Serving SAS®: A Visual Guide to SAS Servers
(Contributed Paper Edition)

Gregory S. Nelson

President and CEO
ThotWave Technologies, Cary, North Carolina

Abstract

SAS® has been running on servers since the late 1960s. Despite the emergence of PCs and workstation-class machines, SAS still reigns supreme on the server. With the introduction of the SAS platform 9 in 2004, the number and types of servers have grown exponentially.

As any good student of the DATA step will attest, knowing what SAS is doing is a critically important step in debugging and authoring efficient programs.

In this paper, you will experience SAS through a visual tour - you will see what SAS is doing, how it works, which server is doing what, when the operating system plays a role, how security functions, and what happens to your data through the entire process.

Introduction

In the late 1960's "S.A.S." began as the "Statistical Analysis System" – a project incubated at North Carolina State University to help agricultural research stations throughout the Southeastern United States with tools for the analysis of data. Fast forward another three decades, and we find SAS (sans-acronym) in over 50,000 sites worldwide, and matured well beyond its research roots. Today, SAS is used in virtually every industry to solve enterprise class problems for companies large and small.

When SAS 9 was introduced in 2004, SAS moved beyond its foundational roots to a metadata-managed, enterprise class, analytics software. Along with this maturation, SAS also introduced a number of "servers" that previously were unknown to the tens of thousands of SAS-enthusiasts.

This paper and the associated presentation is an attempt to help both users and administrators understand how SAS works – and in particular, what all of these servers are doing. The paper utilizes words and static images while the presentation provides a dynamic visualization of how SAS works.

Foundation SAS versus Metadata Managed Solutions

Over the past seven years, SAS has introduced us to some new terminology. It used to be that we had SAS modules such as Base, STAT, GRAPH and so on (of course, we are taking liberties with the actual SAS terms – for example, SAS/STAT®). While we still have those "modules", we now refer to the core of what SAS does with all of these products as "SAS Foundation". So as SAS (the company) has built on solutions such as SAS Business Intelligence, at the core, the SAS Foundation products are still there doing the actual work. The biggest difference now is that SAS Foundation is called by a SAS "Server" (such as the SAS Workspace Server or Stored Process Server), which in turn is called by the Object Spawner; the Object Spawner is a response to a request from any number of clients – the request is authenticated by the operating system, which in turn is authorized by the metadata server (and that is a simple scenario!)

In a nutshell, SAS can be distinguished by whether or not it is a "foundation" install versus a metadata-managed implementation. The former is a standalone SAS environment, which many users have grown up knowing. The latter, introduced with version 9, is what gives us the power of SAS for large, multi-user, enterprise class capabilities.

Clients in SAS 9

The SAS 9 architecture relies on a new component, the SAS Metadata Server, to provide an information layer between the programs and the data they access. By providing a single point of access for this kind of information, SAS servers can be located just about anywhere (consolidated or distributed) and on any platform (operating system and hardware), to be accessed by SAS clients.

