



VINCI SAS/Grid Environmental FAQs and Enhanced Analytics

Mark Ezzo

VINCI SAS Administrator



December 4, 2014



Poll Question: SAS Usage

ETL (Data extraction) _

Data Analysis _

Statistical Analysis _

Reporting _

Other _

Please select all that apply

VINCI SAS/Grid Agenda

- SAS Grid FAQs
 - Environmental
 - Linux vs. Windows
 - Best Practices
- Advanced Analytics
 - SAS and R interface
 - Enterprise Guide
- Summary

SAS/Grid 9.4 Client

1. Basic SAS Installation

- SAS Base
- Oledb for SQL Server Data
- No Statistical Packages!

2. SAS Grid 9.4

- All SAS Products (Base, Stat, etc.)
- ODBC for SQL Server Data (We set up your data)
- No Oledb on the Linux Grid
- Best accessed the EG 6.1 Configuration

3. Batch Processing

- User can submit and forget (no need to remain connected)
- Allows for SAS checkpoint/restart capability
- Uses SAS Grid Manager metadata for centralized control

Space Considerations

- SAS Clients have limitations of 300 GB of Project Space for Consumption in the Windows Environment.
- SAS Grid has ~43 TBs of Space for Consumption and more is planned for the future (~76 TB).
- This does not mean we sacrifice “good practices” with queries, codes and space!!!

Other Features

- SAS 9.4 Stored Processes introduces stored process reports and the STP procedure. A stored process report is a new object type that contains stored process output that is cached. The output can be viewed without re-executing the stored process. PROC STP enables users to execute a stored process from a SAS program. PROC STP can be executed in an interactive, batch, or server SAS session and can even be executed by another stored process. Essentially, anyone with a Web Viewer can execute and view the results, without using SAS itself.

Stored Process Demonstration

- We will assist.
- Users can create Stored Processes (we will provide the training)
- Please send your suggestions for tools that we can build!

Grid Data Transference

- Lib names and File names
- WinSCP software
- Examples

Linux to Windows via Program

```
data x;infile cards;
input pet $ age;
cards;
Festus 6
Bailey 6
Penny 4
Cody 18
;
run;
*** Exports results;
proc export data    = work.x
outfile = '/smb/vhacdwsasrds01.vha.med.va.gov/ME_Share/pets_name1.csv'
dbms    = csv replace;
putnames = yes;
run;
```

Inefficient Program

- /* Old way, involving lots of reads (8) and writes (6) to the storage disk. */
- PROC SORT DATA=FACT_TABLE ; BY SCRSSN ; RUN ;
- PROC SORT DATA=MASTER.PATIENT_DEMO OUT=MASTER(KEEP=SCRSSN) ; BY SCRSSN ; RUN ;
- DATA COMBINE ;
- MERGE FACT_TABLE(IN=IN1) MASTER(IN=IN2) ;
- BY SCRSSN ;
- IF IN1 ;
- RUN ;
- PROC SORT DATA=COMBINE ; BY STA6A ; RUN ;
- PROC SORT DATA=MASTER.SITE_DEMOGRAPHICS OUT=SITE_INFO(KEEP=STA6A) ; BY STA6A ; RUN ;
- DATA COMBINE;
- MERGE COMBINE(IN=IN1) SITE_INFO(IN=IN2) ;
- BY STA6A ;
- IF IN1 ;
- RUN;

Efficient Programming

- `/* Hash object approach, 3 tables read into memory but only one write action. */`
- `DATA COMBINE(DROP=_:);`
- `/* Initialize the exterior demographic fields to the PDV with pseudo SET statement */`
- `IF 0 THEN SET MASTER.PATIENT_DEMO(KEEP=SCRSSN)`
- `MASTER.SITE_DEMOGRAPHICS(KEEP=STA6A);`
- `IF _N_=1 THEN DO; /* define the hash object(s) once, on the first iteration */`
- `/* this 1st hash object contains info on the patient demographics */`
- `DECLARE HASH PAT_INFO(DATASET:'MASTER.PATIENT_DEMO(KEEP=SCRSSN', HASHEXP:12);`
- `PAT_INFO.DEFINEKEY('SCRSSN'); PAT_INFO.DEFINEDATA(ALL:'Y'); PAT_INFO.DEFINEDONE();`
- `/* this 2nd hash object contains info on the site demographics */`
- `DECLARE HASH SITE_INFO(DATASET:'MASTER.SITE_DEMOGRAPHICS(KEEP=STA6A)');`
- `SITE_INFO.DEFINEKEY('STA6A'); SITE_INFO.DEFINEDATA(ALL:'Y'); SITE_INFO.DEFINEDONE();`
- `END;`
- `SET WORK.FACT_TABLE ; /* this is our cohort file, will read thru all these records */`
- `_RC1=PAT_INFO.FIND(); /* find the matching SCRSSN record from the unique key and pull into PDV */`
- `_RC2=SITE_INFO.FIND(); /* find the matching STA6A record from the unique key and pull into PDV */`
- `RUN;`

