Paper 3881-2019

Gimme Five: Building Better SAS® Environments with the Five Pillars

Ryan Martis, Demarq Ltd.

ABSTRACT

A large-scale SAS® Grid platform can be a complex environment to administer, and going from an out-of-the-box installation to running the service as part of business as usual (BAU) can appear daunting. Using the five pillars approach set out in this paper, your enterprise can start leveraging the power of a SAS Grid installation from day one to integrate 1) Workload Management; 2) Alerting and Monitoring; 3) Logging and Audit; 4) Resilience and Availability; and 5) Continuity and Recovery, into your BAU methodology. A toolset consisting of shell scripts, SAS programs, SAS command-line interfaces (CLIs), and third-party alerting tools, combined with automation—these guiding principles, along with proven examples, can smooth the path to enabling analytics across your organization.

INTRODUCTION

As a SAS environment manager, your job is to maintain, improve, and provide effective support for your SAS infrastructure. This paper discusses Demarq's Five Pillars approach to Environment Management: (1) Workload Management, (2) Alerting & Monitoring, (3) Logging & Audit, (4) Resilience & Availability, (5) Continuity and Recovery. By addressing the topics raised under each pillar, your SAS Support function can be transformed from a reactive, troubleshooting, and problem resolution operating model, to a more proactive and predictive business practice. As a result, your team can better support your business' analytical projects.

Although aimed at Linux based SAS 9.4 with Platform Load Sharing Facility (LSF) as the Grid middleware provider, this paper discusses topics applicable to any SAS environment. Some knowledge of SAS platform administration and SAS Grid concepts is assumed, the recommended reading section includes more background on certain topics.

1. WORKLOAD MANAGEMENT

A key consideration for a Grid with multiple compute nodes, the workload management pillar deals with how SAS sessions are distributed across your resources. Grids often result from a consolidation of disparate SAS systems, leading to a multi-tenancy environment hosting various business units, all competing for the same resources. How these competing needs are met is a key factor in the perceived success of the Grid.

Each business will require different SAS workload management characteristics in order to provision the necessary computational capacity for their business processes. Platform LSF can be customized to handle the varying workloads at different times of day to handle your peak usage times as well as your critical processes that need priority when executed.

In this pillar, we look to build a strategy for optimizing workload capabilities and improve computational processing at an application and server level.

LSF LOAD MONITORING

Platform LSF plays a key role in load monitoring across your cluster. With computing resources available across many remote servers, LSF uses instances of the LIM (Load Information Manager) daemon on each host to gather utilization information, and pass it to

your grid control server. When the time comes for a job to be dispatched onto the grid, the control server relies on the LIMs for advice on host availability.

Load monitoring becomes more significant as the load increases and the cluster is scaled. Load indices for CPU, memory, disk I/O and concurrent connections are used by LSF to choose where a task is started. This has a direct impact on performance as it enables better load distribution, better CPU utilization rates, better response times and decreased hardware costs.

WORKLOAD EXECUTION

Many services and configuration files have influence over when, where, and how a SAS job is submitted and executed. The Job goes through a process of queuing, scheduling, dispatching and finally execution. Each of these at a high level have parameters stored within their respective configuration files, and are heavily customizable to support user profiles with varying execution, priority, and timing requirements.

Post SAS Installation configurations should be carried out on these cluster attributes. Based on usage patterns, customizing config files such as lsb.params, lsb.hosts and lsb.queues to your business' workload are critical from a performance tuning perspective.

Queue and Priority Configurations

LSF uses queues as a container for all jobs waiting to be scheduled and dispatched to hosts for execution. Jobs submitted to the same queue, share the same scheduling and control policy, and by using multiple queues, you can control the workflow of jobs that are processed on the Grid. Priorities on your workload can be defined in a number of ways on the SAS Grid environment. It is good practice to have a queuing paradigm in place to determine how you plan on balancing different business area's timing and priority requirements.