The SAS Institute has made a large investment in these new products and technologies, which represent the future of SAS. It must be recognized that some of the new products, such as Information Map Studio, Web Report Studio, and the OLAP server, are aimed at solving general Business Intelligence problem areas. The figure below shows how some of the clients interface with SAS servers.
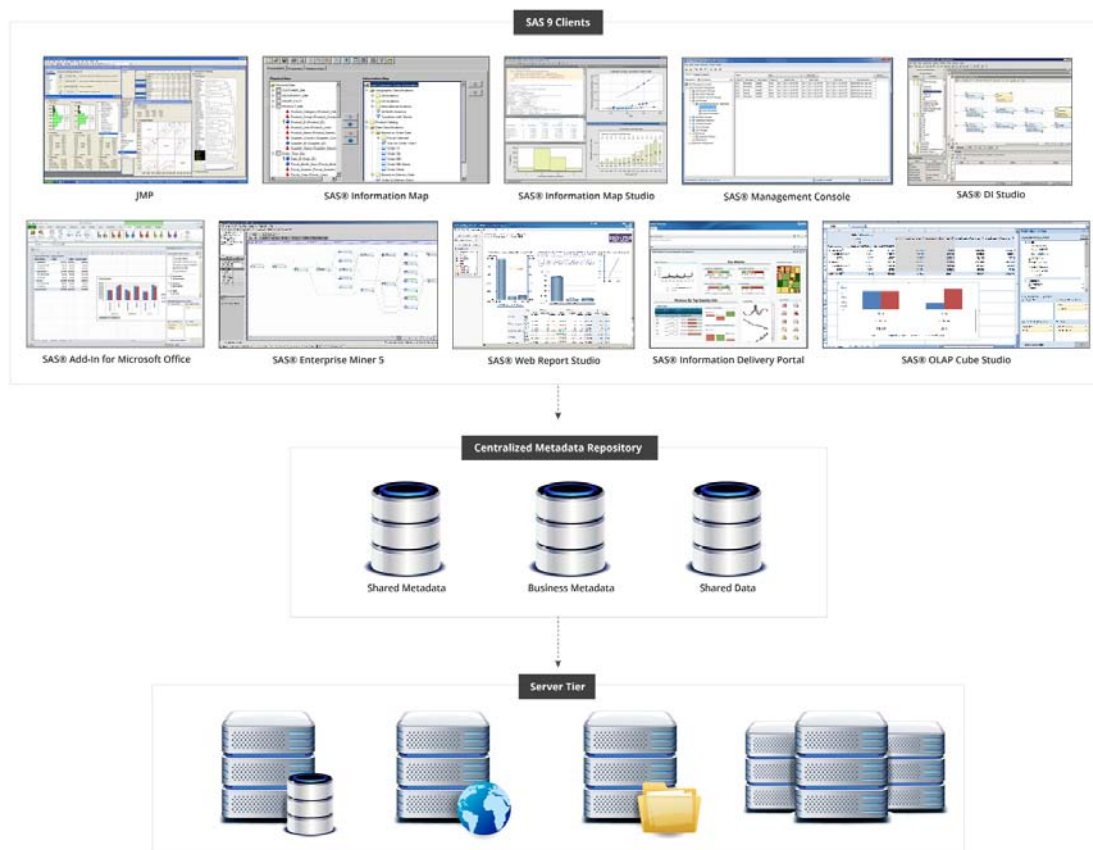

Figure 1. *SAS Client Interactions with SAS Metadata*

In this diagram, typical SAS development utilizes clients running on personal computers that interact with servers running various components that make up the SAS solution. This model

supports the concept of centralized server resources.

Let's turn our attention from the ethereal to the real – and visualize how these models work. In the following pages, we will outline four basic scenarios and turn words into pictures that demonstrate how it all works.

And while there may not be a 'normal', some common implementations of SAS are worth noting. In the section entitled "Sample Architectures" you will see some of these in both words and pictures.

Operating System and Security Implications

SAS currently runs on a variety of UNIX, Windows and z/OS environments (see Supported Operating Environments for UNIX, Supported Operating Environments for Windows and Supported Operating Environments for z/OS). Without extensive discussion of the implications of various operating systems, we will highlight those interactions that are critical to understanding, including the role of the operating system in the following areas:

- Authentication provider

- File system

- Schedulers

- File versioning

For a more detailed discussion of how SAS interfaces with the operating environment, refer to Nelson and Loether, 2007, or Nelson and Swirski, 1997.

Sample Architectures

Each implementation of SAS 9 can be fundamentally the same yet different, depending on the profile of the business problem being solved. These profiles make use of the new services in SAS 9 in varying degrees. The scenarios below are just a few of the architectural options that are possible with SAS.

SAS Foundation

While perhaps the easiest to understand, SAS Foundation-based architectures can indeed be enterprise class. In fact, some companies that began using SAS 20-30 years ago continue to use SAS today – and in some cases, use SAS Foundation on large mainframe, UNIX and Windows servers.

This is an example of a simple SAS Foundation architecture.

Local/ Direct Attached Storage

Figure 2. *Single-tier SAS Foundation Architecture*

Here, we have a personal computer with SAS installed, locally reading and writing files to disk (either local/ direct attached storage or SAN, NAS or clustered file systems). In fact, the simplest form of a SAS Foundation is installed on a single computer. The user writes code and submits it, SAS processes the request, reads/ writes data, and results are generated – all within the context of the computer where it originated. SAS can read and write to files on behalf of the users (as the user) so security is simple – if you can read or write to a file using the operating system, you can do so from SAS.

