

Using SAS Studio with a Grid Environment

Jason (Jianduan) Liang

SAS certified:

Platform Administrator, Advanced Programmer for SAS 9

Email: Jianduan.liang@gmail.com

Agenda

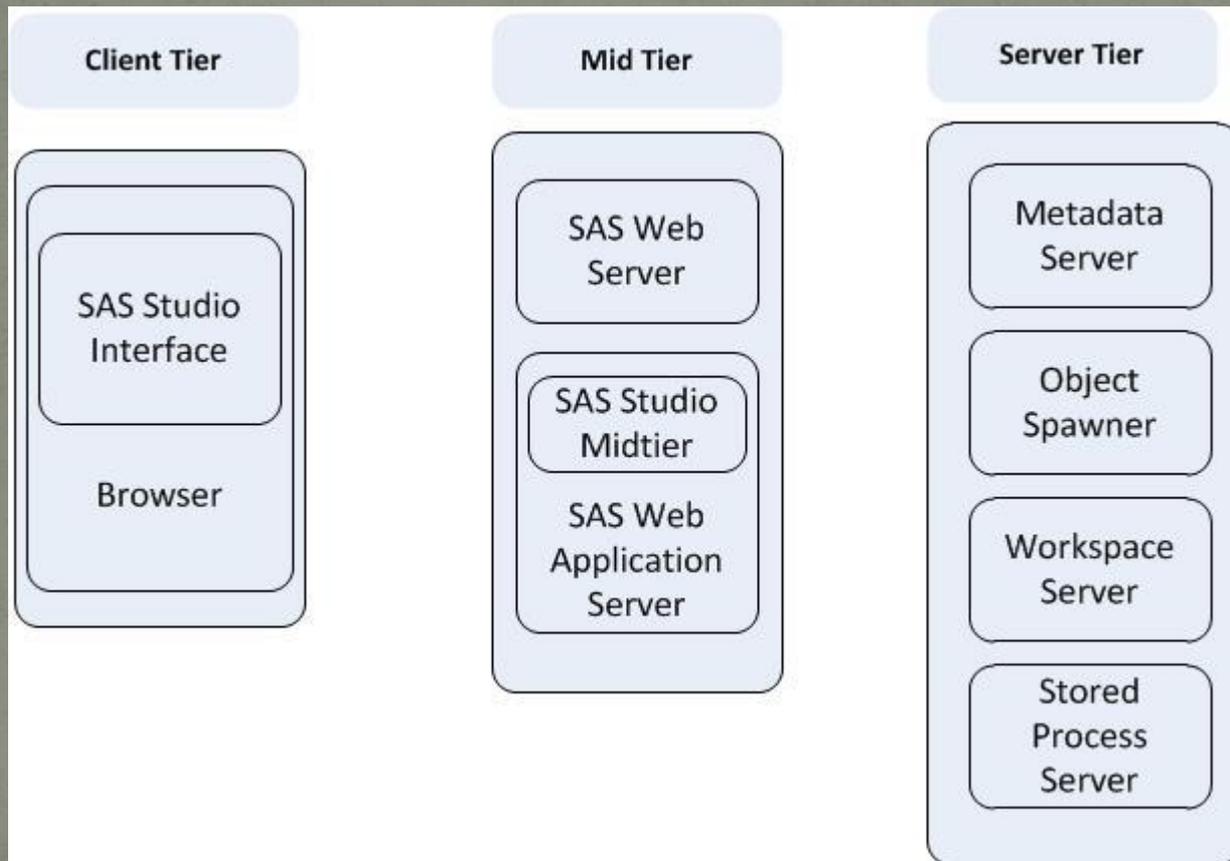
- Introduction of SAS Studio and Grid
- SAS Studio
- SAS Grid and Workload Balancing
- Running SAS jobs in parallel
- Common Library Related Issues
- Job Monitoring