If employing a priority-based queueing system, you may want to split jobs by business area to allow each separate function the ability to prioritize its own jobs on the SAS Grid at specific times. During these specified batch windows, the queues would elevate in priority, and control a greater number of processing resources on the Grid.

Similarly, a fair share or pre-emption based system are valid ways to deal with resource contention.

Grid-Launched IOM Servers

Another benefit of a SAS Grid, is the ability to enable LSF at an application level. When you have a SAS Grid and SAS Grid Manager, the option to "Launch servers via Grid" will be a part of the algorithm properties.

As seen in Figure 1. Workspace Server Launched via Grid, whereas SAS Sessions typically use the Object Spawner to execute code, IOM applications that are grid launched, will have LSF run the job.

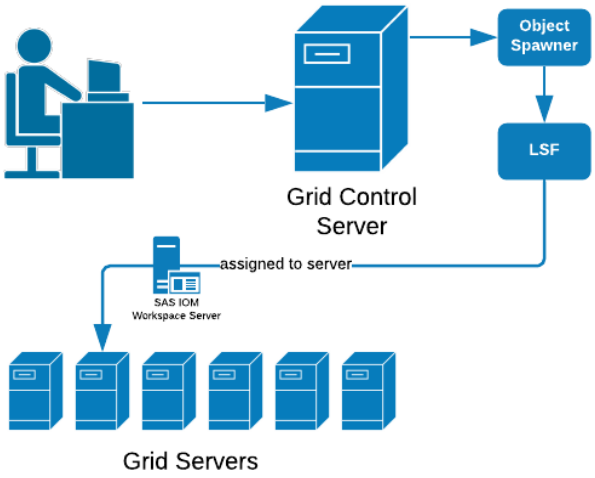

Figure 1. Workspace Server Launched via Grid

SAS sessions are viewed as Grid Jobs occupying job slots managed with Platform Report Track Monitor (RTM) and LSF. With LSF having more tools at its disposal to load balance effectively, grid launched sessions will better handle workload at scale. IOM load balancing is not desirable in a Grid environment as it will not take full advantage of a distributed system in the same way as LSF.

When implementing Grid Launched Workspace Sessions as well as queuing controls, it is recommended to have ENABLE_HOST_INTERSECTION enabled within LSF. This will prevent any errors associated with the number of grid hosts defined in your workspace server definition relative to the grid hosts defined for your LSF queues.

SAS Application Grid Options

Commonly known as grid options sets, when configured correctly they provide an effective means of distributing and managing the resources assigned to users on the SAS Grid. They can be migrated between environments, therefore guaranteeing continuity between logical layers where appropriate.

Grid options sets allow you to map metadata users, group and grid capable SAS applications to queues. Implemented alongside an LSF queuing paradigm (discussed earlier), workload can be redirected based on business area.

2. ALERTING & MONITORING

Alerting and monitoring is a real-time activity, enabling administrators to react quickly to status changes of critical processes. This pillar includes real-time services like automation scripts, server-side background jobs, and any other service with key information about your platform. With effective monitoring tools, your alerting can provide quicker feedback on issues, and thus minimize downtime for business users.

MONITORING

An effective monitoring framework for a SAS environment consists of several tools. Third party products can be used to monitor your servers and sessions in real-time. If your deployment has SAS Environment Manager, you can monitor SAS services and resource

usage within the web application. On a more granular level, job monitoring of IOM and SASGSUB sessions should be monitored for cases of production batch run failures. With a proper monitoring framework established, you can setup alerting high impact failures, with the aim of correcting issues before they impact the business.

ALERTING

Alerting is a combination of monitoring the status of a component at regular intervals, and the alerting of a status change. Ensuring a proper feedback loop from an automated set of scripts or programs is a good way to have monitoring and alerting in place. Further centralizing these alerts to an email address or third-party messaging application such as Slack can simplify administration.

You may consider placing alerting on each key SAS service, such as the Metadata Server and Object Spawner. Presented below is an example alerting structure.

