Points to Cover:

Benefits of Grid Computing
Server Environment
Program Components
The ‘actual” code
Review the log(s)

Benefits of Grid Computing:
- higher productivity
- run SAS jobs asynchronously
- Grid Manager does all the ‘hard’ work
- readily schedule jobs ‘off’ hours
Server Environment

The “SAS Server”
- Meta Server sas01
- Grid Manager sas02
- Compute sas03
- Compute sas04
- Compute sas05

5 node server
Microsoft Server 2012 R2
15 TB server storage
768GB total memory (256GB / compute server)
General Parallel File System (GPFS)

EG6.1 & Base SAS (connect)

DB2
SQL Server
External SAS Libs
Program Tasks:
- activate the grid manager
- create ‘remote’ sessions (signon)
- specify selection criteria (%syslput)
- submit jobs
  - connect to data repositories
  - process data based on selection criteria
- wait for jobs to complete
- end remote sessions (signoff)

Parameters to pass to each grid session:
- Input or ‘source’ data location
- Output or ‘target’ data location
- Criteria or ‘data selection’
- ‘main’ program
The Problem

Create a monthly summary of average balance by product class over the last 3 months (need to process millions of records ‘quickly’)

Use the SAS grid manager to asynchronously run 3 remote sessions – each session to extract and process 1 month of data.

What’s needed:
- connect script to access DB2 data warehouse (2 schema & 3 tables)
- libname for permanent SAS dataset (STAGING)
- start and end dates for Jan 2016, Dec 2015 and Nov 2015
- ‘main’ SAS program

DB2 connect script & Staging libname assignment reside in COMMON directory

Use the %SYSLPUT statement to pass date parameters to each remote session

%SYSLPUT Statement
Creates a single macro variable in the server session or copies a specified group of macro variables to the server session.

Form 1:  %SYSLPUT macro-variable=value </REMOTE=server-ID>;
Form 2:  %SYSLPUT _ALL_ | _AUTOMATIC_ | _GLOBAL_ | _LOCAL_ | _USER_  
        </LIKE='character-string'><REMOTE=server-ID>;
Each remote session is independent with a unique WORK directory
*** Simplistic grid enabled program ***;

**** start the grid manager ****;
%let rc = %sysfunc(grdsvc_enable(_all_, resource=SASApp));

**** create the remote sessions ****;
signon sess1 signonwait=yes connectwait=no;
signon sess2 signonwait=yes connectwait=no;
signon sess3 signonwait=yes connectwait=no;
run;

**** specify date criteria ****;
%syslput startdt = '01jan2016'd / remote=sess1;
%syslput enddt = '31jan2016'd / remote=sess1;

%syslput startdt = '01dec2015'd / remote=sess2;
%syslput enddt = '31dec2015'd / remote=sess2;

%syslput startdt = '01nov2015'd / remote=sess3;
%syslput enddt = '30nov2015'd / remote=sess3;
run;
Initial Log for signon and %syslput statements

801  %let rc = %sysfunc(grdsvc_enable(_all_, resource=SASApp));

802  signon sess1 signonwait=yes connectwait=no ;
     NOTE: Remote session ID SESS1 will use the grid service _ALL_.
     NOTE: SIGNON request submitted to grid as job ID '99623'.
     NOTE: SIGNON request completed to grid host 'app-p-sas04.atb.ab.com'.

803  signon sess2 signonwait=yes connectwait=no ;
     NOTE: Remote session ID SESS2 will use the grid service _ALL_.
     NOTE: SIGNON request submitted to grid as job ID '99624'.
     NOTE: SIGNON request completed to grid host 'app-p-sas03.atb.ab.com'.

804  signon sess3 signonwait=yes connectwait=no ;
     NOTE: Remote session ID SESS3 will use the grid service _ALL_.
     NOTE: SIGNON request submitted to grid as job ID '99625'.
     NOTE: SIGNON request completed to grid host 'app-p-sas04.atb.ab.com'.