Introduction

- SAS Studio
 - ❖ A multifunctional application
 - ❖ Write and run SAS code through web browser
 - ❖ Predefined tasks to generate SAS code
 - ❖ Code is submitted to a SAS server to execute
 - ❖ The SAS server can be on your local machine, or a remote server, a server hosted in a cloud environment, a grid environment
- SAS Grid
 - ❖ SAS computing tasks are distributed among multiple computers on a network, all controlled by SAS Grid Manager
 - ❖ Scheduling jobs
 - ❖ Scalable

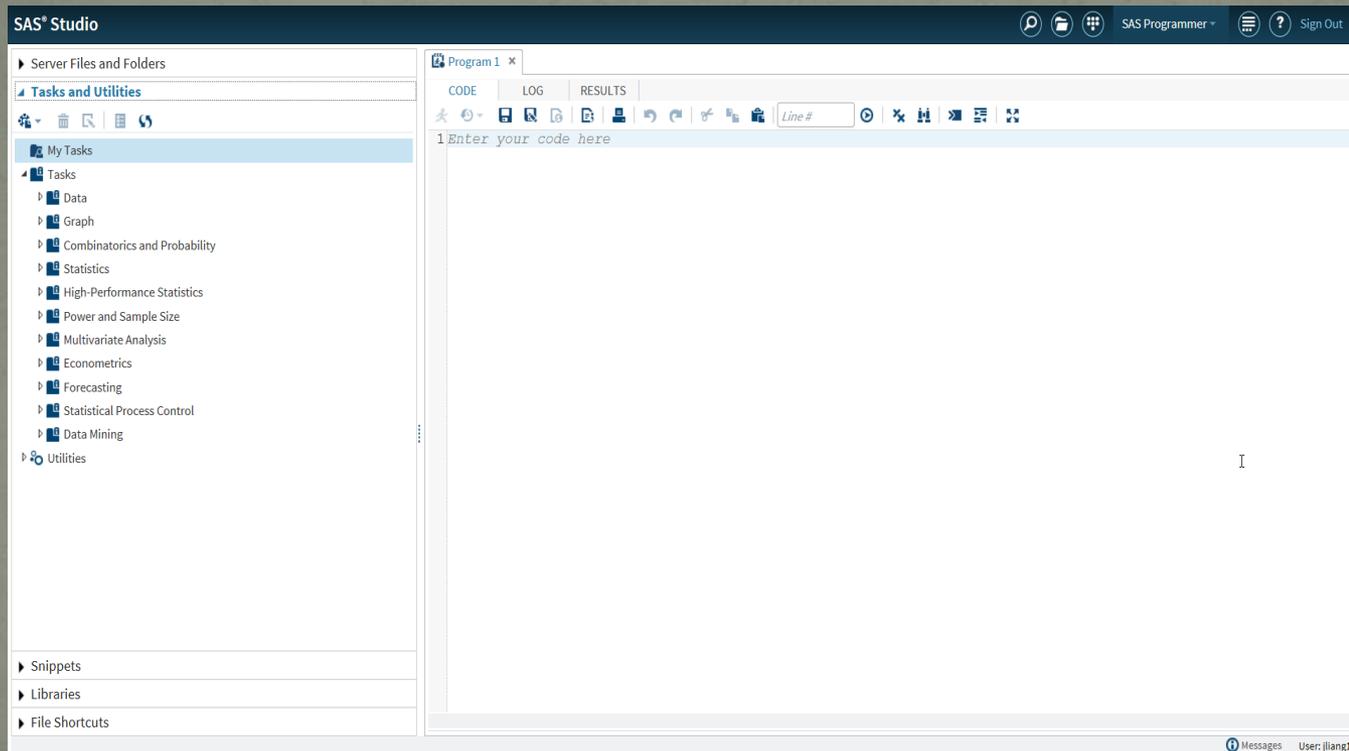
SAS Studio

Conceptual high level architecture:

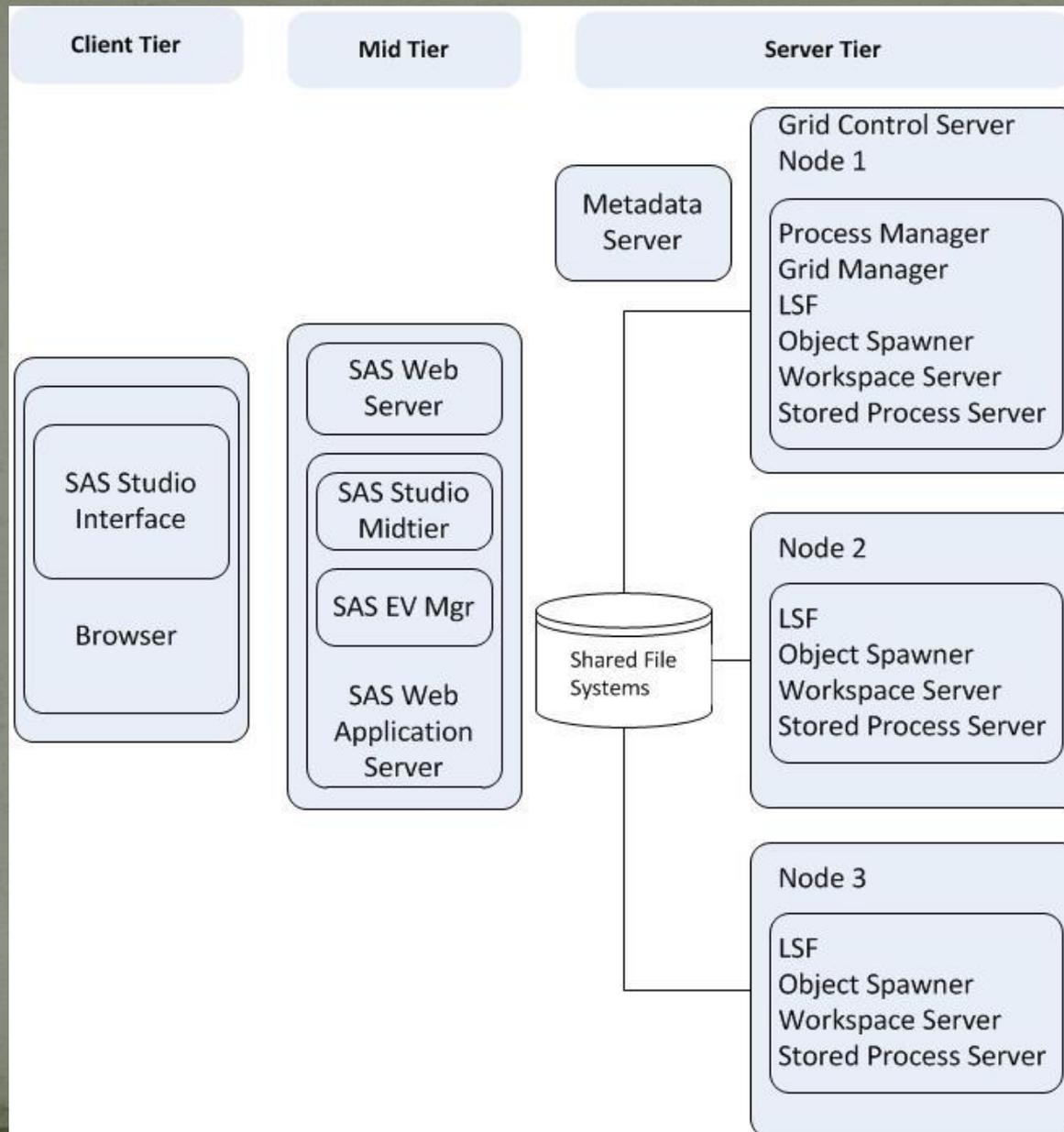


SAS Studio

- Interface: Navigation Pane, Work Area
- Connected to a work space server to run code