Now replace the laptop with a mainframe, UNIX server or Windows server – all are examples of SAS Foundation at work. Extending this notion, we can move the data to a file server or a database, or other data shared through SAS servers such as SPDS or SAS/Share, or external data sources such as databases or non-SAS data formats.
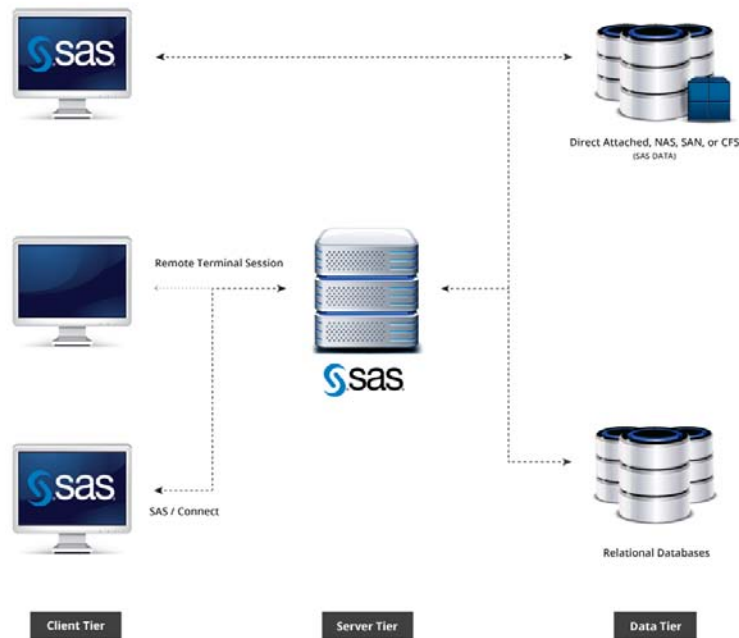
Figure 3. *Examples of SAS Foundation Architectures*

Modern approaches to implementing SAS Foundation include having users connect to servers through technologies such as Remote Desktop, Citrix XenApp/ XenDesktop or Microsoft AppV. In these cases SAS may be running in a dedicated or virtualized environment; but don't be fooled – when SAS is installed on the same computer where the user interacts with it, it is likely that you are dealing with a SAS Foundation architecture.

In the case of SAS Foundation, a metadata server is not required, just access to the SAS executable. Some people might use SAS in batch (command line) or interactively (such as the SAS Display Manager System), but they are still submitting programs to SAS for execution.

SAS Metadata Managed Architectures

The SAS 9 platform can be configured in any number of ways, including a myriad of server topologies and client offerings.   In these architectures, the SAS client tools take on the form of special purpose applications designed to satisfy the needs of various types of SAS users. Tools used to access information now include a substantially improved SAS Enterprise Guide, SAS Add-in for Microsoft Office, and the web-based clients such as the SAS Information Delivery Portal and SAS Web   Report Studio, as well as Base SAS components

(which we now call SAS Foundation).

Here we will provide examples of just a few of the SAS Solutions:

- SAS Data Integration

- SAS Business Intelligence

- SAS Grid Manager

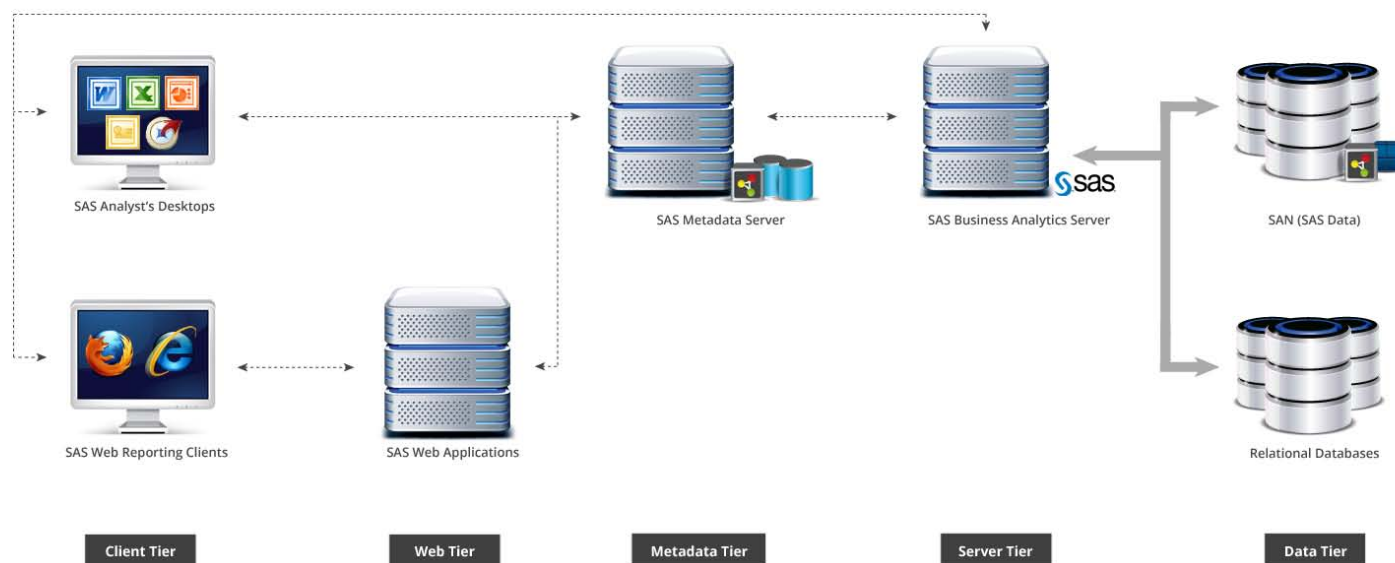Each of these will have similar architectures to the one pictured here.