Stored Process Web Service Alert

Running a SAS Stored Process (STP) and having it execute as a RESTful SAS BI Web Service (BIWS) requires your metadata, compute and middle tiers to be in a healthy state (see Figure 2. A SAS STP executing as a SAS BIWS). As such, successful runs of an STP based web service give you a good general health indicator of your platform.

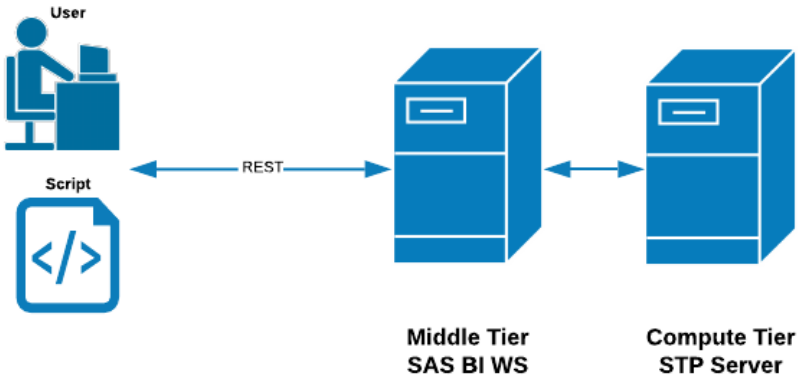

Figure 2. A SAS STP executing as a SAS BIWS

Let's consider an example implementation of an alerting and monitoring script running periodically:

```
#Sending request to webservice
Request=
curl -s --request POST --header "Content-Type: text/xml;charset=UTF-8"
--data-binary @/data/SAS/monitoring/xml_request.xml
http://sasdev:7980/SASBIWS/monitoring/monitor_stp

#file containing the correct results of the request query
Response=/data/SAS/monitoring/xml_response.xml

#comparison taken for success/error condition
diff_output="$(diff $Request $Response)"
```

This stored process running on the Grid, receives an XML request from the script. The STP then responds back with the results. The script, already in possession of the "correct" results, performs a comparison between this and the XML response returned by our SAS Web Service. If the web service is working correctly, the two files should be identical, however in the case of any interruption to this service, a server error would be returned in the diff_output variable. With some additional logic built around this, an alert email can be sent out to concerned parties.

Alert Email to Slack Channel

Continuing the train of thought from our previous example, with an email alert, we can integrate this with a web application such as the Slack API. The IFTTT (If This Then That) service can chain APIs together and presents many possible use cases for integrating applications. The result would look something similar to Figure 3. Stored Process alert in Slack channel below.

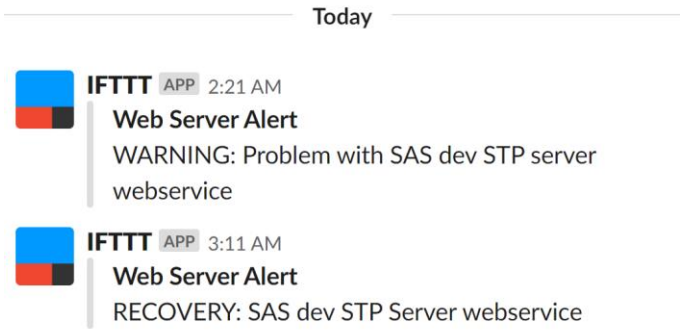

Figure 3. Stored Process alert in Slack channel

Given the use of Slack amongst our team, and the centralized nature of the application, it made sense to post alerts here. This is only an example, and the frameworks used to set this up are very flexible with applications other than Slack. You may need to liaise with your IT security department to allow your server to communicate with third party services.