SQL Pass-Through via Wizard

```
%_eg_conditional_drops(GRIDWORK.QUERY_FOR_ANTIBIOTIC);
```

```
PROC SQL;
```

```
CONNECT TO SQLSVR as con1
```

```
(READBUFF=5000 INSERTBUFF=3000 DEFER=YES AUTOCOMMIT=NO
```

```
CURSOR_TYPE=FORWARD_ONLY
```

```
UTILCONN_TRANSIENT=YES Datasrc=CDWork_cdwa01 authdomain=DefaultAuth);
```

```
CREATE TABLE GRIDWORK.QUERY_FOR_ANTIBIOTIC AS
```

```
SELECT *
```

```
FROM CONNECTION TO con1 (
```

```
SELECT t1.AntibioticSID,
```

```
    t1.Antibiotic,
```

```
    t1.Sta3n,
```

```
    t1.AntibioticLEN,
```

```
    t1.AntibioticDisplayComment
```

```
FROM Dim.Antibiotic t1
```

```
WHERE t1.Sta3n <= 680);
```

```
DISCONNECT FROM con1;
```

```
QUIT;
```

SAS and R – Example #1

- `proc iml;`
- `/* Comparison of matrix operations in IML and R */`
- `print '----- SAS/IML Results -----';`
- `x = 1:3; /* vector of sequence 1,2,3 */`
- `m = {1 2 3, 4 5 6, 7 8 9}; /* 3 x 3 matrix */`
- `q = m * t(x); /* matrix multiplication */`
- `print q;`
-
- `print '----- R Results -----';`
- `submit / R;`
- `rx <- matrix(1:3, nrow=1) # vector of sequence 1,2,3`
- `rm <- matrix(1:9, nrow=3, byrow=TRUE) # 3 x 3 matrix`
- `rq <- rm %*% t(rx) # matrix multiplication`
- `print(rq)`
- `endsubmit;`

SAS to R

```
proc iml; run ExportDataSetToR("Sashelp.Class",  
"df" ); submit / R; names(df); endsubmit;
```

Produces:

```
[1] "Name" "Sex" "Age" "Height" "Weight"
```

R to SAS

Subroutine	R Source	SAS Destination
ImportDataSetFromR	R expression	SAS data set
ImportMatrixFromR	R expression	SAS/IML matrix

Using R to Analyze Data in a SAS Data Set

```
run ExportDataSetToR("Sashelp.Class", "Class");  
submit / R; Model <- lm(Weight ~ Height,  
data=Class, na.action="na.exclude") endsubmit;
```

The R language is case-sensitive so you must use the correct case to refer to variables in a data frame. You can use the CONTENTS function in the SAS/IML language to obtain the names and capitalization of variables in a SAS data set.

Link to Documentation

<http://support.sas.com/documentation/cdl/en/imlug/65547/PDF/default/imlug.pdf>

SAS IML Documentation with all R facilities has recently been uploaded.

SAS EG 6.1

- Standard for coding and Grid Access
- Enhanced and automatic features
- Easy to Configure and Use
- Wizards for any function
- Able to develop SQL Pass-Through Queries (recommended for the DB)
- Live Demonstration!

SAS Command-Line Grid Submission Utility

- Standalone utility that will allow user to
 - submit SAS program to grid for processing
 - display status of user's jobs on the grid
 - retrieve output from user's jobs to local directory
 - kill jobs

Advantages

- User can submit and forget (Batch Jobs)
 - no need to remain connected to process job
- User can view job output while job is running
- Allows for SAS checkpoint/restart capability
- Uses SAS Grid Manager metadata for centralized control
- NOTE - requires shared file system between client and grid

Submitting a Job

- `sasgsub -grids submitpgm <sas_pgm>`
 - other parameters stored in configuration file
 - `-GRIDWORK <shared_file_dir>`
 - `<metadata_connection_parameters>`
 - `-GRIDAPPSERVER <app_server_name>`
 - `[-GRIDLICENSEFILE <license_file_pathname>]`
 - `[-GRIDFILESIN <file_list>]`
 - `[-GRIDJOBNAME <job_name>]`
 - `[-GRIDJOBPTS <job_options>]`
 - `[-GRIDRESTARTOK]`
 - `[-GRIDSASOPTS <sas_options_for_job>]`
 - `[-GRIDWORKLOAD <workload_values>]`
 - `[-GRIDWORKREM <remote_shared_file_dir>]`

Example Output

Job ID: 6772

Job directory: "/CNT/sasgsub/gridwork/sascnn1/SASGSUB-2009-03-17_14.09.52.847_testPgm"

Job log file: "/CNT/sasgsub/gridwork/sascnn1/SASGSUB-2009-03-17_14.09.52.847_testPgm/testPgm.log"

SAS Gsub Example

- Here is the command we will be using for batch processing on the Grid:
- `sasgsub -GRIDSUBMITPGM`
`/data/prod/admin/VHASLCEZZOM/Programs/Program.sas`
-
- It breaks down as follows:
-
- `sasgsub` is the command
- `-GRIDSUBMITPGM` is the option to submit a program
- `/data/prod/admin/VHASLCEZZOM/Programs/` is the directory where the program is located
- `Program.sas` is the program you wish to run
- Live Demo!

SAS Enterprise Miner

- SAS/EM is the state of the art Data Mining and Modeling Software.
- Friendly User interfaces allows less programming and more analysis.
- Access via IE through this link:
- <http://vhacdwsasmid1.vha.med.va.gov:7980/SASEnterpriseMinerJWS/Status>
- Quick Demo

Good Gridding!

Thank you for attending.

Please contact VINCI SAS Administrators:

Vinci SAS Admimistrators@va.gov

with any questions or comments.