806  %syslput startdt = '01jan2016'd / remote=sess1;
807   %syslput enddt = '31jan2016'd / remote=sess1;

808  %syslput startdt = '01dec2015'd / remote=sess2;
809   %syslput enddt = '31dec2015'd / remote=sess2;

810  %syslput startdt = '01nov2015'd / remote=sess3;
811   %syslput enddt = '30nov2015'd / remote=sess3;
812   run ;
**“main” program is a macro** **;**

```sas
%macro rollup(job) ;

rsubmit &job ;
filename common "G:\SASData\teams\ADA\00Pgm\common" ;
%inc common(logonihub) ; /* DB2 connect script */
%inc common(stdlibs) ;       /* STAGING libname   */
run ;

data _null_ ;
   call symput('yymm',put(&enddt,yymmn4.)) ;
run ;

proc sql ;
create table account_rollup_&yymm as
   select
      c.prod_grp_desc_lg as product_class,
      max(a.cal_day) as mth_end_dt,
      count(a.cal_day) as days,
      mean(a.bal) as avg_mth_bal
from
   SOC.SOC_ACCOUNT_ENDING_DAY_BAL_PRD_V as a,
   SOC.SOC_ACCOUNT_SCD_V as b,
   DMC.DMC_PRODUCT_DIM_T1_V as c
where
   a.acct_oid = b.acct_oid
and b.prod_fk = c.prod
and b.curr_versn_flg = 'Y'
and a.cal_day between &startdt and &enddt
/* and a.acct_oid = (??????????)    test only    */
   group by 1
; quit ;
```

```sas
proc copy
   in=work
   out=staging
   ;
   select
      account_rollup_&yymm
   ;
run ;
endrsubmit ;
run ;
%mend rollup ;
```
**** submit the jobs ****;
%rollup(sess1)
%rollup(sess2)
%rollup(sess3)

**** wait for all jobs to finish ****;
waitfor _all_ sess1 sess2 sess3;

**** combine outputs & make a report ****;
data final_rpt;
  set
    staging.account_rollup_1511
    staging.account_rollup_1512
    staging.account_rollup_1601
    ;
run;

proc print data=final_rpt;
run;

**** end the remote sessions ****;
signoff sess1;
signoff sess2;
signoff sess3;
NOTE: Remote submit to REMHOST commencing.
249  %rollup(sess1)
NOTE: Background remote submit to SESS1 in progress.
250  %rollup(sess2)
NOTE: Background remote submit to SESS2 in progress.
251  %rollup(sess3)
NOTE: Background remote submit to SESS3 in progress.

252  **** wait for ALL jobs to complete - then proceed ***** ;
253  waitfor _all_ sess1 sess2 sess3 ;
254  run ;

255  **** combine files and print a report **** ;
256  data final_rpt ;
257     set
258        staging.account_rollup_1511
259        staging.account_rollup_1512
260        staging.account_rollup_1601
261    ;
262  run ;

NOTE: There were 4 observations read from the data set STAGING.ACCOUNT_ROLLUP_1511.
NOTE: There were 4 observations read from the data set STAGING.ACCOUNT_ROLLUP_1512.
NOTE: There were 4 observations read from the data set STAGING.ACCOUNT_ROLLUP_1601.
NOTE: The data set WORK.FINAL_RPT has 12 observations and 4 variables.