SAS Grid



Grid Workload Balancing

- Multi-user workload balancing
 - A job is dispatched to the best available grid node
 - If no resources available in the grid, jobs are held in a queue until resources are available
 - Job priority can be set to determine whether a job is run immediately or held in a queue
- Parallel workload balancing
 - Some SAS programs consist of subtasks that are independent units of work and can be distributed across a grid and executed in parallel

Grid Workload Balancing

- Grid lunched work space server

The screenshot shows a dialog box titled "SASApp - Logical Workspace Server Properties" with a close button in the top right corner. The "Load Balancing" tab is selected, showing the following configuration:

- Balancing algorithm:** Grid (dropdown menu)
- Availability timeout (sec):** 60 (text input)
- Logical server credentials:** (None) (dropdown menu)
- Algorithm properties** (grouped section):
 - Cost per client:** 100 (text input)
 - Launch servers via Grid** (checkbox)
 - Grid server:** SASApp - Grid Server (dropdown menu)
 - Grid server credentials:** (None) (dropdown menu)
 - Grid server connect timeout:** 0 (text input)

At the bottom right, there are three buttons: "OK", "Cancel", and "Help".

Grid Workload Balancing

- Each SAS Studio has two work space server sessions
- One is used to run the user code, while the other is used for file I/O and other internal operations

SAS® Grid Manager 🔔 ? 👤 Jason Liang

Home Monitoring **Jobs**

Jobs ↻

Search (Select Actions) ▾

ID	Name	Status	Queue	User	From Host	Exec Host	Submit Time	Start Time	End Time	Command
1414	Web Infra Pl...	RUNNING	normal	jliang1	sasp-grid2.h...	sasp-grid3.h...	Oct 2, 2016, 11:24:...	Oct 2, 2016, 11:24:...	-	/sasshare/sa...
1413	Web Infra Pl...	RUNNING	normal	jliang1	sasp-grid2.h...	sasp-grid3.h...	Oct 2, 2016, 11:24:...	Oct 2, 2016, 11:24:...	-	/sasshare/sa...
1412	Web Infra Pl...	RUNNING	normal	jliang1	sasp-grid2.h...	sasp-grid1.h...	Oct 2, 2016, 11:23:...	Oct 2, 2016, 11:23:...	-	/sasshare/sa...
1411	Web Infra Pl...	RUNNING	normal	jliang1	sasp-grid2.h...	sasp-grid1.h...	Oct 2, 2016, 11:22:...	Oct 2, 2016, 11:22:...	-	/sasshare/sa...
1410	Web Infra Pl...	RUNNING	normal	jliang1	sasp-grid2.h...	sasp-grid1.h...	Oct 2, 2016, 11:19:...	Oct 2, 2016, 11:19:...	-	/sasshare/sa...
1409	Web Infra Pl...	RUNNING	normal	jliang1	sasp-grid2.h...	sasp-grid1.h...	Oct 2, 2016, 11:19:...	Oct 2, 2016, 11:19:...	-	/sasshare/sa...
1408	SAS Enterpri...	RUNNING	normal	jliang1	sasp-grid2.h...	sasp-grid2.h...	Oct 2, 2016, 11:17:...	Oct 2, 2016, 11:17:...	-	/sasshare/sa...
1407	SAS Enterpri...	DONE	normal	jliang1	sasp-grid2.h...	sasp-grid4.h...	Oct 2, 2016, 8:53:1...	Oct 2, 2016, 8:53:1...	Oct 2, 2016, 11:1...	/sasshare/sa...
1406	Web Infra Pl...	DONE	normal	jliang1	sasp-grid2.h...	sasp-grid4.h...	Oct 2, 2016, 8:04:2...	Oct 2, 2016, 8:04:2...	Oct 2, 2016, 10:5...	/sasshare/sa...
1405	Web Infra Pl...	DONE	normal	jliang1	sasp-grid2.h...	sasp-grid4.h...	Oct 2, 2016, 8:04:1...	Oct 2, 2016, 8:04:1...	Oct 2, 2016, 10:5...	/sasshare/sa...

