

## **Paper SY-02-2016**

### **Key Tips for SAS® Grid Users**

**Venkateswarlu Toluchuri, United Health Group, Hyderabad, India**

#### **ABSTRACT**

SAS Grid Manager, often mistaken as a single entity, has many facets and provides an overarching functional toolset to improve the performance and resource utilization of SAS applications and programs. While SAS Grid is a cluster node system, SAS® Grid Manager is the controller for job scheduling and load balancing which are instrumental to optimize processing. Built on a third-party product Platform LSF™ SAS Grid Manager enables SAS applications to manage workloads more efficiently across for many virtual and physical platforms. Thus, it is critical to understand how SAS® Grid environment can be used effectively to manage available resources for critical application requirements, without dependency on SAS® Administrator.

This paper aims to highlight issues commonly faced by new SAS® Grid users, specifically those have worked previously on a traditional environment (Non-Grid), and subsequently tips to navigate through those issues. Presented here are multiple scenarios and elementary but pragmatic approaches that will enable SAS programmers to work efficiently within SAS® environment.

#### **INTRODUCTION**

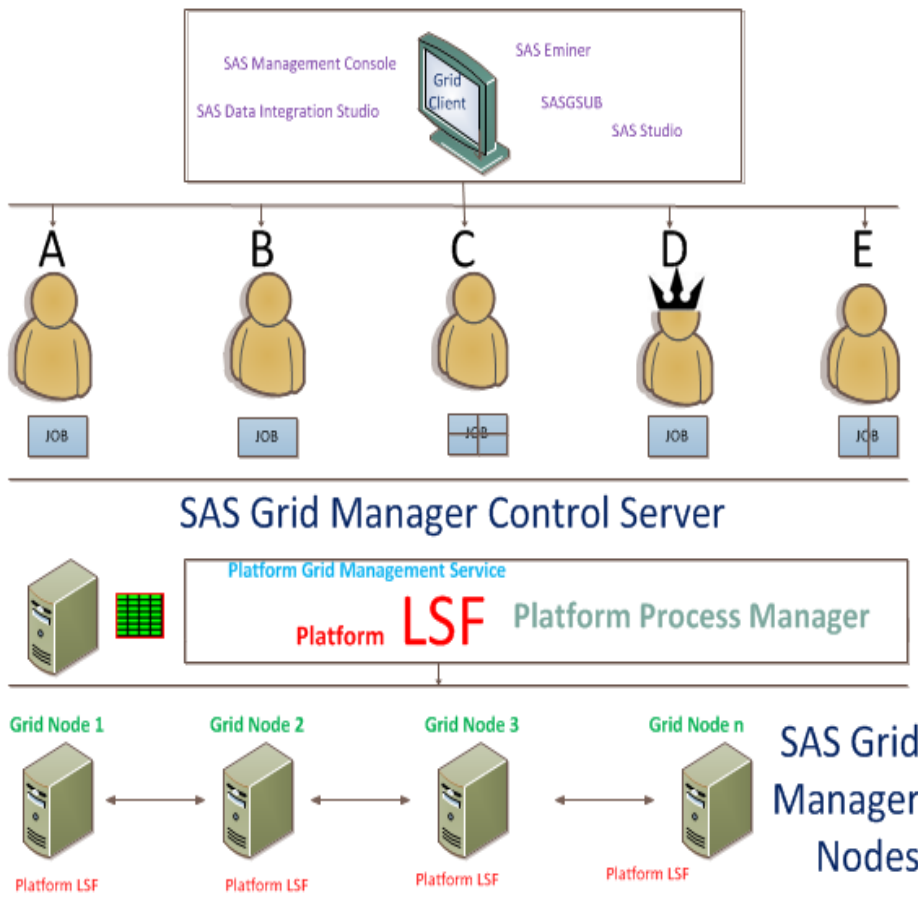
This paper is written from a SAS Programmer's perspective and demonstrates ways to leverage Platform™ LSF™ (Platform™ Load Sharing Facility) commands and tips to efficiently work with a SAS Grid Environment. LSF is a workload management platform that includes job scheduler, equipped with distributed high performance computing environments.

