Paper 047-2009

Interrogate the Interrogator: Presenting SAS® Usage Information Using BI Tools

Emilio Power, ThotWave Technologies, Chapel Hill, NC
Shawn Edney, ThotWave Technologies, Chapel Hill, NC

ABSTRACT

In simple and complex SAS® deployments, performance monitoring can help ensure ideal efficiency when using resources. With out-of-the-box deployments, this information can be difficult to extract and analyze. SAS® Application Response Measurement (ARM) enables you to collect a variety of metrics on how your SAS® environment is performing. While several resources explain how to implement ARM, there is a lack of information on how to analyze or report the information. This paper presents a series of examples using BI principles to gain insight on using resources in your SAS® BI environment. This insight can help you provide feedback to management, system administrators, and users. Whether the goal is to improve system performance, or monitor hardware in the scope of scaling resources for a growing BI environment, existing and underutilized SAS® monitoring tools can help sustain an efficient BI environment.

KEYWORDS: ARM, Monitoring, Resource, Utilization

INTRODUCTION

As large organizations upgrade to SAS® Enterprise BI Server (EBI) the information and insights gleaned through EBI become a strategic advantage which demands more rigorous monitoring and control of the SAS® EBI environment to reduce downtime and provide maximum performance. EBI processes though can prove to be fairly difficult to monitor due to the wide array of access mechanisms and application servers. One way to collect a wide range of monitoring details is to take advantage of the SAS® Application Response Measurement (ARM) interface to gather process information.

This paper will focus on presenting brief examples of how ARM data can be leveraged to provide powerful insights into SAS® environment usage. The implementation of ARM will not be covered. The recommended reading section at the end of this paper contains a list of sources for further information on implementing ARM.

The ARM information we will focus on will include process timing (when it occurred), process CPU statistics, data source accesses and user information. While this is far from the full range of information available through ARM it delivers sufficient information to perform a range of system monitoring and analysis, providing a mechanism for locating system issues and monitoring users with processes exhibiting inefficient or suspicious behavior.

ARM DATA SCOPE- MICRO VS MACRO APPROACH

From a pragmatic perspective there are two logical approaches to ARM data usage. One perspective is what could be thought of as a “micro” perspective, being focused on the execution details within individual SAS® execution steps (data step, proc step). The second perspective is the “macro” approach (not to be confused with the SAS® Macro language) which focuses on SAS® usage at the user, data resource or process level.

The micro approach relies on programmers adding instructions to their custom SAS® programs to specifically control when an ARM event is logged and what kind of messaging is relayed. For instance if there is a desire to track when a certain event occurs during iterations of a data step then ARM macro calls would be inserted appropriately inside the data step. Another micro approach is to use ARM system options that are added to the code to provide similar types of information on each execution step. The micro methods provide the highest level of customizable logging detail available through ARM and it is the dominate perspective presented in the SAS® 9.1.3 documentation.

The macro perspective is focused on the whole SAS® environment, relying on aggregated process data as opposed to the micro approaches low level details. This means the macro approach is focused on receiving general SAS® process information across all of the logical SAS® servers implemented on any of an organization’s physical servers. In particular information on user processes, data sources accessed, and process timing is captured. This is accomplished primarily by implementing ARM logging via SAS® startup scripts allowing basic ARM data to be collected without user intervention or source code modification. This method also picks up the SAS® processes associated with SAS® clients such as SAS® Enterprise Guide and SAS® solutions such as SAS® Enterprise Miner.

Both the micro and macro approaches to ARM provide useful information but the macro approach is ideal from a system administration and monitoring perspective because it is the approach that provides the “big picture” of usage across an organization’s SAS® environment. It’s this “big picture” or macro perspective that we are concerned with presenting in this paper.

BI CONCEPTS IN MANAGING THE SAS® ENVIRONMENT

When considering the development of an information delivery system for the purposes of managing and monitoring the EBI environment, careful planning must be taken to ensure that the data delivered actually solves or provides insight on business problems. In this context we would like to provide information that would allow a SAS® administrator a mechanism to quickly diagnose ongoing activities in order to become proactive in environment management.