Parallel Load Balancing

SAS® Studio

Server Files and Folders

- sasp-mdr.health.alberta.ca
 - Folder Shortcuts
 - Files (/sasshare)
- Tasks and Utilities
- Snippets
- Libraries
- File Shortcuts

Program 1 x *program1.sas x

CODE LOG RESULTS

```
9 options fullstimer source source2 msglevel=i notes;
10
11 %let mylib=work;
12 data &mylib..h1(drop=i);
13     do i=1 to 500000000;
14         id=i;
15         height=150+int(ranuni(6)*40);
16         output;
17     end;run;
18 proc sort data=&mylib..h1;
19     by id;run;
20
21 data &mylib..w1(drop=i);
22     do i=1 to 500000000;
23         id=i;
24         weight=45+int(ranuni(100)*105);
25         output;
26     end;
27 run;
28 proc sort data=&mylib..w1;
29     by id;
30 run;
31
32 proc sql;
33     create table &mylib..hw1 as
34     select h.*,w.weight
35     from &mylib..h1 h
36     inner join &mylib..w1 w
37     on h.id=w.id
38     where h.height<152
39     and w.weight>147;
40 quit;
41
```

Line 27, Column 1

Messages User: jliang1

Parallel Load Balancing

```
NOTE: The data set WORK.H1 has 500000000 observations and 2 variables.  
NOTE: DATA statement used (Total process time):  
      real time          1:25.19
```

```
NOTE: There were 500000000 observations read from the data set WORK.H1.  
NOTE: SAS threaded sort was used.  
NOTE: The data set WORK.H1 has 500000000 observations and 2 variables.  
NOTE: PROCEDURE SORT used (Total process time):  
      real time          4:14.08
```

```
NOTE: The data set WORK.W1 has 500000000 observations and 2 variables.  
NOTE: DATA statement used (Total process time):  
      real time          1:25.03
```

```
NOTE: There were 500000000 observations read from the data set WORK.W1.  
NOTE: SAS threaded sort was used.  
NOTE: The data set WORK.W1 has 500000000 observations and 2 variables.  
NOTE: PROCEDURE SORT used (Total process time):  
      real time          4:11.64
```