#### **APPROACH**

SAS® Grid is gaining popularity over traditional SAS Environments as SAS® Grid offers many benefits to enterprise organizations, including the ability to optimally utilize and manage available resources for critical applications. With this evolution of an enhanced and efficient computing system, it might be challenging for traditional programmers to adapt quickly to SAS® Grid Servers. Especially, it might be challenging to understand the way LSF commands can get the required information from Grid servers. The LSF system provides required information about the job status, node information and job running status at any particular time interval. This information can help programmers to review and take appropriate action for the set of jobs that are submitted from a single parent SAS job, producing 20 or more jobs. All these

information will be available in SAS® Environment Manager, which programmers can easily access to view process state information. Also, LSF allows the user to identify SAS jobs and their current status within the SAS® Grid environment. The LSF works in sequential environment, where jobs may not trigger immediately and SAS or application Jobs could be triggered on priority level and even at a state when other resources may be currently running. Nevertheless, LSF allows programmers to look into those queues and determine the current state and the anticipated time of completion.

**Grid Environment High-Level Overview:**



- ❖ User A and B have submit their Jobs. They end up being on different nodes.
- ❖ User C has written a SAS processing in such a manner that it broken down into smaller independent chunks.
- ❖ Control server dispatches those jobs on the servers that are less busy.

- ❖ User D gets the highest priority. If he/she submits the SAS Process and all the nodes are busy, user has the ability to replace with one of the jobs previously submitted by other users.
- ❖ User E simultaneously sends two jobs of relatively larger sizes. These jobs will go to two different nodes those have the least amount of load. In case all the systems are completely occupied, these two jobs are queued up until a server is freed up.

**TIPS and Techniques for Troubleshoot:**

**To know which version of LSF you are using:**

**bjobs -V, lshosts -V, bhosts -V, lsid -V:** Any one of these commands gives you the exact same Information.

```

bjobs -V
IBM Platform LSF 9.1.3.0 build 243073, Feb 12 2015
Copyright IBM Corp. 1992, 2014. All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

binary type: aix-64

lsload -V
IBM Platform LSF 9.1.3.0 build 243073, Feb 12 2015
Copyright IBM Corp. 1992, 2014. All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

binary type: aix-64

bhosts -V
IBM Platform LSF 9.1.3.0 build 243073, Feb 12 2015
Copyright IBM Corp. 1992, 2014. All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

```

**To know the details of all Grid Controls/Nodes [along with the number of active nodes, percentage utilization of these nodes]**

**lsload:** Displays load information for hosts. Load information can be displayed either on a ‘per-host’ basis or on a ‘per-resource’ basis. By default, load information for all the hosts in the local cluster are displayed on a ‘per host’ basis. In multi cluster, load information for all hosts are also displayed in equivalent clusters.

HOST_NAME	status	r15s	r1m	r15m	ut	pg	ls	it	tmp	swp	mem
	ok	1.7	1.8	1.8	11%	0.0	28	0	2000M	23.9G	217.9G
	ok	5.2	5.7	6.2	31%	0.0	0	269	2021M	23.9G	450.5G
	ok	6.9	5.8	5.9	29%	0.0	0	3768	1971M	23.8G	454.2G

**bhosts:** Displays hosts and respective static and dynamic resources. By default, this command returns the following information about all hosts: host name, host status, job state statistics, and job slot limits.

HOST_NAME	STATUS	JL/U	MAX	NJOBS	RUN	SSUSP	USUSP	RSV
	closed	-	0	0	0	0	0	0
	ok	-	125	31	31	0	0	0
	ok	-	125	10	10	0	0	0

To know the list of available queues with priority and other useful options:

**bqueues** : Displays information about queues. By default, this command returns the following information about all queues: queue name, queue priority, queue status, task statistics, and job state statistics.

bqueues

QUEUE_NAME	PRIO	STATUS	MAX	JL/U	JL/P	JL/H	NJOBS	PEND	RUN	SUSP
habnrm1	70	Open:Active	10	-	-	-	0	0	0	0
habnrm2	70	Open:Active	20	-	-	-	0	0	0	0
habnrm3	50	Open:Active	40	-	-	-	1	0	1	0
habnrm4	50	Open:Active	25	-	-	-	0	0	0	0
habnrm5	50	Open:Active	5	-	-	-	0	0	0	0
habnrm6	50	Open:Active	5	-	-	-	0	0	0	0
habnrm7	50	Open:Active	20	-	-	-	2	0	2	0
habnrm8	50	Open:Active	40	-	-	-	5	0	5	0
habnrm9	50	Open:Active	5	-	-	-	1	0	1	0
habnrm10	50	Open:Active	5	-	-	-	0	0	0	0
habnrm11	50	Open:Active	40	-	-	-	19	0	19	0
habnrm12	50	Open:Active	10	-	-	-	0	0	0	0
habnrm13	50	Open:Active	30	-	-	-	3	0	3	0
habnrm14	50	Open:Active	10	-	-	-	0	0	0	0
habnrm15	50	Open:Active	20	-	-	-	0	0	0	0

