Paper 94-25

Working Efficiently in a Global Environment: Client/Server Processing with SAS/CONNECT® Software

Robert W. Graebner, PRA International, Inc., Lenexa, KS

ABSTRACT

Working efficiently in a global environment requires the ability to process data from a variety of locations and platforms. SAS/SCONNECT provides three ways to accomplish this, data transfer services, remote library services and compute services. This paper illustrates how compute services can be used to efficiently process data in a global client/server environment. This paper is intended for users with a basic knowledge of SAS.

INTRODUCTION

The PRA global network consists of similar SAS installations at each Trial Management Center (TMC) which are interconnected via a wide area network (WAN) as illustrated in Figure 1.

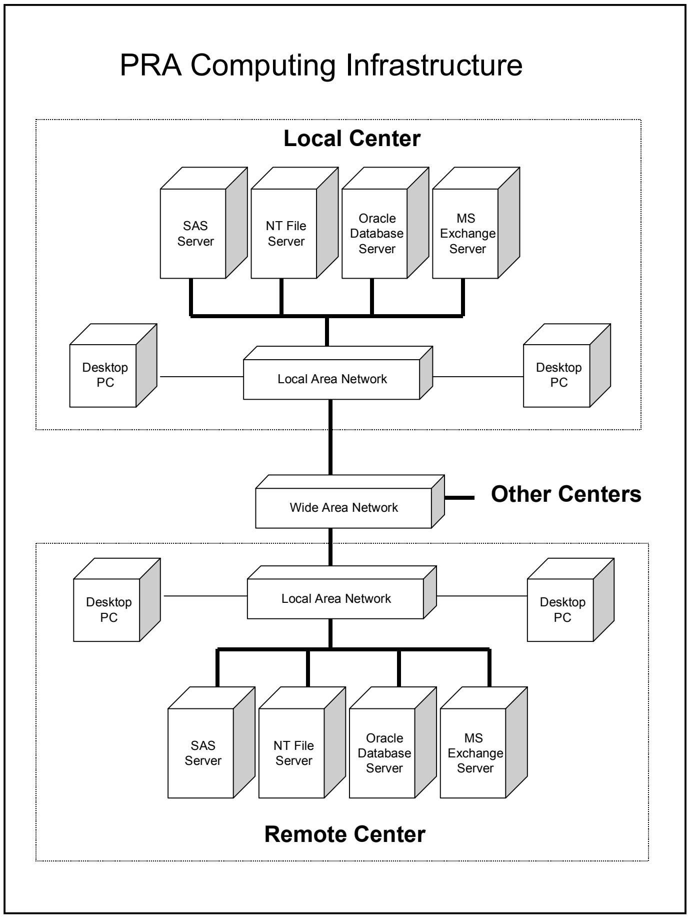

Figure 1: PRA Computing Infrastructure

Each TMC has a Windows NT server setup as a dedicated SAS server. The SAS server is connected to a Windows NT file server and an Oracle database server. This configuration allows SAS jobs to be submitted from any SAS workstation to the SAS server at any TMC. Once a job has been submitted, a new SAS session is initiated on the server for that job. When the job is complete, the output and log are sent back to the workstation or to a file. SAS jobs may also be run on the workstation, but there are several advantages to running them on the server.

ADVANTAGES OF USING SAS CLIENT/SERVER

- **Increased Processing Speed** – SAS programs are run on a dedicated SAS server with dual high-speed Pentium processors and an internal scratch disk for temporary data sets, sorting, etc.

- **Reduced Network Traffic** - When a program is submitted to a server, only the program goes over he network. The program runs on the server and data only needs to travel back and forth from the file server or Oracle database server. When the program has completed, the Output and Log are sent to your workstation. When a program is run on a workstation that accesses data sets on the file server, all of the data must move across the network and back again if permanent data sets are created.

- **Ability to Run Jobs at Any TMC** – Users can submit SAS programs to any of PRA’s SAS servers when they need access to data at that TMC. Due to greatly reduced network traffic, programs will execute more quickly and no data transfer will be necessary.

- **Ability to Run Jobs on Your Workstation** – Users can test new ideas using non-production data on their PC workstation which makes testing quick and easy.

- **Ability to Access Data Sets on Different Platforms** – SAS data sets which reside on other platforms accessible by the PRA WAN can be accessed or copied without using PROC CPORT or PROC COPY.

