ABSTRACT
This presentation is for users wishing to take advantage of the boost in processing speed for Base SAS programs that will execute in SAS Viya. The paper leverages the power of SAS Cloud Analytic Services (CAS) to access, manage, and manipulate in-memory tables.

This presentation will attempt to answer three primary questions:
1. What is New in SAS® Viya
2. How is SAS®9 Different From SAS® Viya
3. Where do I Integrate Existing SAS® Code in SAS® Viya

INTRODUCTION
SAS Viya is a cloud-enabled, in-memory analytics engine that provides quick, accurate and reliable analytical insights. The latest enhancement of the SAS platform, SAS Viya is an open, cloud-enabled, analytic runtime environment with a number of supporting services. One of those supporting services is SAS Cloud Analytic Services, or CAS. CAS provides a powerful in-memory engine that delivers blazing speed to accurately process your big data. It uses scalable, high-performance, multi-threaded algorithms to rapidly perform analytical processing on in-memory data of any size.

Terminology

File is used to refer to the source data that is in a caslib’s data source. For a Caslib that uses a path-based data source, this is natural. For a Caslib that uses a database as a data source, the tables in the database are referred to as files.

Table is used to refer to in-memory data. After a file (using the preceding definition) is loaded into the server, it is referred to as a table.

CAS - in Viya, SAS Cloud Analytic Services (CAS) is the star of the show, providing lightning-fast analytics of in-memory data for SAS Visual Analytics and other software offerings.

SPRE - Foundation SAS, the long-time workhorse of SAS analytics is also offered, referred to as the SAS Programming Runtime Environment (SPRE). SPRE provides a user interface and data processing environment for executing classic SAS program code. It offers the Foundation SAS software we’re all familiar with, including Base SAS, SAS/ACCESS engines, and more as well as the SAS Studio web application.
1.1 What’s New In SAS® Viya®

What’s New – Cloud Enabled

The big change is that Viya is cloud enabled. That means there’s a fundamental change in architecture. It is cloud enabled, allowing scalable, web-based access for your data processing needs.

What’s New – Data Sources

In SAS Viya, you can access a variety of data sources. In addition to SAS tables, you can access data that is on-premises or in the cloud. Data can be in relational databases or unstructured data. Viya can also use other familiar file formats such as XML, JSON, CSV, or XLSX.

SAS Viya includes several services and applications. Here are a few of these applications:

- SAS Drive for organizing, accessing all SAS content
- SAS Visual Analytics for web-based reporting and dashboards
- SAS Visual Data Mining and Machine Learning
- SAS Studio for writing and submitting SAS code
SAS Viya includes multiple servers to execute SAS code. The two primary servers are the SAS Compute Server and SAS Cloud Analytic Services, or CAS. Consider the SAS Compute Server as equivalent to the SAS®9 workspace server. Your familiar SAS®9 programs can be submitted as is to SAS Viya, and, by default, they'll run on the Compute Server. There's no need to learn new syntax to use the Compute Server. SAS Cloud Analytic Services is the high-performance server that performs parallel processing on in-memory data.

This program will run on a COMPUTE server with no changes whatsoever. Will generate reports, all those tasks you were able to do before, still going to perform in VIYA.

There are different ways to submit requests to CAS for in-memory parallel processing:
1. One way is to use fairly familiar syntax with data steps and proc steps, but we call them, CAS-enabled steps that can actually go to CAS and process that in memory data.
2. The other alternative to execute code in CAS is to actually speak CAS’s native language, which is what we call CASL. It looks a little different than our standard SAS syntax.
1.2 What Is CAS

This program is written to execute in CAS in order to take advantage of parallel processing of in-memory data in the DATA step and PROC MEANS. You will notice that the basic syntax is familiar. However, there are some new statements and steps used to direct processing in CAS.

Let’s take a high-level look at how CAS works. When a program is submitted, it first goes to the CAS controller. The controller distributes the code and data to separate worker nodes. The nodes perform coordinated parallel processing on their portion of the data. This means that the nodes are executing the same action at the same time.

So essentially, we’re distributing the data & the work, the processes & then having the results come back from each node & basically collating it all back together.
1.3 Connect To CAS

Let’s Start with What We Know
SAS Libraries

LIBNAMElibref engine "path";

libname mysas '/workshop/How/Shankar_5332/data';

mysas

Libraries are also used in SAS Cloud Analytics Services, but they are called CAS libraries. The CAS library, or caslib, is the mechanism by which data is accessed in the SAS Viya environment. At its simplest, a caslib is a container that has two main areas:

1. An In-Memory Space where the processing takes place, and a Data Source.
2. The data source might include SAS data sets, csv files, databases, and sashdat files.