Creating Other Alert Types

Alerting on other services should be implemented on anything who's status/availability impact the platform as a whole. Services within your environment will often have a status check as a form of validation. Examples of this would be the SAS Metadata Servers, Object Spawners, Storage Monitoring CLIs, and all sas.servers scripts that operate on your platform. Scripting their use across all your servers can offer a centralized and easier way of performing server administration

3. LOGGING AND AUDIT

Logging and Audit is a historical activity enabling administrators to review how the platform is used, and which users have been accessing/utilizing which resources. Resources in this context refers to the disposition of the installed SAS services including the Workspace and Stored Process server sessions; it does not refer to I/O, CPU and Memory usage. The SAS 9.4 technology stack provides a number of user activity audit capabilities to allow for a comprehensive Logging & Audit design to be deployed.

This pillar aims to deliver a comprehensive approach to collect, audit, manage and archive logging on a SAS environment.

There is already detailed documentation on most SAS service loggers, and experience will determine which loggers are most commonly looked at for errors. Furthermore, there are several papers on automated log checking and suggestion on how to better use logs. These can and should be implemented. From an administrator's perspective, you will want to have a document detailing (1) log locations, (2) logging levels, (3) archival strategy, and (4) backups.

LOG LOCATIONS

Logs generated during processing are saved into a specific location. User logs can also be harvested via the altlog option and written into a secure location. Given the volume of user logs likely to be generated, it is recommended these logs be archived and retained in line with audit and business requirements. Certain Job processing logs, such as SASGSUB or SAS Batch Server, will often need user input to troubleshoot and analyze. In these situations, inserting a log appender in the relevant loggers logconfig.xml file will write log outputs to a secondary location such as the user's home directory, or a single directory accessible to the developers.

It should not be neglected to include documentation detailing the locations of all Metadata, Compute Tier, Web Tier, Platform LSF, Platform EGO, Platform GMS and other loggers available on your platform.

LOG ARCHIVING

Based on your requirements, you will want to keep a certain period of logs before they are archived or deleted. A housekeeping script can be deployed to perform tasks like conditionally identifying logs that need to be archived, compressing these files into an archive folder and deleting previous archives older than your retention period. Audit and user logs can grow in size very quickly, so you may want to keep an archive of 14 to 28 days. Other services may be less critical and can be left on the SAS config mount and added to an archiving script as needed.

To illustrate an example of log housekeeping, let's consider the script logic presented below:

```
#metadata server log archiving
files=$(find "$meta1_loc"/SASMeta/MetadataServer/Logs/ -type f -mtime
+7)

#tar the logs and save to archive
tar -zcvf
"$archive_loc"/SASMeta/MetadataServer/Logs/MetadataServer1Logs_$(date +
"%Y%m%d").tar.gz ${files} >> "$log_file" 2>&1

#remove archived files
rm -f ${files}
unset files
```

The first command will search a metadata server node's log directory, for any logs over 7 days old. Once the old logs are found, it will then compress them into a tar archive. The third and final step then deletes the archived files.

You can have this set for any and every logger in your system with varying retention periods. Setting this up on a daily schedule will ensure your loggers do not clutter any parts of your filesystem. These operations should be run outside of peak usage to avoid interfering with any user resource contention. This can be similarly setup to clear down any logging that begins to occupy too much space.

LOG BACKUPS

While specific logs may require retention periods and archiving, there will be loggers that are not often checked or used unless there is an error. Such logs may not require an archiving strategy; however, they may be a part of your storage team's backups. If your SAS Platform is considered a business-critical environment, the filesystem and SAS mounts may find themselves subject to a separate storage backup. In this case, you will have you full mounts retained for specific periods and potentially the ability to selectively restore files.

4. RESILIENCE & AVAILABILITY

With increasingly complex analytics and growing data volumes, businesses are facing an ever-greater need for reduced downtime and improved levels of service continuity. These expectations on a system can be broken down into two characteristics, Service Resilience and High Availability (HA).

When dealing with the SAS Grid, Service Resilience needs to provide a mechanism to monitor the availability of services, and to automatically restart a failed service on the same server or an alternate resource. The concept of HA for SAS covers the requirements of having service redundancy, and sometimes even service relocation. HA is generally a measure in percentage terms for the environment's uptime. As part of any Resilience and HA implementation, you need to identify the target availability percentages required and the key grid components that need to remain available. This section will cover three technologies on a typical Grid that enable these system characteristics: Metadata server clustering, LSF & Enterprise Grid Orchestrator (EGO), and load balancing.