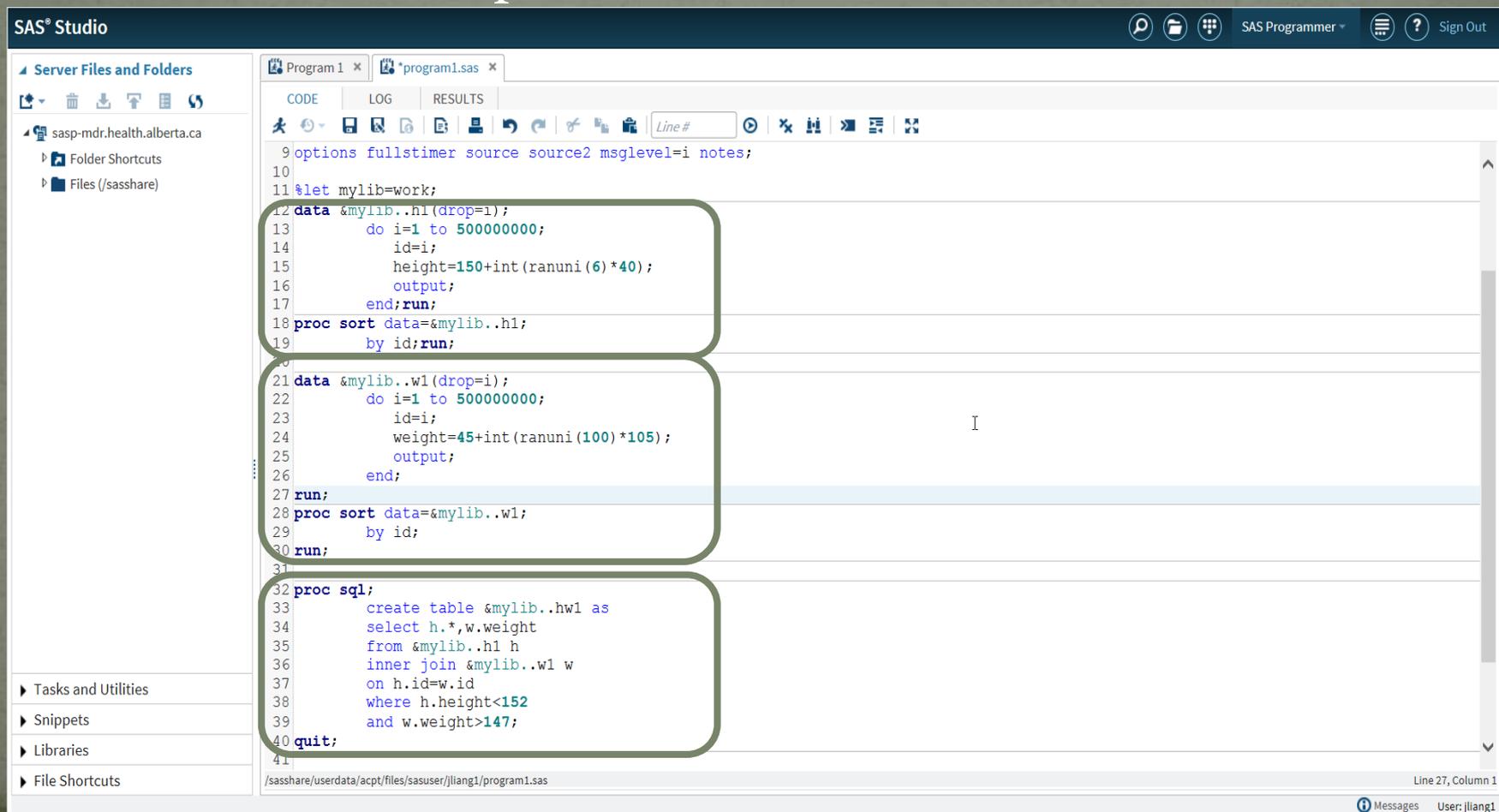
```
Total real time : 11:14.78
```

```
NOTE: Table WORK.HW1 created, with 380248 rows and 3 columns.
```

```
97          quit;  
NOTE: PROCEDURE SQL used (Total process time):  
      real time          1:10.50
```

Parallel Load Balancing

- Divide the SAS program into different tasks, submit each task to a separate SAS session:



The screenshot displays the SAS Studio interface. The main window shows a SAS program with the following code:

```
9 options fullstimer source source2 msglevel=i notes;
10
11 %let mylib=work;
12 data &mylib..h1(drop=i);
13     do i=1 to 500000000;
14         id=i;
15         height=150+int(ranuni(6)*40);
16         output;
17     end;run;
18 proc sort data=&mylib..h1;
19     by id;run;
20
21 data &mylib..w1(drop=i);
22     do i=1 to 500000000;
23         id=i;
24         weight=45+int(ranuni(100)*105);
25         output;
26     end;
27 run;
28 proc sort data=&mylib..w1;
29     by id;
30 run;
31
32 proc sql;
33     create table &mylib..hw1 as
34     select h.*,w.weight
35     from &mylib..h1 h
36     inner join &mylib..w1 w
37     on h.id=w.id
38     where h.height<152
39     and w.weight>147;
40 quit;
```

Three green rounded rectangles highlight the following tasks in the code:

- Task 1: Lines 12-19, creating and sorting data set `h1`.
- Task 2: Lines 21-30, creating and sorting data set `w1`.
- Task 3: Lines 32-40, executing a SQL query to create a table `hw1` based on the previous two data sets.

The SAS Studio interface includes a left-hand pane for "Server Files and Folders" showing the path `sasp-mdr.health.alberta.ca` and a bottom status bar indicating the current file path and user information.