In 2007 ThotWave conducted a survey to identify what are the driving interests in creation of dashboards; “Performance Management” and “Guide Tactics” were the top responses. Taking the proper principles in mind one can develop an efficient environment where maintenance and scalability action items can be identified. There are 4 principles that should be kept in mind when presenting information: (Wright, 2008)

- Present the right information
- Provide context for measures
- Communicate effectively
- Facilitate taking action

In the case of monitoring the SAS® environment and the presentation of ARM data there are various items an administrator would like to know about the ‘health’ of the system. Unlike other BI environments, SAS® provides an open framework where users are capable of building custom applications which run processes ranging from ETL to analytics. While this flexibility allows an organization to create custom applications with great ease it can also lead to a level of inefficiency if proper SAS® application development practices are not followed. Creating an ARM reporting system with a proper design allows the dashboard and EBI interface to provide the right information to enact any action items when needed. Some important high level information that ARM data can provide includes SAS® application efficiency, user activity, data source utilization, and report efficacy.

DASHBOARDS FOR HIGH LEVEL INFORMATION

When considering how to monitor EBI processes a SAS® administrator’s first thoughts might be to look at the total environment to get a sense of how the primary systems are being used. This gives us a good opportunity to use the SAS® Information Delivery Portal (IDP) to deliver information via dashboards and SAS® Web Report Studio (WRS) reports. The dashboard is especially good for presenting high level information from across the system. A sample dashboard is presented in Figure 1.

The “CPU Process Time” dashboard (Figure 1, top left) is meant to provide immediate feedback on the total CPU seconds consumed by SAS® processes on each server over a specified period of time. Often this time period would be set to report on either the current or previous day’s processing. This information item gives immediate feedback on the SAS® processing load being carried out on a particular server as well as a comparison across servers to see how well balanced the load is across servers. Significant imbalance across the available servers might point to a configuration or system issue within a grid environment. If monitoring multiple servers that are not in a grid environment then the dashboard provides immediate feedback on what servers are underutilized. Underutilized servers could then be tasked to perform some of the work of servers reporting high CPU usage.

The “User Count” dashboard (Figure 1, top right) is meant to provide immediate feedback on the number of distinct users that have logged on to each server within a specific period of time. As with the previous dashboard this one would usually provide feedback for the current day or previous day. When reviewing this information as presented the SAS® administrator can gain insight into how well the servers are utilized by users.

The graph, shown in the lower left corner of Figure 1, shows the number of times specific libraries were accessed during a period of time. This particular example counts the individual step accesses such as when a data step, Proc SQL step or other procedure accesses a library for read or write access. For example when a user submits a SQL step to extract from library TEST and writes output to library WORK, the count for TEST and WORK are incremented by one. This type of chart is especially useful when looking for issues related to accessing specific groups of data sources. For example it might be beneficial to show the number of accesses that occurred on all the libraries that are used to access databases. From one standpoint it shows data usage, but from another aspect it could also point to inefficiencies caused by users not using SQL pass through queries when extracting data from databases.

The lower right chart in Figure 1 uses the Information Map Viewer Portlet to present CPU processing time compared to total process time. This information is important in diagnosing potential bottlenecks. Some of the most common bottlenecks are around memory settings or I/O limitations. While the example given in this figure is for an entire environment, showing information for each server or providing drill down reports broken down by users and or processes could be helpful when diagnosing issues. Chew et al.(1999) shows that comparison between user time and actual process time is significant when diagnosing various issues related to RAID stripping, disk caching, and memsize settings.

DIGGING DEEPER

While the high level information presented in Figure 1 is certainly valuable it will be useful to create additional dashboards and reports to monitor specific pain points unique to the environment. For example consider Figure 2 which provides a dashboard and charts presenting metrics on CPU usage, user count by server and data libraries accessed.