CLUSTERING FOR RESILIENCY

SAS Metadata servers can be clustered as a means of staying online during a partial service failure. The clustering adds redundancy to your metadata tier so that if a single node goes offline, the other nodes in the cluster will remain online. This redundancy keeps the overall metadata cluster available to users with no visible change in service. A minimum three node metadata server cluster should be the default position for a SAS Grid implementation.

The SAS middle tier allows for a both horizontal and vertical clustering. Vertically clustering web application servers on the same machine can be a viable strategy for HA. The web server will provide load balancing across the instances and will detect if a server is down but does not guard against a server level fault.

Horizontally clustering removes the single point of failure by having multiple web servers running the necessary web application server instances across several servers. In this configuration a load balancer is required between the servers to distribute connections.

RESILIENCE USING LSF, EGO AND RTM

The Platform Suite for SAS is a core component of SAS Grid, and is licensed to provide resiliency for services running on your grid nodes. Unless you decide to co-locate Metadata or Middle tier services on your grid servers, EGO cannot be used. Since we typically keep metadata, compute and middle tiers on separate machines, we're looking at EGO for compute tier services only.

EGO is recommended if resiliency of services is a priority for your business. EGO runs on your grid control server and is responsible for monitoring and starting your compute tier services. Define each of your EGO controlled components within the same LSF cluster and configure EGO to manage service startup, failover and service dependencies on the grid. Tying this all together is the Platform RTM dashboard that can be used to implement LSF

and EGO configurations. SAS Environment Manager can also be used to accomplish the same goal.

ALIASING FOR HA

By definition, highly available servers could be running on one of several machines. In the case of a failover, client applications must be able to find the location of the service. One option to accomplish this is using hostname aliasing. This mechanism simplifies connections and provides transparent access to the SAS Platform.

In the event of a failover the alias is updated to point to the server where the service is now running. Keeping the DNS records up to date can be done with a hardware or software solution, discussed below.

Hardware Aliasing

A hardware load-balancing device (HLD) can provide your platform with load balancing and aliasing for a compute tier or middle tier at scale. The HLD would be setup with knowledge of the location of your servers in your clusters and detect the load and availability of each server before forwarding SAS requests. In the event of a failure on one of your servers, the HLD will stop sending requests to the offline server. This would help minimize downtime as your SAS users would not notice the outage, save for the ones already on the defective box. Figure 4. Active Passive Hardware Aliasing shows an example of a hardware aliased compute tier with an active passive configuration.