Parallel Load Balancing

- Create SAS Sessions

```
1 /*-----*/
2 /* This is the user-modifiable number of connect sessions */
3 /*-----*/
4 %let SCAPROC_SESSIONS_COUNT=2;
5
6 /*-----*/
7 /* Enable grid service */
8 /*-----*/
9 %let rc=%sysfunc(grdsvc_enable(_all_, resource=SASApp));
10
11 /*-----*/
12 /* This macro starts up the connect sessions */
13 /*-----*/
14 %macro scaproc_start_sessions(count);
15     %do i = 1 %to &count;
16         signon sess&i signonwait=no connectwait=no cmacvar=scaproc_signon_&i;
17     %end;
18 %mend scaproc_start_sessions;
19 /*-----*/
20 /* Start up our connect sessions. */
21 /*-----*/
22 %scaproc_start_sessions(&SCAPROC_SESSIONS_COUNT);
23
24 /*-----*/
25 /* This function call initializes data structures for our SCAGRID functions. */
26 /* We pass in the number of sessions and the number tasks in this job. */
27 /*-----*/
28 proc scaproc; startup 3 &SCAPROC_SESSIONS_COUNT; run;
29
```

Parallel Load Balancing

- Submit task 1 to an available session

```
30 /*-----*/
31 /* Get an available session */
32 /*-----*/
33 proc scaproc; getsession 1 "sess"; run;
34 %put sess=&sess;
35 /*-----*/
36 /* rsubmit for task 1 */
37 /*-----*/
38 rsubmit &sess sysrputsync=yes cmacvar=scagrid_task_1;
39 /* task 1 rsubmit */
40 options fullstimer source source2 msglevel=i notes;
41 libname j1 Base '/sasshare/saswork';
42 %let mylib=j1;
43 data &mylib..h1(drop=i);
44     do i=1 to 500000000;
45         id=i;
46         height=150+int(ranuni(6)*40);
47         output;
48     end;
49 run;
50
51 proc sort data=&mylib..h1;
52     by id;
53 run;
54
55 %sysrput MYLIB=&MYLIB;
56 endrsubmit;
```

Parallel Load Balancing

- Submit task 2 to another session

```
59 /*-----*/
60 /* Get an available session */
61 /*-----*/
62 proc scaproc; getsession 2 "sess"; run;
63 %put sess=&sess;
64 /*-----*/
65 /* rsubmit for task 2 */
66 /*-----*/
67 rsubmit &sess sysrputsync=yes cmacvar=scagrid_task_2;
68 /* task 2 rsubmit */
69 options fullstimer source source2 msglevel=i notes;
70 libname j1 Base '/sasshare/saswork';
71 %let mylib=j1;
72
73 data &mylib..w1(drop=i);
74     do i=1 to 500000000;
75         id=i;
76         weight=45+int(ranuni(100)*105);
77         output;
78     end;
79 run;
80
81 proc sort data=&mylib..w1;
82     by id;
83 run;
84 %sysrput MYLIB=&MYLIB;
85 /* for incomplete steps */ ;run; quit; run;
86 endrsubmit;
```

Parallel Load Balancing

- Wait for task 1 and 2 to complete

```
89 /*-----*/
90 /* Sync with task 1                                */
91 /*-----*/
92 proc scaproc; taskwait 1; run;
93 /*-----*/
94 /*-----*/
95 /* Sync with task 2                                */
96 /*-----I-----*/
97 proc scaproc; taskwait 2; run;
98
```

Parallel Load Balancing

- Submit task 3 to an available session

```
99 /*-----*/
100 /* Get an available session */
101 /*-----*/
102 proc scaproc; getsession 3 "sess"; run;
103 %put sess=&sess;
104 /*-----*/
105 /* syslput for MYLIB */
106 /*-----*/
107 %syslput MYLIB=&MYLIB /remote=&sess;
108 /*-----*/
109
110 /*-----*/
111 /* rsubmit for task 3 */
112 /*-----*/
113 rsubmit &sess sysrputsync=yes cmacvar=scagrid_task_3;
114 /* task 3 rsubmit */
115 proc sql;
116     create table &mylib..hw1 as
117     select h.*,w.weight
118     from &mylib..h1 h
119     inner join &mylib..w1 w
120     on h.id=w.id
121     where h.height<152
122     and w.weight>147;
123 quit;
124 endrsubmit;
```