- **Ability to Use SAS Products Installed on the SAS Server** – When programs are run on a SAS server, they can use procedures and components not installed on the workstation.

The combination of increased processing speed and reduced network traffic obtained when SAS jobs are run on the SAS server can dramatically reduce the time required for jobs to complete.

The flexibility of the SAS client/server system allows many options as to where the data will be stored and where the processing will be done. The following three are the most useful:

1. **Processing and Data on Desktop** - Mainly for testing
2. **Processing on the Desktop with Data on Server** - Alternative production method
3. **Processing on SAS Server with Data on Server** – Preferred for production work

For production work, the third method is preferred because it is the fastest and has the least impact on network traffic. Method number three is the only practical method to use when working remotely with a laptop PC due to the slow data transfer rate. Method number two is available as an alternative, however it can be slow because all of the data must travel over the network from the file server to the workstation for processing and results or permanent data sets must travel back across the network. Method two should never be used with jobs that access large data sets. Method number one is fast and no network traffic is involved but since production data sets are not stored on workstation hard drives it is limited to testing. Each of these methods is described below.

To maximize performance and minimize network traffic, you should submit your production jobs to the SAS server located in the same region as the data you are accessing. Processing your job on another server or on your desktop can result in large amounts of data being transferred over the network, slowing down both your job and the work of others.

PROCESSING ON SAS SERVER WITH DATA ON SERVER

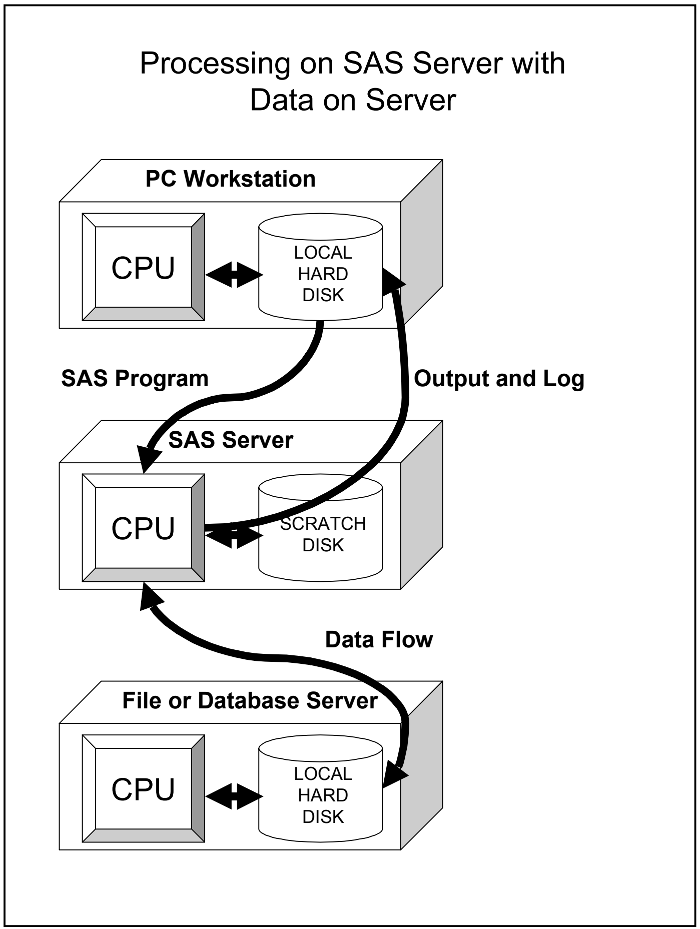

Figure 2: Processing and Data on Server

SUBMITTING A JOB TO A REMOTE SAS SERVER

There are two ways to submit a job to a SAS server:

1. Use the **Signon**, **Remote Submit** and **Signoff** commands from the **Locals** menu.
2. Include statements in your program to accomplish these three tasks and then submit your program as you would to run it on your workstation.

In most cases the second method is preferred. With this method the program is self-contained, you only need to define the connection parameters once, when you write the program, rather than every time you run it, and the program statements document where it will be run.