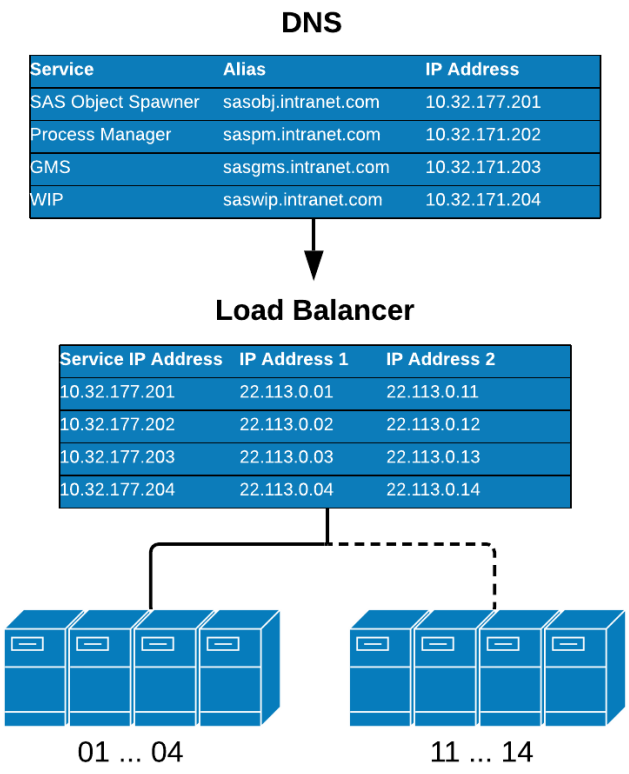

Figure 4. Active Passive Hardware Aliasing

Software Aliasing

Another possibility for aliasing your SAS servers can be done with software. EGO itself can be configured to update your DNS with the IP address of a failover server. Clients now connecting to the same alias name defined in the DNS will be sent to the newly started server without realizing that the server went down. Another possibility is to have EGO setup as a DNS server and configure your corporate DNS to forward aliases to the EGO DNS for resolution.

5. CONTINUITY & RECOVERY

In the case of a disaster event where multiple production servers are lost, a plan should be documented to mitigate time delays in recovery, while maintaining an acceptable level of system performance and data recovery. Businesses that class the platform as business critical will require some form of Disaster Recovery (DR) in place. Your Recovery Point Objective (RPO) and the Recovery Time Objective (RTO) will help determine the type of DR process you may want to adopt. The RTO is the most important when deciding how to recover from a DR event, where the RPO is perhaps more important when deciding your backup and recovery options.

Prior to deciding on a DR model, several factors need to be considered. Your expected recovery time after a DR event is critical, as is the performance relative to your primary environment and data replication abilities. These factors will also drive the cost of any implementation not only from a hardware perspective, but also as far as software licensing is concerned.

Taking the case a SAS Grid Environment, the DR strategy will need to have the ability to failover the permanent storage filesystem, software binaries, configuration files, physical compute servers, and virtualized metadata and mid servers.

VIRTUALIZED HARDWARE & PHYSICAL HARDWARE

When planning for disaster events, a decision must be made on what resources will be failed over. Questions that should be considered in this pillar are:

Is it going to be automatic? What is defined as critical for failover? Will the resources of the secondary environment match the primary? How do we plan to provision the hardware for the secondary site?

FILESYSTEM REPLICATION

Depending on where your software binaries and configuration files are stored, replication will need to occur to keep off-site resources adequately in sync for a failover event. The permanent SASDATA storage and you SAS binary/config filesystem may be both configured differently. It is important to ensure the relevant data and SAS configurations are replicated on your failover site to ensure minimum downtime.

The environment in Figure 5. Data Replication follows an active/passive DR setup, meaning that the replication is asynchronous. There are no real-time copies of the data and you will be recovering from the last replication point.