Figure 4. *SAS Metadata-based Architecture*

SAS DATA INTEGRATION

As mentioned previously, the Workspace Server is simply a server that takes requests and turns them into something that SAS Foundation can process.   Let's apply what you have learned about SAS Foundation, and use that knowledge to extend the architecture to include SAS technologies, which can help you manage data warehouses – and in particular, the ETL process (extract-transform-load).

In the SAS Data Integration implementation[1], you will see some new components:

---

[1] Note: " typical implementation" doesn't mean that your installation will necessarily include all of these nor is this list comprehensive

- Clients
    - SAS Data Integration Studio
    - SAS Management Console
- Servers
    - SAS Data Integration Server
        - SAS Metadata Server
        - SAS Workspace Server
        - SAS Integration Technologies
        - SAS/SHARE
        - SAS/CONNECT
    - Platform Computing Suite for SAS (Platform Computing)

Let's see how it works! Here we will explore a simple use case where someone is using SAS Data Integration Studio to develop an ETL process.

Step #	What the User Sees	What SAS Does

❶	User launches SAS DI Studio
	
	DIS prompts for connection information (user, password, connection details for SAS metadata server)
		Metadata Server:
		1. Verifies user credentials via authentication mechanism[2] that can include:
		    a. Host authentication
		    b. LDAP or Active Directory
		    c. Integrated Windows Authentication (IWA)
		2. Identifies user in SAS metadata repository.
		3. Resolves assignment of SAS metadata roles and access controls.
		4. Connection is established between client application and metadata server.

❷	User creates a new process flow and adds which includes table metadata	Metadata Server receives request for table metadata, ensures that user has correct access permissions and delivers metadata back to client application.

❸	User submits a job for immediate execution[3]	1. Metadata server authorizes user to create a new workspace server connection.
		2. Job is submitted to SAS Application Server (and to any server specified in metadata for a transformation within the job).
		3. Object spawner creates a new workspace server session on behalf of user. It might be worth noting that there is an additional authentication event here where the host authenticates the user through one of the following mechanisms:
		    a. Host
		    b. IWA
		    c. SAS Token (but not direct AD/LDAP)
		4. SAS Workspace Servers interact with SAS by creating a server process for each client connection; client user who made server request owns process. (Only with host and IWA. If configured for SAS Token Authentication, a service account owns the process instead.)
		5. Each workspace server process enables client programs to access SAS libraries, perform tasks by using SAS language, and retrieve results.

❹	The job is closed and/or the user closes/ quits DI Studio	1. Workspace session that was opened on behalf of user is shut down when clients have completed their work.

SAS BUSINESS INTELLIGENCE

SAS Data Integration discussed above is most often used for what we call "beating data into submissions". Once sufficiently beaten (prepared for use), we can now turn our attention to report development and consumption. Often, programmers will use interactive development tools such as Enterprise Guide to aid in the generation of results and then

---

[2] Note: Web authentication is a fourth option for authentication, but since we are talking about DI Studio, it is not relevant here.

[3] Note: There are a number of additional options for running a job not described in detail here. These including: deploy the job for scheduling, deploy the job as a SAS stored process or deploy a stored process as a Web service.

utilize SAS stored processes or web reports to allow others to consume the data in useful ways.

Two of the most common SAS clients are SAS Enterprise Guide and Web Report Studio.