263  proc sort data=final_rpt ;
264     by product_class mth_end_dt ;

265  proc print data=final_rpt ;
266     title "Avg Balance Report" ;
267     run ;

NOTE: There were 12 observations read from the data set WORK.FINAL_RPT.
### Work library allocation

#### Final Report

<table>
<thead>
<tr>
<th>Obs</th>
<th>product_class</th>
<th>mth_end_dt</th>
<th>avg_mth_bal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HELOC (Loans)</td>
<td>30NOV2015</td>
<td>43420.54</td>
</tr>
<tr>
<td>2</td>
<td>HELOC (Loans)</td>
<td>31DEC2015</td>
<td>43348.21</td>
</tr>
<tr>
<td>3</td>
<td>HELOC (Loans)</td>
<td>31JAN2016</td>
<td>43330.85</td>
</tr>
<tr>
<td>4</td>
<td>Mortgage (Loans)</td>
<td>30NOV2015</td>
<td>47510.65</td>
</tr>
<tr>
<td>5</td>
<td>Mortgage (Loans)</td>
<td>31DEC2015</td>
<td>47139.19</td>
</tr>
<tr>
<td>6</td>
<td>Mortgage (Loans)</td>
<td>31JAN2016</td>
<td>46692.20</td>
</tr>
<tr>
<td>7</td>
<td>Personal</td>
<td>30NOV2015</td>
<td>1135.89</td>
</tr>
<tr>
<td>8</td>
<td>Personal</td>
<td>31DEC2015</td>
<td>1959.72</td>
</tr>
<tr>
<td>9</td>
<td>Personal</td>
<td>31JAN2016</td>
<td>899.34</td>
</tr>
<tr>
<td>10</td>
<td>Registered</td>
<td>30NOV2015</td>
<td>2783.86</td>
</tr>
<tr>
<td>11</td>
<td>Registered</td>
<td>31DEC2015</td>
<td>2783.86</td>
</tr>
<tr>
<td>12</td>
<td>Registered</td>
<td>31JAN2016</td>
<td>2783.86</td>
</tr>
</tbody>
</table>
NOTE: Remote submit to SESS1 commencing.

ROLLUP JOB sess1

1   filename common "G:\SASData\teams\ADA\00Pgm\common" ;

2   %inc common(logonihub) ;
NOTE: Libref DMC was successfully assigned as follows:
   Engine:       DB2
   Physical Name: IHUBPRD1

NOTE: Libref SOC was successfully assigned as follows:
   Engine:       DB2
   Physical Name: IHUBPRD1

165  %inc common(stdlibs) ;
NOTE: Libref STAGING was successfully assigned as follows:
   Engine:       V9
   Physical Name: G:\SASData\teams\ADA\data\staging

171   run ;

172   data _null_ ;
173   call symput('yymm',put(&enddt,yymmn4.)) ;
174   run ;

176   proc sql ;
177   create table account_rollup_&yymm as select c.prod_grp_desc_lg as product_class, max(a.cal_day) as mth_end_dt,
177!    count(a.cal_day) as days, mean(a.bal) as avg_mth_bal
177!    from SOC.SOC_ACCOUNT_ENDING_DAY_BAL_PRD_V as a, SOC.SOC_ACCOUNT_SCD_V as b, DMC.DMC_PRODUCT_DIM_T1_V as c
177!    where a.acct_oid = b.acct_oid and b.prod_fk = c.prod and b.curr_versn_flg = 'Y' and a.cal_day
177!     between &startdt and &enddt and acct_oid in (?)
177!     group by 1 ;

NOTE: Table WORK.ACCOUNT_ROLLUP_1601 created, with 4 rows and 4 columns.

NOTE: PROCEDURE SQL used (Total process time):
   real time          0.09 seconds
   cpu time           0.01 seconds

179   proc copy in=work out=staging ;
180   select account_rollup_&yymm ;

NOTE: Copying WORK.ACCOUNT_ROLLUP_1601 to STAGING.ACCOUNT_ROLLUP_1601 (memtype=DATA).
NOTE: There were 4 observations read from the data set WORK.ACCOUNT_ROLLUP_1601.
Review:
- Input (Source) data
- Output (target)
- criteria
- ‘main’ program

Next steps:
Grid computing can be applied to:
- ‘stable routine’ large scale applications
- repeatable processing (monte-carlo, modelling)
Move ‘grid’ programs to the SAS Scheduling manager