAL_UNI	\$12
WITHHOLD	\$8
SEAL_OPE	\$4
DATE_PRO	\$8
PROCESSO	\$4
AEDT_PLT	\$10
AEDT_UNT	\$12

“\$” indicates that the variable is character type.

The program is running on SP, but it requests to call the service on mainframe, therefore the program must be submitted to SAS/MVS to run on mainframe, the connection process is realized by SAS/CONNECT.

SAS/CONNECT provides the cooperation environment to SAS application and the communication connection between the SAS procedures which run on different platforms.

With SAS/CONNECT, this program running on SP can build a connection with one or more procedures on mainframe. Based upon the connection, the procedure on SP can access the remote DB2 table, submit program to remote SAS procedure to run and return the results.

In our development environment, this connection is a terminal-based one using TCP/IP.

If there is Chinese in DB2 table, in the SCL “OPTIONS TRANTAB=(,,,,,,CHINESE);” must be added so that the Chinese can be correctly transferred from mainframe to SP, otherwise the Chinese on SP will be in disorder.

3.0.2 Download DB2 table to SAS data set

There are four DB2 tables: table_a, table_b, table_c and table_d should be downloaded. Each table has ten thousands of records. If downloaded all the data every time, without question, it will waste communication resource. Therefore only the data which is newly inserted into DB2 table and has not downloaded yet should be appended to SAS data set. Table_a, table_b, in which some records would be deleted, so that the

common time column “date_pro” in the two tables is selected to act as a mark to compare whether a record has been downloaded or not.

On SP, two data sets are created, one for each DB2 table. The mark’s value is initiated to be stored in it, and every time download data, the record in which the value of “date_pro” is greater than that of the mark is permitted to download.

Among the new download data, the maximum value of “date_pro” is put to replace the value of the mark.

Table 3.5 the SCL which is to download table_a and table_b

```

DT20ID=OPEN('RPTEST.DATE20','U');
RC=FETCH(DT20ID);
n_DT20=varnum(DT20ID,'DATE');
LASTDT20=getvarn(DT20ID,n_DT20);
RC1=CLOSE(DT20ID);
                                SUBMIT

CONTINUE REMOTE;
PROC DOWNLOAD DATA=LZZJRP.DTIC0020 OUT=TEMP20;
WHERE INPUT(DATE_PRO,8.)>&LASTDT20;
RUN;
ENDSUBMIT;

SUBMIT CONTINUE;
PROC APPEND BASE=RPTEST.DTIC0020 DATA=TEMP20
FORCE;
RUN;
ENDSUBMIT;

SUBMIT CONTINUE REMOTE;
PROC SQL NOPRINT;
CREATE TABLE LZZJRP.DATE20 AS SELECT
MAX(INPUT(DATE_PRO,8.)) AS DATE FROM
LZZJRP.DTIC0020;
QUIT;

PROC DOWNLOAD DATA=LZZJRP.DATE20
OUT=RPTEST.DATE20;
RUN;
ENDSUBMIT;
    
```

In table_c, the “rls_d” column where system time is stored is selected to act as the mark. On SP, one data set is created, the mark’s value is initiated to be stored in it. Because the value of the column is created by system automatically, not manually, in order to ensure the downloaded data would not be missed or duplicated, only the records in which the value of “date_pro” is not only greater than that of the mark but also less than the intraday time are permitted to download.

Among the new download data, the maximum value of “rls_d” is put to replace the value of the mark.

Some of the information is manually entered in table_c, so errors are inevitable. After the judgement of which data should be downloaded, then these data are temporarily stored in temporary data sets on SP, if they are checked and proved to be correct, then they can be appended to the normal data sets, otherwise they must be modified by end users

before downloading.