SUBMITTING REMOTE JOBS WITH INTERNAL COMMANDS

Statements within the program can provide the information necessary to sign on, remote submit and signoff. When this method is used, you submit you job as you would a local job to be run on your workstation. When the statements are executed, they make the connection to the SAS server, which starts a new instance of SAS on the server, which will process the commands within a resubmit block. The following sections contain sample programs to illustrate the commands and their syntax.

BASIC REMOTE SUBMISSION

The following program gives an example of the basic commands necessary to submit a job to one of the PRA SAS servers.

```
① filename rlink
'\\kcisas\sasscr\tcpwin.scr';
② options comamid=tcp remote=kcisas;
③ signon;
④ rsubmit;
⑤    libname test
'\\kcifile\projects\SAS Client Server
Testing';
⑥    proc freq data=test.demog;
          tables name;
     run;
⑦ endrsubmit;
⑧ signoff;
```

PROGRAM DETAILS

① A fileref must be defined which points to the tcpwin script file that is to be run when establishing or breaking a link to the SAS server. When the SIGNON and SIGNOFF statements are used without a parameter, the default fileref of RLINK is used. The fileref RLINK must be associated with the tcpwin script file on the SAS server you are submitting your job to. The script files are created and maintained by the corporate IT staff and they will be in the same location on each SAS server. The only difference will be the NT file server specification of XXXFILE, where XXX is the three-letter region identifier.

② An OPTIONS statement that specifies which protocol to use must be present. In this case COMAMID=TCP is used to specify that the TCP protocol is to be used. The REMOTE=*serverid* option is optional. It allows you to define the remote server ID once rather than having to specify it in each SIGNON, SIGNOFF and RSUBMIT command.

③ The SIGNON statement establishes connection to the remote SAS server. In this case the REMOTE= option has been used, so no server ID is necessary. If this were not the case, then a valid SAS server ID would need to be added as part of the SIGNON statement (e.g. SIGNON LHRSAS).

④ The RSUBMIT statement begins a block of SAS programming statements that will be sent to the specified SAS server for execution. The ENDRSUBMIT statement ends this block. All statements outside of this block will be executed on your workstation. Because a REMOTE= option has been set, the SAS server ID does not have to be specified.

⑤ The statements in the RSUBMIT block will be executed on the SAS server in a separate instance of SAS. None of the specifications on your workstation instance of SAS (such as filerefs) will be in effect in this instance. Any that are required by the statements within the RSUBMIT block must be included within that block.

⑥ Procedures run on the remote SAS server will return their output and log files to your workstation in the output and log windows. If you submitted your job in batch mode, they will go to files in the directory where the program is located, with the same name as the program and extensions of .LST and .LOG. PROC PRINTTO can be used to reroute or rename the output and log files if necessary.

⑦ The ENDRSUBMIT statement is necessary to end the RSUBMIT block. Any statements after this will be processed on your workstation.

⑧ The SIGNOFF statement releases the connection to the remote SAS server. If the server ID was specified in a REMOTE= option, then you do not need to specify it here.

REMOTE SUBMISSION USING TWO SERVERS FOR ONE JOB

This program illustrates how to connect to more than one SAS server and then submit some commands to one and some to another. Because you establish a connection to more than one server, the rsubmit commands must include the name of the server.

```
options comamid=tcp;
filename rlink
'\\kcisas\sasscr\tcpwin.scr';
signon kcisas;
filename rlink
'\\rbosas\sasscr\tcpwin.scr';
signon rbosas;

rsubmit kcisas;

    libname kcitest
'\\kcifile\projects\SAS Client Server
Testing';

    proc freq data=kcitest.smdemog;
        title 'Results from KCI';
        tables name;
    run;

endrsubmit;

rsubmit rbosas;

    libname rbotest
'\\rbofile\training\RBO\SAS test data';

    proc freq data=rbotest.smdemog;
        title 'Results from RBO';
        tables name;
    run;

endrsubmit;

signoff rbosas;

filename rlink
'\\kcisas\sasscr\tcpwin.scr';
signoff kcisas;
```

ACKNOWLEDGMENTS

SAS is a registered trademark or trademark of SAS Institute, Inc. in the USA and other countries. ® Indicates USA registration.

CONTACTING THE AUTHOR

Robert Graebner
PRA International, Inc.
16400 College Blvd.
Lenexa, KS 66219

graebnerbob@praintl.com
graetech@grapevine.net