2. bqueues -l <queue\_name>

```

QUEUE: habnrm1
-- No description provided.

PARAMETERS/STATISTICS
PRIO NICE STATUS          MAX JL/U JL/P JL/H NJOBS  PEND  RUN  SSUSP USUSP  RSV
50   0   Open:Active        10  -   -   -   2    0    2    0    0    0
Interval for a host to accept two jobs is 0 seconds

SCHEDULING PARAMETERS
r15s  r1m  r15m  ut    pg    io    ls    it    tmp    swp    mem
loadSched -    -    -    -    -    -    -    -    -    -    -
loadStop  -    -    -    -    -    -    -    -    -    -    -

USERS: ingx wtranti bu_ha
HOSTS: all

```

To know the information history of all the jobs in a list running at a particular time:

**bhist**: Displays historical information about jobs. Searches the event log file that is currently used by the LSF

bhist -w -C 2016/06/10,2016/06/14 -u <user\_name>

Summary of time in seconds spent in various states:												
JOBID	USER	JOB_NAME	PEND	PSUSP	RUN	USUSP	SSUSP	UNKWN	TOTAL			
250511	pap	HABNRM1_MODEL_COR_EI_INET_COM_2016_MONTHLY	0	0	3040	0	0	0	3040	0	0	0
250923	pap	HABNRM1_GET_CRM_Q1	0	0	17	0	0	0	17	0	0	0
250924	pap	HABNRM1_GET_CRM_Q2	0	0	16	0	0	0	16	0	0	0
250925	pap	HABNRM1_GET_CRM_Q3	0	0	14	0	0	0	14	0	0	0
250926	pap	HABNRM1_GET_CRM_Q4	0	0	40	0	0	0	40	0	0	0

bhist -l <job\_id> : History of the job information for a job\_id to know the history.

```
Job <256135>, Job Name <[REDACTED]>, User <[REDACTED]>, Project <default>, Command <[REDACTED] Lev4/SASGrid /GridServer/sasgrid "SASBATCHSASPGM:'daily_history_move_prov.sas'" SASWANTJOBINFO:1 "SASBATCHWRKDIR:'[REDACTED]' /SASGSUB-2016-06-14_00.59.52.55 [REDACTED] SASCOMMANDARG: -UTILLOC [REDACTED] saswork -memsize 4G -sortsize 2G -log '[REDACTED] logs/daily_move_psm_prov_cad.log' -print '[REDACTED]' -sysin '[REDACTED]' -work '[REDACTED] saswork' -noterminal >
Tue Jun 14 00:59:52: Submitted from host <[REDACTED]>, to Queue <habnrml>, CWD <[REDACTED]>, Re-runnable, Requested Resources <SASGrid>;
Tue Jun 14 00:59:53: Dispatched 1 Task(s) on Host(s) <[REDACTED]>, Allocated 1 Slot(s) on Host(s) <[REDACTED]>, Effective RES_REQ <select[(SASGrid) && (type == any)] order[r15s:pg]>;
Tue Jun 14 00:59:53: Starting (Pid 24641980);
Tue Jun 14 00:59:53: Running with execution home </home/[REDACTED]>, Execution CWD <[REDACTED]>, Execution Pid <24641980>;
Tue Jun 14 00:59:55: Done successfully. The CPU time used is 0.5 seconds;
Tue Jun 14 00:59:55: Post job process done successfully;

Summary of time in seconds spent in various states by Tue Jun 14 00:59:55
PENDING PSUSP RUN USUSP SSUSP UNKWN TOTAL
1 0 2 0 0 0 3
```

To know name of queue that belongs to you (As a SAS User)

bqueues -w -u <user\_name>

QUEUE_NAME	PRIO	STATUS	MAX	JL/U	JL/P	JL/H	NJOBS	PEND	RUN	SUSP	RSV
habnrml	50	Open:Active	20	-	-	-	1	0	1	0	0
chkpnt_rerun_queue	40	Open:Active	-	-	-	-	0	0	0	0	0
normal	30	Open:Active	5	-	-	-	2	0	2	0	0

To kill an entire job array that belongs to a user/batch account

**bkill**: Sends a set of signals to kill the specified jobs. By default, it kills the last job submitted by the user if one doesn't give a job id. One must specify a job ID to kill a job.