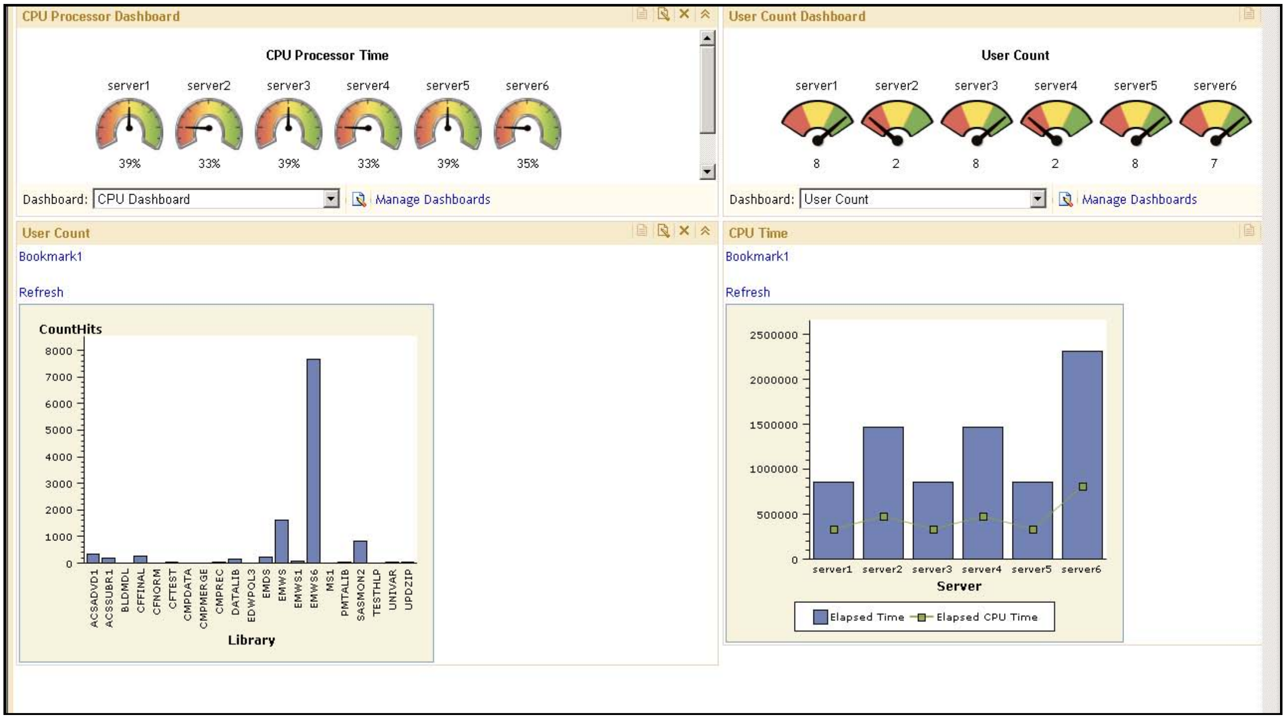
The “Percent Processor Wait Time” dashboard (Figure 2, top left) provides derived information on a specific aspect of the environment. In this case it provides a measure of the average amount of time a SAS® process spends on tasks that are not directly attributable to actual SAS® processing. This time lag difference is often attributed to

- SAS® waiting on I/O resources such as disks
- Delay while waiting for a databases to respond
- Delay while the host operating system schedules the SAS® process threads

High wait times could point to process inefficiencies, network problems or host operating system issues and give the SAS® administrator a proactive tool to deal with issues before they impact the whole environment.

The “Utilization Stack by User by Hour” chart (Figure 2, bottom left) illustrates how SAS® processing can be tracked by user by hour for a specific period of time. This kind of presentation could be useful to a SAS® administrator as a way of reviewing who is consuming significant resources across the environment, or on a single server. It might also be useful to present this information for a subset of users such as those belonging to a particular group. One other interesting point with this type of information is that it can pinpoint users who may be using resources outside of normal business hours. This may be due to scheduled jobs, an overloaded employee or possibly prompt an organization to investigate suspicious activity.

While a lot of the examples we have given lack a time dimension, the inclusion of timestamp data in ARM logging output provide the potential to analyze trends or usage patterns over time. Analyzing trends could help an administrator identify system load conflicts. One example might be to identify that certain user groups are running jobs at the same time as important nightly ETL loads, raising the risk of performance issues that could affect the ETL jobs ability to finish in a timely manner. On the opposite side it can also identify unused resources to help in planning for increased loads or improving scheduling plans. In the given example it is clear that certain users are responsible for significant loads but there is also a large amount of unused resources that could be used for jobs scheduled after 12 AM. In this example we are using a stored process which allows the developer access to all of the SAS® analytics capabilities in his/her environment. Methods such as trending, regression, and predictive modeling can all be leveraged through this delivery mechanism.