Let start with what we know about libraries. As a SAS programmer, you’re familiar with SAS libraries and the LIBNAME statement. You might have a folder or directory on your local machine that contains various file types including SAS data sets, csv files, Microsoft Excel files, and more. In order to access the SAS data sets, you need to associate a SAS library with the folder.
2.1 Session Scope Vs. Global Scope

In-memory tables can have either session or global scope. So what’s the difference between a session-scope table and a global-scope table? In-memory tables have session scope by default. A session-scope table is only accessible in the CAS session where it was created, and it's only visible to the user who created it.

A session-scope in-memory table only exists for the duration of the session. When the CAS session ends, the table is dropped.

But what if you need to share data across your sessions, or with other users? You must create a global-scope table if you need to share data. A global-scope table is also called a promoted table. You can promote a table when you load a file into memory or promote an in-memory session table.

Files of any kind that are not mapped to the caslib are called “client-side” files. You load files into the in-memory space in CAS using your client software – in our case, we will use SAS and PROC CASUTIL.

You've started a CAS session and accessed a CAS library. CAS can only process data in its in-memory space, so your next step is to load your file into memory. Source data files mapped to a caslib are referred to as “server-side” files. These files can be rapidly loaded into the caslib’s in-memory space for processing.

To load a server-side file that’s stored in the caslib's data source, use the CASUTIL procedure LOAD statement with the CASDATA= option.
SECTION 3 MODIFYING SAS PROGRAMS TO RUN IN SAS VIYA

3.1 Running a SAS Data Step in Base SAS vs. SAS Viya

When a DATA step is executed in Base SAS, it runs in a single-thread on the SAS work space server. Processing data in a single thread reads data sequentially, one row at a time.

SAS Viya enables data to be divided and processed simultaneously on multiple-threads. When a data step executes in CAS, each thread executes the program statements on its data, and returns the results to the controller.
CAS and the Compute Server each have their unique strengths. It depends on the size of your files and the nature of your code as to which steps leverage those strengths. It might be most efficient for your code to execute with a combination of the two servers.

There's a certain amount of overhead processing that occurs when code runs in CAS.

As you work in your Viya environment and consider writing or modifying programs to run in CAS, there are a few best practices to consider.

- CAS generally outperforms the Compute Server if you have data sources larger than 50 GB,
- steps that run for 30 minutes or longer,
- PROCs that are computationally demanding, or DATA steps with extremely long or complex logic.
CONCLUSION
This paper attempted to showcase the power of Viya and CAS from assigning libraries, to moving data and manipulating it. Performance benefits were highlighted so that the reader weighing options can perhaps begin to consider Viya for their daily data work.

ACKNOWLEDGEMENTS
The author is grateful to the many SAS users that have entered her life. Charu is grateful to the Pharmasug Conference, 2022 for the opportunity to present this paper. She would also like to express her gratitude to her manager, Stephen Keelan without whose support and permission, this paper would not be possible.

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

Charu Shankar
SAS Institute Canada, Inc.
Charu.shankar@sas.com
https://blogs.sas.com/content/author/charushankar/

REFERENCES
SAS Viya Programming

SAS Viya Learning Path
https://support.sas.com/training/us/paths/prg.html

Data Step Processing in CAS
https://go.documentation.sas.com/doc/en/pgmsascdc/v_018/casdspgm/p0ujjmynr82tfsn1ppyp475bhvaib.htm

SAS Education Resources for Viya

SAS 9 and SAS Viya Functional Comparison

SAS Global Forum Tutorial - Coding in SAS Viya
https://www.youtube.com/watch?v=Beeh2HjWqWA

Seriously Serial or Perfectly Parallel Data Transfer with SAS Viya

CAS Concepts
https://go.documentation.sas.com/?docsetId=calserverscas&docsetTarget=n05000viyaservers000000admin.htm&docsetVersion=n=3.4&locale=en

An intro to Viya Programming
https://go.documentation.sas.com/api/docssets/pgmdiff/3.4/content/pgmdiff.pdf

Differences in the SAS® 9 and SAS® Viya™ 3.1 Platforms
https://go.documentation.sas.com/api/docssets/whatsdiff/3.1/content/whatsdiff.pdf?locale=en#nameddest=n0evbd1ha0clqvn1sbz5yag06xi6
SPRE (SAS Programming Runtime Environment)

SAS® Cloud Analytic Services 3.1: Language Reference
https://go.documentation.sas.com/api/docsets/casref/3.1/content/casref.pdf?locale=en#nameddest=p05ccny5glgvwan19mkisxl8z1jk