**bkill 0 <user\_name>** :Used to kill all the processes running for a specific Unix account. This is useful when multiple jobs are running against the same Unix account and it is difficult to pass PID as a parameter for each of the processes.

**bkill 0 <queue\_name>** :Used to kill all the processes running for a specific queue. This is useful when multiple jobs are running within a queue and it is difficult to pass PID as a parameter for each of the processes.

To displays static resource information about the hosts

**lshosts** : By default, returns the following information: host name, host type, host model, CPU factor, number of CPUs, total memory, total swap space, whether the host is a server and static resources. Exclusive resources are prefixed with an 'exclamation' mark (!). Displays information about all hosts in the cluster.

HOST_NAME	type	model	cpuf	ncpus	maxmem	maxswp	server	RESOURCES
[REDACTED]	IBMAIX64	POWER8	250.0	6	248G	24G	Yes	(mg SASGrid SASEG)
[REDACTED]	IBMAIX64	POWER8	250.0	8	498.2G	24G	Yes	(mg SASGrid SASEG)
[REDACTED]	IBMAIX64	POWER8	250.0	8	498.2G	24G	Yes	(mg SASGrid SASEG)

**To displays information about configurable system parameters in lsb.params**

bparams -a

bparams: Displays information about configurable system parameters in lsb.params.

```

DEFAULT_QUEUE = normal
DEFAULT_HOST_SPEC = NULL
MBD_SLEEP_TIME = 1 (seconds)
SBD_SLEEP_TIME = 1 (seconds)
JOB_ACCEPT_INTERVAL = 0
PG_SUSP_IT = 180
CLEAN_PERIOD = 84600
MAX_JOB_NUM = 10000
MAX_SBD_FAIL = 3
HIST_HOURS = 5
DEFAULT_PROJECT = default
JOB_TERMINATE_INTERVAL = 10 (seconds)
MAX_JOB_ARRAY_SIZE = 1000
DISABLE_UACCT_MAP = N
JOB_SPOOL_DIR = NULL
FY_START = 8
MAX_USER_PRIORITY = -1
JOB_PRIORITY_OVER_TIME = -1/-1 (minutes)
JOB_ATT_DIR = [REDACTED]
MAX_JOB_MSG_NUM = 128
MAX_JOB_ATT_SIZE = 2147483647
MBD_REFRESH_TIME = 60 10 (seconds)
SHARED_RESOURCE_UPDATE_FACTOR = 2147483647
CPU_TIME_FACTOR = 0.700000
RUN_TIME_FACTOR = 0.700000
RUN_JOB_FACTOR = 3.000000
ABS_RUNLIMIT = Y
JOB_EXIT_RATE_DURATION = 5 (minutes)
EADMIN_TRIGGER_DURATION = 1 (minutes)
NQS_REQUESTS_FLAGS = 2147483647
NQS_QUEUES_FLAGS = 2147483647
JOB_DEP_LAST_SUB = 1
REMOTE_MAX_PREEEXEC_RETRY = 5
LOCAL_MAX_PREEEXEC_RETRY = 2147483647
LOCAL_MAX_PREEEXEC_RETRY_ACTION = SUSPEND
MC_RUSAGE_UPDATE_INTERVAL = 300 (seconds)
SYSTEM_MAPPING_ACCOUNT = NULL
PREEMPTION_WAIT_TIME = 300 (seconds)
NO_PREEEMPT_RUN_TIME = -1 (minutes)