CPU Processor Dashboard

Dashboard: CPU Dashboard

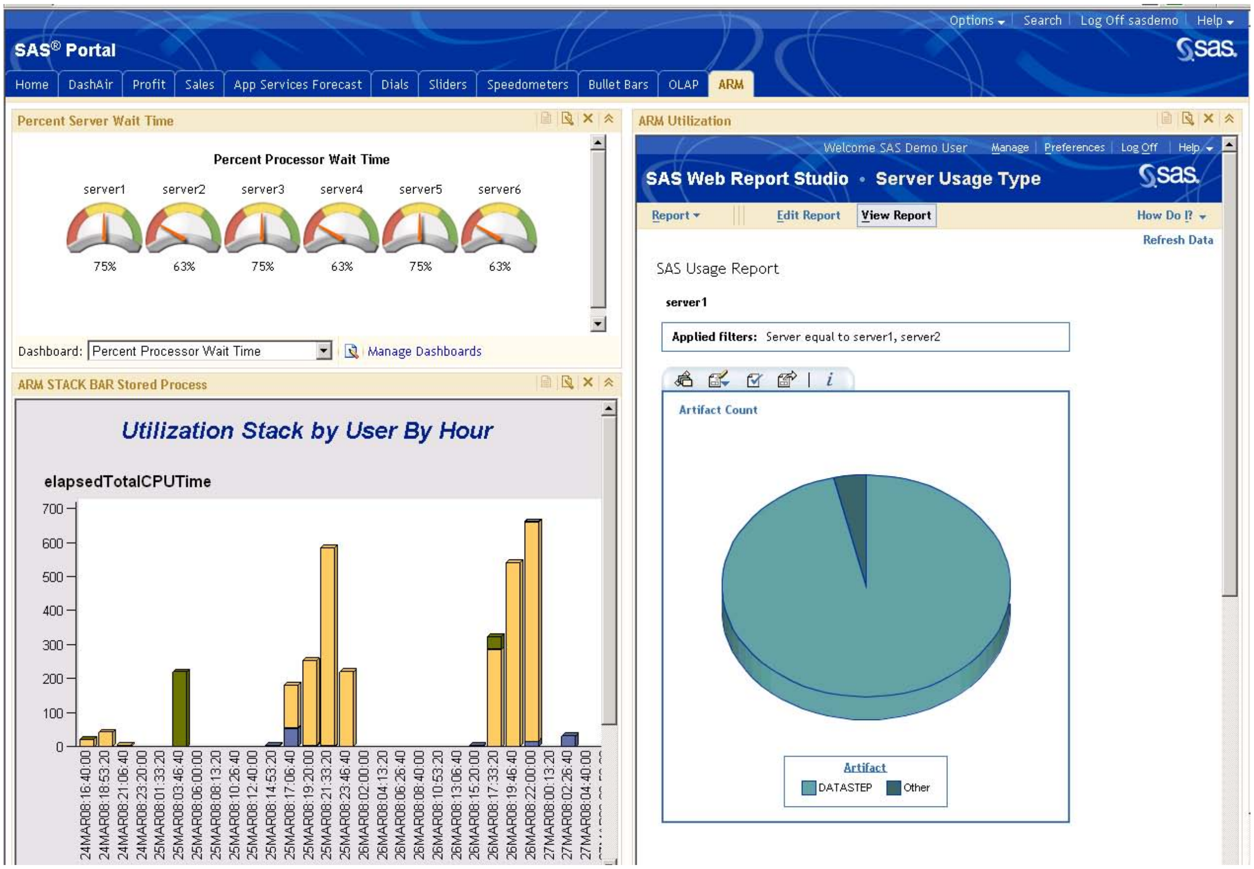
User Count Dashboard

Dashboard: User Count

User Count

CPU Time

FIGURE 1.