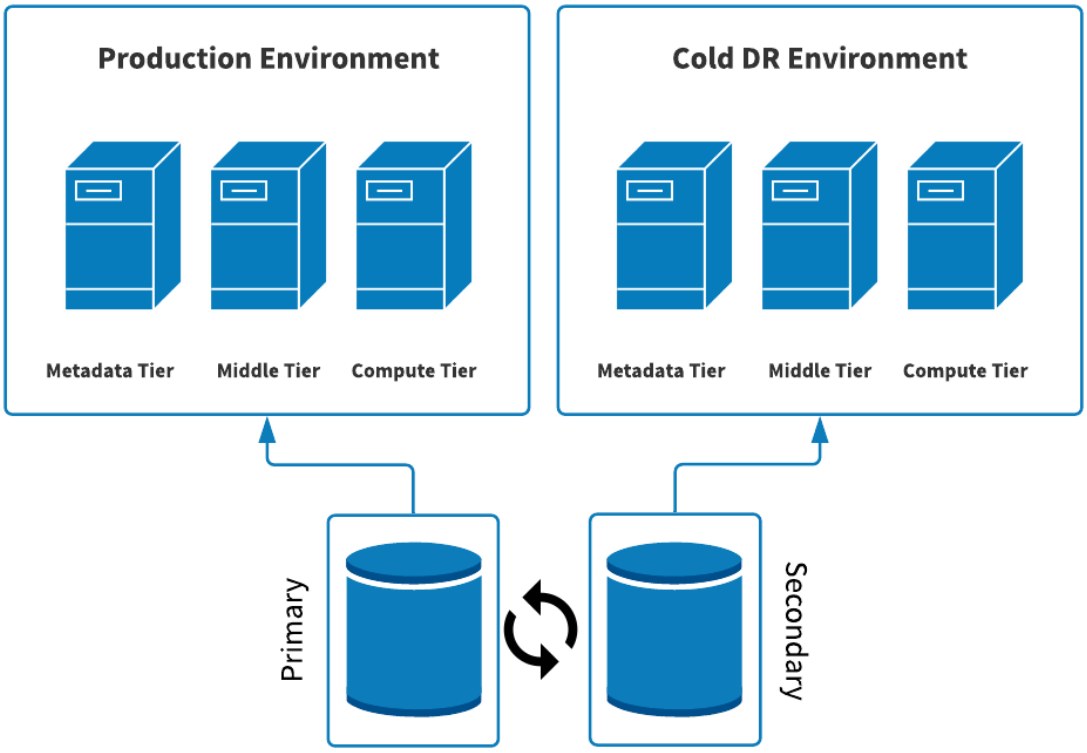

Figure 5. Data Replication

ENVIRONMENT PROVISIONING

Another aspect of DR, is to determine what your post DR environment is expected to handle. Will the computing capacity and resources be identical? Will the access be limited to critical services and select users? Will your business sacrifice a lesser used environment like Pre-Production, and designate that to be the new Production? These are all important considerations in determining the architecture of your DR environment.

CONCLUSION

The Five Pillars of Environment Management are a comprehensive set of principles for taking an out of the box SAS deployment, tailoring it to the individual needs of the business, and transitioning the support role into a business as usual, pro-active engagement. You can configure workload management, alerting and monitoring frameworks and logging configuration to improve performance and facilitate troubleshooting. Furthermore, you can build out a strategy for service resiliency and high availability as well as disaster event continuity to minimize business impact.

REFERENCES

Riva, Edoardo. "Help, I Lost My SAS Server Again!" SAS Blogs, 3 July 2013, https://blogs.sas.com/content/sgf/2013/07/03/help-i-lost-my-sas-server-again/.

Riva, Edoardo. "Help, I Lost My SAS Server!" SAS Blogs, SAS, 14 June 2013, https://blogs.sas.com/content/sgf/2013/06/14/help-i-lost-my-server/.

RECOMMENDED READING

- "Creating a SAS® Stored Process and Executing It as a RESTful SAS® BI Web Service." SAS Technical Support, SAS, 14 Jan. 2019, http://support.sas.com/kb/60/964.html.
- Doninger, Cheryl, and Tom Keefer . "Best Practices for Data Sharing in a Grid Distributed SAS® Environment." Best Practices for Data Sharing in a Grid Distributed SAS® Environment, SAS, July 2010, http://support.sas.com/rnd/scalability/grid/Shared_FileSystem_GRID.pdf.
- Haig, Doug. "The Top Four User-Requested Grid Features Delivered with SAS® Grid Manager 9.4." SAS Technical Papers, SAS, 2013, http://support.sas.com/resources/papers/proceedings13/470-2013.pdf.
- Jackson, Robert, and Scott Parrish. "High Availability Services with SAS® Grid Manager." SAS Grid Computing, SAS, 11 Aug. 2014, http://support.sas.com/rnd/scalability/grid/HA/GridMgrHAServices.pdf.
- Zhou, Songnian, et al. UTOPIA: A Load Sharing Facility for Large, Heterogeneous Distributed Computer Systems. University of Toronto, 1992, UTOPIA: A Load Sharing Facility for Large, Heterogeneous Distributed Computer Systems, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.6568&rep=rep1&type=pdf.

CONTACT INFORMATION

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

Ryan Martis
Demarq Limited
ryan.martis@demarq.uk
www.demarq.uk