

SCSUG-2017

SAS® Grid Job Search Performance

Piyush Singh, Ghiyasuddin Mohammed Faraz Khan, Prasoon Sangwan
TATA Consultancy Services Ltd.

ABSTRACT

Have you ever tried to find the job execution information in SAS® Grid? Though, you can use SASGSUB to check the job status but if you really need detailed information then you can use LSF command “bacct”. It provides the detailed information about total job execution but it is quiet slow and usually takes around 15 sec or more to search for jobs executed by a single user. If Grid platform has more than 100 users, which is quiet common in any medium to big sized organization, then bacct may take more than 25 minutes (100*15 sec) to list jobs for all users. If there are around 500 to 1000 users, then bacct may take hours to bring the list of executed jobs.

This paper explains the method to extract the job execution reports (or job search performance) for 1000s of users in just few minutes. It also explains how we can automate the entire process and we can create different kind of reports in a few minutes.

INTRODUCTION

SAS® Grid Job Execution Information can be retrieved by using the LSF command “bacct”. This command provides the standard statistics of all the jobs executed by a single user in around 10-15 seconds. However, generating a detailed report of Job Statistics of all the users in an environment becomes a tedious task. This type of reports are generally used for business analysis and planning. This paper demonstrates a way to extract the grid usage report in a time efficient manner. .

WHAT IS “BACCT” COMMAND AND HOW DOES IT WORK?

By default, LSF stores the job execution information in /LSB_SHAREDIR/cluster_name/logdir/lsb.acct. It stores the information for each job executed on Grid. This file contains one record per job, per user with the entire job accounting information. LSB_SHAREDIR is defined in lsf.conf and cluster name is the name of the LSF cluster, as returned by lsid.

A lot goes on when LSF BACCT command executes. It reads the lsb.acct file along with other lsb.events* files and depending on the command options collects them into readable format with adjusted date and time information, etc. If you go through this process even for one user at a time, for a very active user base and job submission rate, it takes 10-15sec (again based on user base and activity) to format and extract a month’s account data. Consider the situation where you have thousands of active users with SAS Grid. It would take an average of 5 hours to complete. Not only this, if you run bacct for all user in such type of environment

```
-bash-4.1$ bacct -u <user>
```

Accounting information about jobs that are:

- submitted by users student,
- accounted on all projects.
- completed normally or exited
- executed on all hosts.
- submitted to all queues.
- accounted on all service classes.

```

-----
---
SUMMARY:      ( time unit: second )
Total number of done jobs:      190      Total number of exited jobs:      74
Total CPU time consumed:      12473.5    Average CPU time consumed:      47.2
Maximum CPU time of a job:      4279.0    Minimum CPU time of a job:      0.0
Total wait time in queues:      17920.0
Average wait time in queue:      67.9
Maximum wait time in queue:      17177.0    Minimum wait time in queue:      0.0
Average turnaround time:      6904 (seconds/job)
Maximum turnaround time:      743210    Minimum turnaround time:      0
Average hog factor of a job:      0.19 (cpu time / turnaround time )
Maximum hog factor of a job:      1.00    Minimum hog factor of a job:      0.00
Total Run time consumed:      1801496    Average Run time consumed:      6823
Maximum Run time of a job:      743209    Minimum Run time of a job:      0
Total throughput:      0.02 (jobs/hour) during 13193.62 hours
Beginning time:      Mar 8 14:55      Ending time:      Sep 9 09:32

-bash-4.1$

```

Output 1. Output from a BACCT LSF Command

LSB.ACCT FILE STRUCTURE

LSF stores the statistics of every job in a separate line. The statistics are stored in this file in a predefined format separated by blank spaces. Empty Quotations (" ") are placed in the fields when LSF is unable to store any job field information. Each field in this file represents different attribute of job execution. Below is the sample format of lsb.acct file:

```

"JOB_FINISH" "8.0.1" 1494959689 102 40615 33816579 1 1494959678 0 0
1494959682 "<UserId>" "normal" "<ApplicationContextServer>" "" "" "<Server
Name>" "<sasconfig>/Levl/SASApp94" "" "" "" "1494959678.102" 0 1 "<Server
Name>" 64 60.0 "SASGrid:4517"
"<sasconfig>/Levl/SASApplicationServer/GridServer/sasgrid
SASDAEMONHOST:<ServerName> SASDAEMONPORT:42887 SASCLIENTPORT:1" 0.157975
0.216967 24016 0 -1 0 0 41097 0 0 0 136 -1 0 0 0 1260 88 -1 "" "default" 0
1 "" "" 0 3552 50236 "" "" "" "" 0 "" 0 "" -1 "" "" "" "" -1 "" "" 16 ""
1234567890 "" "" 0 0

```

Output 2. Sample lsb.acct File

For detailed information about BACCT and each field of a record in lsb.acct file, please refer http://support.sas.com/rnd/scalability/platform/PSS9.1/lsf9.1.3_config_ref.pdf documentation or contacting the SAS Technical Support.