Applied filters: Server equal to server1, server2

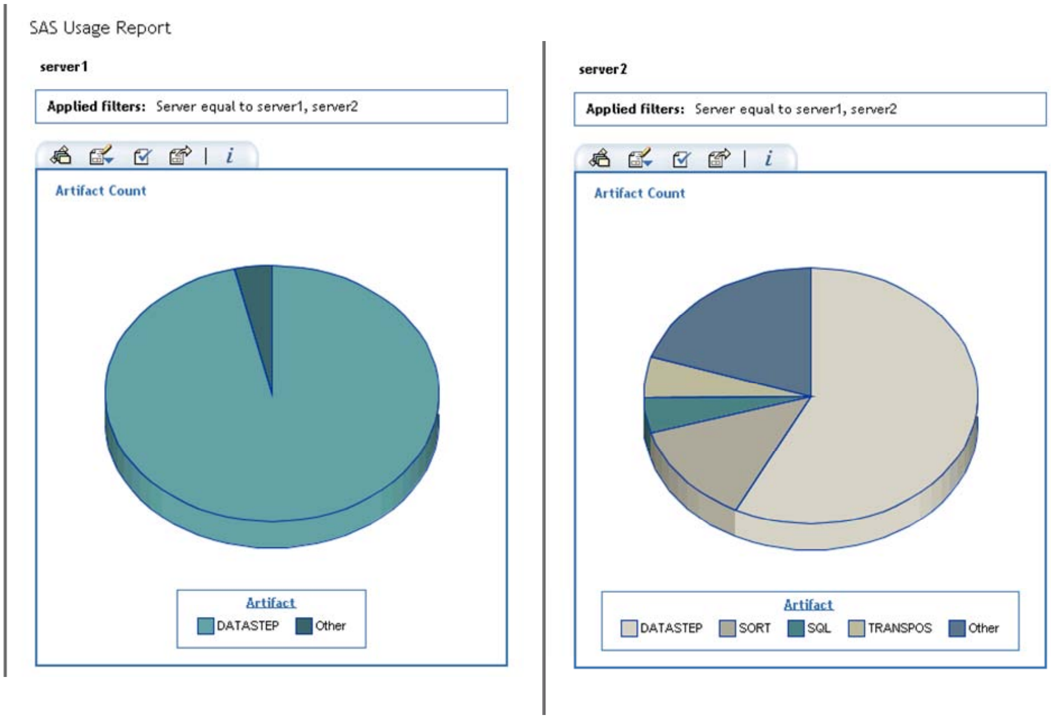
FIGURE 2.

PROFILING SERVER ACTIVITY

One of the first questions asked by an organization working to maintain a server is ‘How do we optimize server settings?’. Since SAS® has such an open ended environment it is impossible to give a simple answer. One of the primary items a SAS® administrator needs to know in order to optimization an environment is the types of applications the SAS® servers are running. Servers with heavy data step applications may require different optimizations compared to a server that is running a larger portion of procedure steps such as SORT or SQL. Leveraging the artifact data provided in ARM output can help the administrator profile the load on a particular server or it can be focused down to a specific user. Looking for deltas on utilization can allow an administrator to predict usage and develop action plans to address potential changes in order to avoid unnecessary bottlenecks. Figure 3 provides an example of this kind of information.

USER UTILIZATION

Analysis of user code processes and activity can be done through ARM and displayed in several ways. Figure 4, a report published through SAS® Web Report Studio, illustrates how it is possible to monitor a generalized profile of user SAS® processes. Artifact tracing by user was tracked for a development group and User05 and User08 were given very similar ETL tasks. By analyzing this report we can see that User05 is writing code that is leveraging PROC SQL steps vs. User08’s code that is relying on many data step transformations. This type of information can give developers and or administrators important tools when trying to understand why there might be performance issues.



SAS Usage Report

server1

Applied filters: Server equal to server1, server2

server2

Applied filters: Server equal to server1, server2

FIGURE 3:

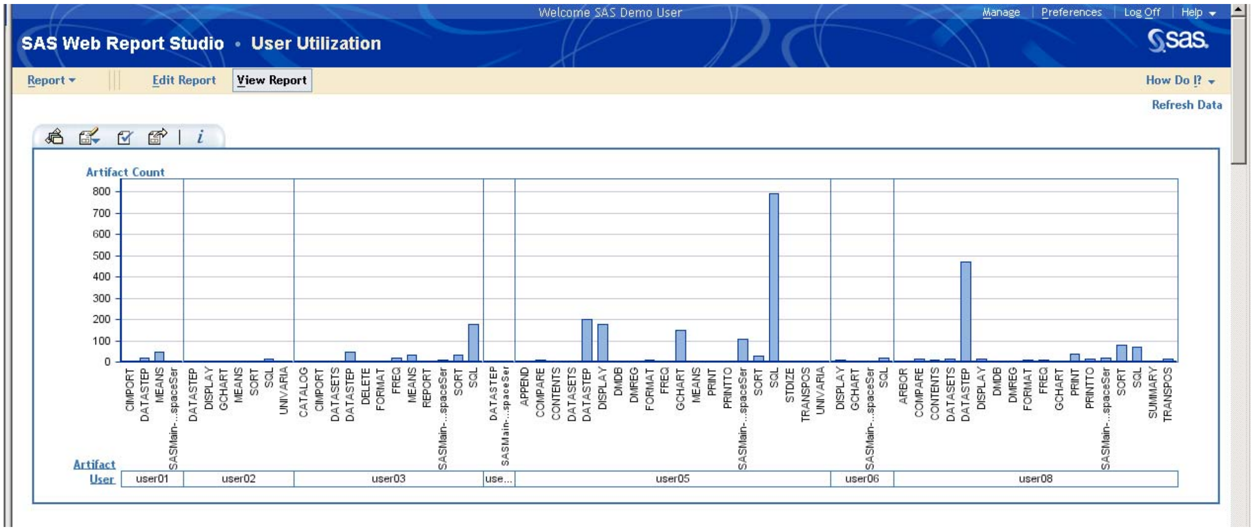


FIGURE 4:

CONCLUSION

EBI tools have been designed with the intention to address an organization's business problems. Creating an ARM reporting interface in the EBI environment allows an organization to expand its deployment of SAS® EBI promoting increased efficiency which should lead to a greater return on investment. When implementing the EBI methods covered in this paper one must remember to ensure that the main principals listed by Wright(2008) are met.

PRESENT THE RIGHT INFORMATION.

The ARM examples are presented through a trusted data source giving information that is unavailable through normal SAS® logging procedures.

PROVIDE CONTEXT FOR MEASURES.

The ability to drill down and present data by server, group, user, and specific processes allows for an administrator to look at information with the correct granularity to address many potential issues.

COMMUNICATE EFFECTIVELY

The ability to use multiple SAS® analysis and reporting tools with a single delivery platform allows the presentation of ARM data in an efficient manner. Dashboards provide a wide range of information delivery graphics that provide a quick snapshot of data and trends. Web Report Studio allows for detailed reports that are quickly customizable by the user. Information Maps allows for specific data to be surfaced in a quick manner and Stored Processes allow for advanced analytic techniques including leveraging the full power of SAS® analytics.

FACILITATE TAKING ACTION

The SAS® EBI tools allows for near live as well as historical data delivery but also have the capabilities of drilling down to specific detail information that is often needed for action items to be completed.

In this paper we have presented examples that deliver information using EBI using SAS® Dashboards, SAS® Web Report Studio, SAS® Information Maps and SAS® Stored Processes all presented through a single application the SAS® Information Delivery Portal. With proper design it is clear that SAS® Enterprise BI is outstanding at presenting ARM data. We have provided a series of examples on how to leverage ARM data and show how SAS® Administrators can benefit from having an environment that monitors itself. While this is not an exhaustive review of either ARM or EBI capabilities the information reviewed offers an excellent starting point on the way to develop an EBI solution to monitor your EBI investment.

REFERENCES

Chew et al.(1999) Peace between SAS Users & Solaris/Unix System Administrators:
Finding a Middle Ground Available at: http://www.sas.com/partners/directory/sun/performance/#memsize
.
Wright, JEFF SAS®: The Ultimate Dashboard Machine, ThotWave Technologies, Chapel Hill, NC. Available at: www.thotwave.com

RECOMMENDED READING

Rubendall, Craig (2008) I Didn't Know That Was Being Used! Keeping Track of What Is Being
Used in Your SAS® EBI Environment: http://www2.sas.com/proceedings/forum2008/311-2008.pdf

ARM - Frequently Asked Questions: http://support.sas.com/rnd/scalability/tools/arm/armfaq.html

CONTACT INFORMATION

Contact the authors at:

Emilio Power
ThotWave Technologies
510 Meadowmont Village Circle #192
Chapel Hill, NC 27517
800-591-THOT
epower@thotwave.com
http://www.thotwave.com

Shawn Edney
ThotWave Technologies
510 Meadowmont Village Circle #192
Chapel Hill, NC 27517
800-591-THOT
sedney@thotwave.com
http://www.thotwave.com