Table 3.6 the SCL which is to download table_c

```

DT22ID=OPEN('RPTEST.DATE22',U);
RC=FETCH(DT22ID);
n_DT22=varnum(DT22ID,'DATE');
LASTDT22=getvarn(DT22ID,n_DT22);
RC1=CLOSE(DT22ID);
SUBMIT CONTINUE REMOTE;
PROC DOWNLOAD DATA=LZZJRP.DTIC0022 OUT=TEMP22;
where input(rls_d,8.)>&lastdt22 and
input(rls_d,YYMMDD8.)<&todaydt;
RUN;
ENDSUBMIT;

submit continue;
data ERROR22;
set TEMP22;
if RLS_RSLT='1' AND DETERMIN=' ' then delete;
if RLS_RSLT='1' AND DETERMIN='a' THEN DELETE ;
if RLS_RSLT='1' and DETERMIN='b' then delete;
if rls_rslt='1' and DETERMIN='c' then delete;
if rls_rslt='1' and DETERMIN='d' then delete;
if rls_rslt='1' and DETERMIN='e' then delete;
if rls_rslt='1' and DETERMIN='f' then delete;
if rls_rslt='1' and DETERMIN='g' then delete;
if rls_rslt='1' and DETERMIN='h' THEN DELETE;
if rls_rslt='1' and DETERMIN='i' THEN DELETE;
if rls_rslt='1' and DETERMIN='j' THEN DELETE;
if rls_rslt='1' and DETERMIN='k' THEN DELETE;
if rls_rslt='1' and DETERMIN='l' THEN DELETE;
if rls_rslt='1' and DETERMIN='m' THEN DELETE;
if rls_rslt in ('2','3','4') then delete;
RUN;
PROC SQL NOPRINT;
CREATE TABLE ERROR AS SELECT COUNT(*) AS
ERRORNUM FROM ERROR22;
QUIT;
ENDSUBMIT;

NUM22ID=OPEN('ERROR',U);
RC=FETCH(NUM22ID);
n_NUM22=varnum(NUM22ID,'ERRORNUM');
ERROR22=getvarn(NUM22ID,n_NUM22);
RC1=CLOSE(NUM22ID);

IF ERROR22<>0 then do;
SUBMIT CONTINUE;
DATA _NULL_;
WINDOW Error
COLUMNS=48 ROWS=16 ICOLUMN=18 IROW=14
#3 @10 'in table_c, rls_rslt does not match determin;'
#5 @10 'please modify table_c before download it;'
#7 @10 'please push ENTER to return';
    DISPLAY ERROR;
    STOP;
    RUN;
    ENDSUBMIT;
    LINK TERM;
    END;
ELSE DO;
    SUBMIT CONTINUE;
    PROC APPEND BASE=RPTEST.DTIC0022 DATA=TEMP22
    FORCE;
    RUN;
    ENDSUBMIT;

    SUBMIT CONTINUE;
    PROC SQL NOPRINT;

```

```

CREATE TABLE rptest.DATE22 AS SELECT
MAX(INPUT(RLS_D,8.)) AS DATE
FROM temp22;
QUIT;
ENDSUBMIT;

END;

```

The records in table_d continuously increase and will not be deleted, so that the number of the records is selected to act as a mark to compare whether a record has been downloaded or not yet. The other steps are the same as that of the three tables.

According to the end users' requirement, the download step of table_a, table_b, and the step of table_c, table_d are separated, likewise, this arrangement is more economic than the method to download all of the four tables at one time.

SQL is widely used in relational database, SAS provides SQL procedure to implement structured query language, the procedure increase the work efficiency of application system in the fields of accessing, creating and modifying tables.

4.0 Conclusion

During the course of the development, the intensive data processing in SAS provides powerful support to achieve speed and efficiency in dealing with the large DB2 tables. Up to now, there are ten thousands of records have been downloaded to SP correctly.

5.0 Reference

SAS BASAL TUTORIAL,

Version 1, first edition(1997), SAS Institute Ltd. Shanghai Representative Office.

Lin, Wen(1998), "Cold-rolled Products Quality Analyzing System development document"

6.0 Author

Wen Lin

Software Engineer

Microsoft Certified Systems Engineer

Shanghai BaoSteel Computer System Engineering Co., Ltd

No. 688, Fu Jin Road, Shanghai, China

Zipcode: 201900

Work

phone:

86-21-56191018

e-mail: linwen@cybergal.com