USE LSB.ACCT TO RETRIVE THE USER'S JOB

As mentioned above, `bacct` command can be used to find job execution information in SAS Grid but this is a time consuming process. It can take hours to get the reports for all of your users.

The job records in `lsb.acct` file contains many date/time stamp fields that each are in "Posix" format. These date/time fields are 10 digit numbers representing the number of seconds since the UNIX "epoch". Job search processing with `bacct` will be a lot faster if you can specify your start, end time range values in this format, and compare directly with `lsb.acct` file. This provides you following flexibilities:

- You can ignore the jobs from this file if you do not want to count like unfinished jobs.
- You can select the jobs (file record) if you want to see only ZERO exit code jobs.
- `lsb.acct` file can be used as data source to create different reports based on your business need.

Generally, it is advisable to be very careful when you are dealing with `lsb.acct` file. This contains lot of Grid information and is being used by many LSF commands. To avoid any conflict, you can first copy the file and then do your processing

Each field in this file represent a different metric and each line consists of all the metrics of a separate job. Below is a sample script, which is used to extract the below metrics using the `lsb.acct` file. This script is supposed to calculate following metrics:

1. The total number of jobs executed by users
2. The CPU time used by different users

```
#!/bin/bash

#Section 1
lsbacct_file=<LSB_SHAREDIR>/cluster_name/logdir/lsb.acct
workloc=/top/usr/metrics
userlist=/top/usr/userlist.txt
starttime=1234567000
endtime=1234569000

#Section 2
cp -p $lsbacct_file $workloc/lsb.acct_bck
chmod 777 $workloc/lsb.acct_bck
awk '$3> '$starttime' && $3<=' $endtime' ' $workloc/lsb.acct_bck >
$workloc/lsf_extract.txt

#Section3
for var in `cat $userlist`;do
finaljobs=`awk '$12 ~ /\<'$var'\>/' $workloc/lsf_extract.txt |wc -l`
echo -e "$var \t $finaljobs" >> $workloc/total_jobs.xls
done

#Section 4
cat $workloc/lsf_extract.txt | awk '{print $12}'| tr -d '"' > $workloc/uids
cat $workloc/lsf_extract.txt |awk '{n = 51; for (--n; n >= 0; n--){ printf
"%s\t",$(NF-n)} print ""}' | awk '{print $1, "\t" $2}' > $workloc/cputimes
pr -m -t -s\ $workloc/uids $workloc/cputimes | gawk '{print $1,"\t" $2,
"\t" $3}' > $workloc/Final.csv

#Section 5
```

```

cat $workloc/Final.csv | awk '{print $2+$3}' > $workloc/timecal
echo -e "userid \t time1 \t time2 \t cpu_tm" > $workloc/cpu_tm.xls
pr -m -t -s\ $workloc/Final.csv $workloc/timecal | gawk '{print $1,"\t"
$2, "\t" $3, "\t" $4}' >> $workloc/cpu_tm.xls

#Section 6
rm $workloc/uids $workloc/cputimes $workloc/Final.csv $workloc/timecal
$workloc/lsb.acct_bck

```

Sample Code 1. Script to Read lsb.acct File

The above script can be modified to retrieve various metrics based on business need. The three main sections of the script are explained as below:

1. This section assigns the required variable used in the script. You can changed these variables as per your environment.
 - “lsbacct_file” is the full path to lsb.acct file.
 - “workloc’ is a temporary working directory for script execution.
 - “Userlist” is the list of users on SAS Grid.
 - “starttime” and “endtime” are the upper and lower limits of time period for the script query.

Please note that LSF stores the time statistics in POSIX/EPOCH time format. Hence, the time variable “starttime” and “endtime” need to be entered in Epoch time format.

2. This section takes a backup of the original lsb.acct file in the temporary working directory. This helps in preventing the original files from any corruption/ damage. LSB.ACCT file contains all the job records from the initial SAS Grid installation.
3. This section consolidates all the job records, calculates the total number of jobs executed by users in a new file, which can be used to create business reports. The output generate from this section of the script would be in the below format.

userid	finaljobs
usr1	0
usr2	0
usr3	166
usr4	22
usr5	0
usr6	0

Table 1: Sample output file created from Section 3 of script

4. This section creates a subset from the original lsb.acct file which contains the jobs executed in given time duration with start and end time. Reading the required fields from job records in lsb.acct file is little difficult since jobs can be invoked on SAS Grid either by using SAS Enterprise Guide or by SASGSUB and the number of fields in a single job records depend on the client used to submit the job. This is not fixed format file so while reading you need to first read the clients used to submit the job and then read the fields for the job execution.

The section generates an excel file which contains the total CPU time used by all the per user creates the temporary files. You can write your SAS code to read the excel file and create your own report. Next topic in this paper, explains how you can create different report.

userid	time1	time2	cpu_tm
usr1	6.324038	3.114023	9.438061
usr2	7.524111	1.624001	9.148112
usr3	0.41301	0.29302	0.70603
usr4	40.324038	6.674098	46.998136
usr5	120.415027	12.555072	132.970099
usr6	152.223137	21.053156	173.276293

Table 2: Sample Output from Section 5

5. This section deletes the temporary files created during the intermediate execution. This clean the unnecessary files once execution completed.