```

**To report accounting statistics on completed LSF jobs**

**bacct** – Displays accounting statistics about completed jobs. Displays a summary of accounting statistics for all completed jobs (with a DONE or EXIT status) submitted by the user who invokes the command on all hosts, projects and queues in the LSF system. This command also displays statistics for all jobs logged in the current 'Platform LSF' accounting log file. Report is generated from the specific account which is running.

```

Accounting information about jobs that are:
- submitted by users ██████████
- 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:      427      Total number of exited jobs: 194569
Total CPU time consumed: 1223228.7      Average CPU time consumed:      6.3
Maximum CPU time of a job: 5871.0      Minimum CPU time of a job:      0.0
Total wait time in queues: 2195295.0
Average wait time in queue:      11.3
Maximum wait time in queue: 130.0      Minimum wait time in queue:      0.0
Average turnaround time:      49 (seconds/job)
Maximum turnaround time:      88198      Minimum turnaround time:      1
Average hog factor of a job: 0.19 ( cpu time / turnaround time )
Maximum hog factor of a job: 22.33      Minimum hog factor of a job: 0.00
Total Run time consumed: 7373389      Average Run time consumed:      37
Maximum Run time of a job: 18446744073709551615      Minimum Run time of a job:      1
Total throughput:      131.37 (jobs/hour) during 1484.33 hours
Beginning time:      Apr 12 08:45      Ending time:      Jun 13 05:05

```

**bacct -q <queue name> -C 2016/06/03,2016/06/04 -u <username>**

```

Accounting information about jobs that are:
- submitted by users ██████████
- accounted on all projects.
- completed normally or exited
- completed between      Fri Jun  3 00:00:00 2016
      and      Sat Jun  4 23:59:59 2016
- executed on all hosts.
- submitted to queues ██████████
- accounted on all service classes.
-----
SUMMARY:      ( time unit: second )
Total number of done jobs:      1      Total number of exited jobs:      2
Total CPU time consumed:      1.1      Average CPU time consumed:      0.4
Maximum CPU time of a job:      0.4      Minimum CPU time of a job:      0.4
Total wait time in queues:      0.0
Average wait time in queue:      0.0
Maximum wait time in queue:      0.0      Minimum wait time in queue:      0.0
Average turnaround time:      1 (seconds/job)
Maximum turnaround time:      1      Minimum turnaround time:      1
Average hog factor of a job: 0.38 ( cpu time / turnaround time )
Maximum hog factor of a job: 0.38      Minimum hog factor of a job: 0.37
Total Run time consumed:      3      Average Run time consumed:      1
Maximum Run time of a job:      1      Minimum Run time of a job:      1
Total throughput:      0.07 (jobs/hour) during 40.21 hours
Beginning time:      Jun  3 07:47      Ending time:      Jun  4 23:59

```

**bsub -b 4:55 <job\_name>**

**bsub:** Submits a job to LSF by running the specified command and its arguments. One needs to use the keyword 'all' for displaying all options and the keyword description to display a

detailed description of the bsub command. For more details on specific categories and options, one needs to specify bsub -h with the names of the categories and options.

```
Job <251582> is submitted to default queue <normal>.
Job will be scheduled after Mon Jun 13 04:55:00 2016
                                bjobs -w
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
251582 vt7   PEND  normal  [REDACTED]  -          test.sas  Jun 13 04:52
```

bswitch -u <user\_name> <queue\_name>

**bswitch** : Switches unfinished jobs from one queue to another. Switches one or more of the unfinished jobs to a specified queue. LSF administrators and root can switch jobs submitted by other users.

```
Job <251582> is switched to queue habnrml
                                bjobs -w
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
251582 vt7   PEND  habnrml  [REDACTED]  [REDACTED]  -          test.sas  Jun 13 04:52
```

bjobs -d -q habnrml -u prod | grep DONE

bjobs : Displays and filters information about LSF jobs. Specifies one or more job IDs ( and an array index list, if required) to display information about specific jobs (and job arrays).

**To displays all the recently finished jobs submitted by prod account to the queue habnrml**

```
251146  prod  DONE  habnrml  ap  1  ap  *_MOVE_PAT  Jun 13 03:54
251145  prod  DONE  habnrml  ap  ap  *_MOVE_PROV  Jun 13 03:54
251351  prod  DONE  habnrml  ap  ap  *_MOVE_PAT  Jun 13 04:22
251350  prod  DONE  habnrml  ap  ap  *_MOVE_PROV  Jun 13 04:22
251584  prod  DONE  habnrml  ap  ap  *_MOVE_PAT  Jun 13 04:54
251583  prod  DONE  habnrml  ap  ap  *_MOVE_PROV  Jun 13 04:54
251803  prod  DONE  habnrml  ap  ap  *_MOVE_PROV  Jun 13 05:25
251802  prod  DONE  habnrml  ap  ap  *_MOVE_PAT  Jun 13 05:25
```

Displays load information for LSF hosts and periodically updates the display.