Parallel Load Balancing

- Signoff each session

```
135 /*-----*/
136 /* This macro issues waitfors and signoffs for our sessions. */
137 /*-----*/
138 %macro scagrid_waitfors(count);
139     %do i = 1 %to &count;
140         waitfor sess&i;
141         signoff sess&i;
142     %end;
143 %mend scagrid_waitfors;
144
145 /*-----*/
146 /* Wait for and sign off all sessions. */
147 /*-----*/
148 %scagrid_waitfors(&SCAPROC_SESSIONS_COUNT);
149
150 /*-----*/
151 /* Termination for our SCAGRID functions. */
152 /*-----*/
153 proc scaproc; shutdown; run;
154
155 /*-----*/
156 /* All done. */
157 /*-----*/
```

Parallel Load Balancing

- Performance - without parallel load balance

```
NOTE: The data set WORK.H1 has 500000000 observations and 2 variables.  
NOTE: DATA statement used (Total process time):  
      real time          1:25.19
```

```
NOTE: There were 500000000 observations read from the data set WORK.H1.  
NOTE: SAS threaded sort was used.  
NOTE: The data set WORK.H1 has 500000000 observations and 2 variables.  
NOTE: PROCEDURE SORT used (Total process time):  
      real time          4:14.08
```

```
NOTE: The data set WORK.W1 has 500000000 observations and 2 variables.  
NOTE: DATA statement used (Total process time):  
      real time          1:25.03
```

```
NOTE: There were 500000000 observations read from the data set WORK.W1.  
NOTE: SAS threaded sort was used.  
NOTE: The data set WORK.W1 has 500000000 observations and 2 variables.  
NOTE: PROCEDURE SORT used (Total process time):  
      real time          4:11.64
```

```
Total real time : 11:14.78
```

```
NOTE: Table WORK.HW1 created, with 380248 rows and 3 columns.
```

```
97          quit;  
NOTE: PROCEDURE SQL used (Total process time):  
      real time          1:10.50
```

Parallel Load Balancing

- Performance - with parallel load balance

```
NOTE: Remote submit to SESS1 commencing.  
NOTE: The data set JL.H1 has 500000000 observations and 2 variables.  
NOTE: DATA statement used (Total process time):  
      real time          1:05.72
```

```
NOTE: There were 500000000 observations read from the data set JL.H1.  
NOTE: SAS threaded sort was used.  
NOTE: The data set JL.H1 has 500000000 observations and 2 variables.  
NOTE: PROCEDURE SORT used (Total process time):  
      real time          2:31.78
```

```
NOTE: Remote submit to SESS2 commencing.  
NOTE: The data set JL.W1 has 500000000 observations and 2 variables.  
NOTE: DATA statement used (Total process time):  
      real time          1:22.17
```

```
NOTE: There were 500000000 observations read from the data set JL.W1.  
NOTE: SAS threaded sort was used.  
NOTE: The data set JL.W1 has 500000000 observations and 2 variables.  
NOTE: PROCEDURE SORT used (Total process time):  
      real time          4:00.01
```

Total real time : 5:22.18

```
NOTE: Remote submit to SESS1 commencing.  
NOTE: Table JL.HW1 created, with 380248 rows and 3 columns.  
  
NOTE: PROCEDURE SQL used (Total process time):  
      real time          47.93 seconds
```

Common Library Related Issues

- Work library only works for the current SAS session
- Permanent library is required for parallel workload balancing
- Code analyzer won't work if using work library
- File doesn't exist due to non-shared file systems
- Data file locked in a shared file systems when running multiple sessions

Summary

- SAS Studio is a multifunctional user friendly tool
- With a grid environment, SAS studio performance can be improved significantly with two levels of workload balancing
- SAS jobs are monitored by using the Grid Manager interface