Interactive Programming (with Enterprise Guide)

SAS Enterprise Guide is a Windows application that allows either traditional programming or a point-and-click interface that guides users through the analytical process, making it easy to create reports, graphs and charts. SAS Enterprise Guide is very similar to SAS Data Integration Studio in that it is a client application that is authorized by the SAS Metadata Server and then creates a workspace session for the submission of SAS code.

Step #	What the User Sees	What SAS Does
❶	User launches SAS Enterprise Guide EG prompts for connection information (user, password, connection details for SAS metadata server)	Metadata Server: 1. Verifies user credentials via authentication mechanism. 2. Identifies user in SAS metadata repository. 3. Resolves assignment of SAS metadata roles and access controls. 4. Connection is established between client application and metadata server.
❷	User expands server "SASApp" to view libraries and files	1. Metadata server authorizes user to create a new workspace server connection. 2. Object spawner creates a new workspace server session on behalf of user. 3. SAS Workspace Servers interact with SAS by creating a server process for each client connection; client user who made server request owns process.
❸	User submits a program execution	1. Workspace session(s) associated with process/ program processes program(s), accesses SAS libraries, performs tasks by using SAS language, and retrieves results. 2. Results are streamed back to client application and workspace server sits idle until user makes another request or closes application.
❹	User closes/ quits SAS Enterprise Guide[4]	Workspace session that was opened on behalf of user is shut down when clients have completed their work.

Given the paper submission guidelines for publishing SAS Global Forum contributed papers, we refer you to the full-length version of this paper found www.thotwave.com for additional SAS solutions including SAS Web Report Studio and SAS Grid using both Enterprise Guide and SASGSUB.

---

[4] Note: there is an option within Enterprise Guide to explicitly shutdown the Workspace Server and remain in the application.

Summary

SAS is a feature rich application. As architectures have evolved, a corresponding increase in complexity has also occurred. Knowing where things are happening is the first step in really understanding the modern, metadata-managed SAS environment. The processes described in this paper include some of the more common implementations of SAS, and will give you a better understanding of SAS and how it works. As this paper was intended to be a "visual" guide to SAS servers, we encourage you to view the visual treatment of this content at www.ThotWave.com - just search for "How SAS Works".

References and Recommended Reading

Fenton, Ladd, and Spakes, *"A Practical Approach to Re-Architecting a SAS® Deployment"*, Invited paper presented at the SAS Global Forum, 2011.

Nelson, G. and Loether, J. 2007 *"Architecting SAS in a Modern World: Best Practices for Design, Configuration and Management of SAS® 9".*   Invited paper presented at the SAS Global Forum. Orlando, Fl. April, 2007.

Nelson and Swirski. *"Choosing the Right Platform for SAS® Client/ Server Applications: An evaluation of factors in selecting NT and UNIX"* Invited paper presented at the SAS Users Group International. Nashville, TN. March, 1998.

Schneider, Bennett, and Robison, *"Understanding the Anatomy of a SAS® Deployment: What's in My Server Soup?"* Invited paper presented at the SAS Global Forum, 2011.

Talley and Kalich *"Troubleshooting Tips and Techniques for a SAS 9.2 Business Intelligence Environment",* Invited paper presented at the SAS Global Forum, 2011.

Biography

Greg Nelson, President and CEO of ThotWave Technologies, LLC.

Greg is a certified practitioner with over two decades of broad Business Intelligence and Analytics experience, gained in several life sciences and global organizations as well as government and academic settings. He has extensive software development life cycle experience and knowledge of informatics and regulatory requirements and has been responsible for the delivery of numerous projects in private and commercial environments. Greg's passion begins and ends with helping organizations create *thinking data*® – data which is more predictive, more accessible, more useable and more coherent.

His current area of interest is helping companies take advantage of the shifting world of convergence around data and systems and how modernization and interoperability will change the way that we discover new relationships, manage change and use data and analytics to improve organizational outcomes.

Mr. Nelson has published and presented over a 150 professional papers in the United States and Europe. He holds a B.A. in Psychology along with PhD level work in Social Psychology and Quantitative Methods. He also holds certifications in project management, Six Sigma, balanced scorecard and healthcare IT.

Greg can be reached at greg@thotwave.com or www.linkedin.com/in/thotwave or

Greg Nelson　　　greg@thotwave.com

ThotWave Technologies, LLC

Chapel Hill, NC 27517 (800) 584 2819

http://www.thotwave.com

thinking data® is registered trademark of ThotWave Technologies, LLC.

Other brand and product names are trademarks of their respective companies.

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.