**lsmon** is a full-screen LSF monitoring utility that displays and updates load information for hosts in a cluster. By default, it displays load information for all hosts in the cluster, detailed upto the number of lines that fit in the-screen. By default, it displays raw load indices.



HOST_NAME	status	r15s	r1m	r15m	ut	pg	ls	it	tmp	swp	mem
...	ok	0.1	0.1	0.3	1%	0.0	0	1258	2033M	31.8G	455.2G
...	ok	0.1	0.2	0.5	1%	0.0	0	1286	2025M	31.8G	453.5G
...	ok	0.3	0.2	0.5	1%	0.0	0	1254	2033M	31.8G	455.7G
...	ok	0.4	0.4	0.6	0%	0.0	0	17168	2027M	15.9G	46.4G
...	ok	0.8	1.2	0.6	9%	0.0	0	1268	2033M	31.8G	456.5G
...	ok	0.9	0.8	0.8	0%	0.0	0	5160	2018M	15.9G	44.4G
...	ok	1.5	1.4	1.4	9%	0.0	0	1263	2033M	31.8G	455.2G
...	ok	1.8	1.4	1.7	0%	0.0	0	1615	2028M	15.9G	100.3G
...	ok	2.2	1.7	1.2	15%	0.0	10	0	2039M	31.8G	206.6G
...	ok	3.8	3.7	3.6	34%	0.0	7	10	2037M	31.8G	206.3G
...	ok	4.5	4.1	1.1	16%	0.0	0	1274	15G	31.8G	454.2G

**Cheat Sheet for most useful LSF Command:**

S.No	LSF Command	Description
1	bjobs -V	Shows the version of LSF
2	lsload	Displays load information for all hosts in the local cluster, per host
3	bhosts	Displays hosts and their static and dynamic resources
4	bqueues	Displays information about queues
5	bhist	Displays historical information about jobs
6	bkill	Kills a running job
7	lshosts	Displays static resource information about hosts
8	bparams	Displays information about configurable system parameters in lsb.params
9	bacct	Displays accounting statistics about finished jobs
10	bsub	Submits a batch job to the LSF system
11	bswitch	Switches one or more of your unfinished jobs to the specified queue

## **CONCLUSION:**

This paper intends to help users feel more confident for interacting with LSF within a SAS Grid and environment and thus enable them to understand where their specific SAS jobs fit within the environment through appropriate examples. Programmers may find further details and reviews on LSF through the links listed below [References]

## **REFERENCES:**

[https://support.sas.com/rnd/scalability/platform/PSS5.1/lfs7.05\\_command\\_ref.pdf](https://support.sas.com/rnd/scalability/platform/PSS5.1/lfs7.05_command_ref.pdf)

<http://help.unc.edu/help/lfs-monitoring-and-controlling-jobs>

[https://www.ibm.com/support/knowledgecenter/SSETD4\\_9.1.3/lfs\\_kc\\_cmd\\_ref.html](https://www.ibm.com/support/knowledgecenter/SSETD4_9.1.3/lfs_kc_cmd_ref.html)

<https://www.ncsu.edu/hpc/Documents/usefulLSF.php>

## **ACKNOWLEDGMENTS**

I would like to thank Sedhou Ranganathan, Vincent Su and Neeraj Sharma for his constant support and encouragement. I also like to thank my managers Krishna Chakravarthi, Shaun Maloney and my team Kunal, Devki and Dibya for their valuable inputs and support.

## **CONTACT INFORMATION**

We encourage you to kindly share your valuable comments and questions.

Please contact the author at:

Name: Venkateswarlu Toluchuri

Enterprise: United Health Group

Address: Hi-Tech City 2 Building H09, Hyderabad, India-500081

E-mail: venkat.toluchuri@optum.com, venkateswarlutoluchuri@gmail.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.