REPORT FOR USER'S GRID UTILIZATION

As discussed earlier, you can create different reports once you have all the user data from lsb.acct file. If you want to create report for Grid utilization per group (defined in SAS Management Console). List of the users and corresponding group can be found from SAS Management Console. Then you can merge the data from lsb.acct and SAS management console by users/ group to group the utilization data. Above given script create the job information in excel file. You can read this file in SAS to work with the user and Group list from SAS Management Console.

```
proc import datafile="/home/usr/cpu_tm.xlsx"
out=cpu_tm dbms=xlsx
replace;
getnames=yes;
run;

proc sort data= cpu_tm;
by userid;
run;

proc sql;
create table users_cput as
select userid as UserId, sum(cputime2) as Total_cputime from cpu_tm
group by userid;
run;

proc export data=users_cputime outfile='/home/usr/user_cpu.xlsx'
dbms=xlsx replace;
run;
```

Sample Code 2. SAS Code to Create Report

Below is the flow chart which demonstrate, how lsb.acct file data can be merged with SAS Management Console user list. This helps you create the report for Grid utilization based on different Groups or user.

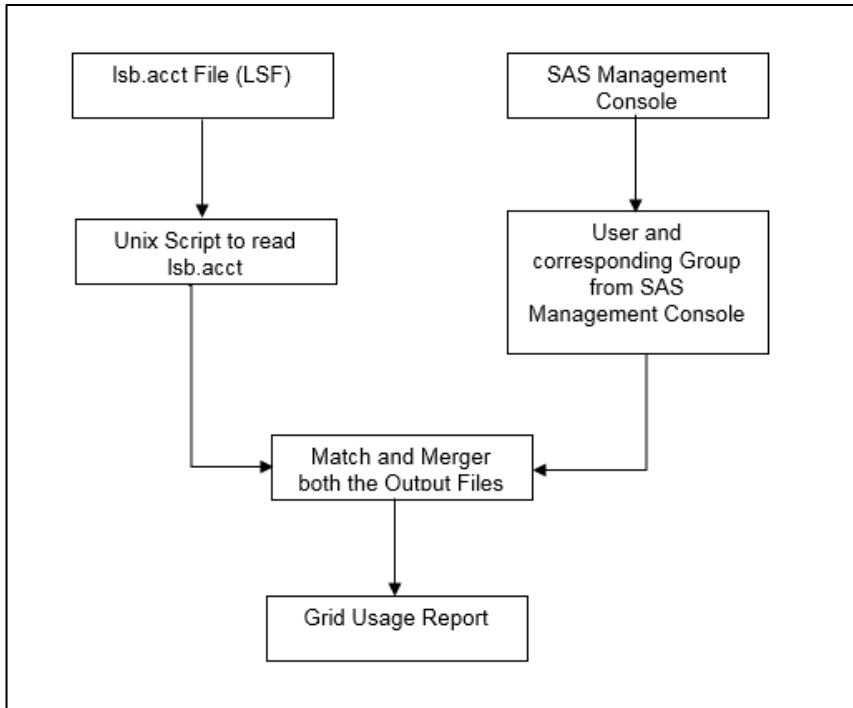


Figure 1. Caption for Sample Figure

CONCLUSION

This paper provides an alternative method to retrieve SAS Grid Job execution information in a much faster and time efficient manner. This method can be utilized to extract a wide range of job metrics like CPU time used, Job Execution node name etc. based on the business requirement. The metrics when extracted using the LSF command `bacct` can take hours to complete. The method mentioned in this paper can be automated as well and scheduled such that reports are generated automatically.

REFERENCES

Sangwan, Praseon. Gupta, Tanuj. and Singh, Piyush. Apr-2016. "Key Requirements for SAS[®] Grid Users" *Proceedings SAS Global Forum*. Las Vegas, NV, USA: SAS Institute. Available at: <http://support.sas.com/resources/papers/proceedings16/7140-2016.pdf>

Singh, Piyush. Sangwan, Praseon. Kumar. "SASGSUB for Job Workflow and SAS Log Files" *Proceedings of South Central SAS User Group 2016*. San Antonio, TX, USA: SCSUG. Available at: <http://www.lexjansen.com/scsug/2016/SASGSUB-for-Job-Workflow-and-SAS-Log-Files.pdf>

Piyush Singh, Praseon Sangwan, Steven D Randolph, "Read SAS[®] Metadata in SAS[®] Enterprise Guide" *Proceedings of SAS Global Forum 2017*. Orlando, FL, USA <http://support.sas.com/resources/papers/proceedings17/1275-2017.pdf>

RECOMMENDED READING

- *Platform LSF Version 9 Release 1.3*
- *SAS[®] 9.4 Language Interfaces to Metadata*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Piyush Singh

piyushkumar.singh@tcs.com

Ghiyasuddin Mohammed Faraz Khan

ghiyasuddin.farazkhan@tcs.com

Prasoon Sangwan

prasoon.sangwan@tcs.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.