

Performance Testing for SAS Running on AIX and Windows NT Platforms

Paul Gilbert & Steve Light at DataCeutics, Inc
 Andy Siegel & Shylendra Kumar at Astra USA

Abstract

At Astra USA, Statistical Programmers and Statisticians use the SAS System to produce datasets, listings, tables, graphics and analyses which are the basis for a Clinical Trial Report. SAS software is configured in a distributed UNIX environment using IBM AIX workstations and an IBM RS/6000 mainframe. SAS users have often experienced significant performance degradation, seemingly during hours of peak network and RS/6000 system use. Performance testing is conducted to document and identify the source of AIX performance degradation, and to investigate analagous Windows NT client/server configurations. A secondary purpose is to obtain performance data for several different Windows NT client/server configurations in preparation for selection of a PC-based configuration for running SAS/PH-Clinical. Three AIX configurations and five Windows PC client/Windows NT server configurations are evaluated.

Scope

The scope of testing is twofold. First, to obtain performance information using three AIX configurations of SAS software and SAS data library storage. Second, to obtain the performance information on five configurations running on a Windows PC client and Windows NT server platform.

AIX Configurations:

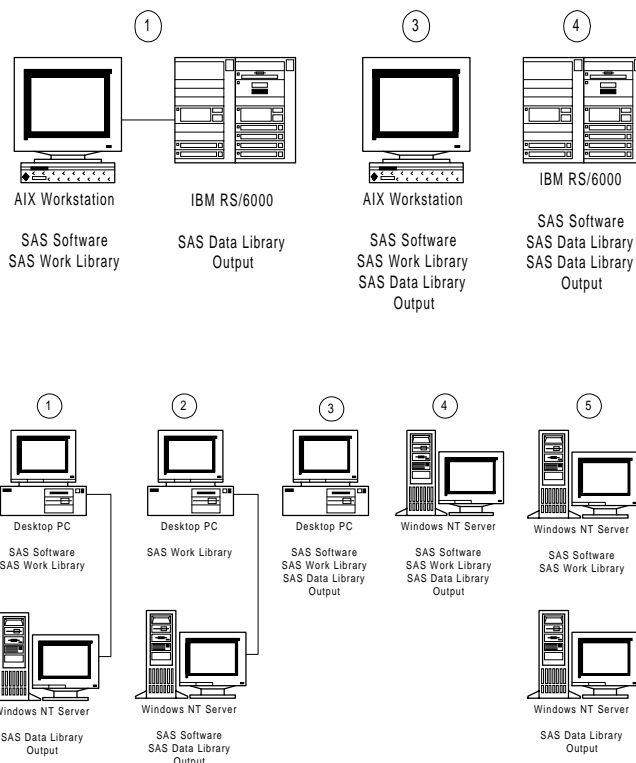
In the first AIX configuration (1) the SAS software and SAS work library are located on the users' AIX workstation; and the SAS data libraries and output are located on a network disk drive. The second configuration (3) is standalone with all components located on the users' AIX workstation. The third configuration (4) is also standalone with all components located on the RS/6000 server. The base network configuration (routed) will have the client located on different network ring than the server. A second network configuration will be tested using configuration 1, where the client and server are located on the same network ring (local). This will give us performance measures using a single ring network link, as well as, a routed network link.

Windows Configurations:

The first Windows configuration (1) is distributed, with the SAS software and SAS work library located on the users' desktop PC and the SAS data libraries and output located on a network disk drive. The second configuration (2) is distributed, with the SAS work library located on the users' desktop PC with all other components located on

server. The third configuration (3) is standalone, with all components located on the users' desktop PC. The fourth configuration (4) is also standalone, with all components located on the Windows NT server. The fifth configuration (5) is distributed, with the SAS software and SAS work library located on the a Windows NT server and the SAS data libraries and output located on a second networked Windows NT server disk drive. Two Windows clients will be tested: Windows 95 and Windows NT Workstation. Only configurations 1 and 3 will be tested using the Windows 95 client.

*AIX Configurations
 Windows Configurations*



The base network configuration (local) will have the client and server located on the same network ring. A second network configuration (routed) will be tested using configurations 1 and 2. This will give us performance measures using a non-routed network link and network link using routers.

Performance results data will be stored in permanent SAS datasets where it can be used for analysis. Graphical comparisons of the averaged results for all configurations

will be generated from the test data.

Table 1, on page 5 details test platform and configuration specifics.

Hardware Specifications

This section contains the specification for the typical workstations, network and servers. The standard AIX workstation is an IBM RS/6000 model 355 with 64MB memory and a 800MB SCSI disk drive. The standard PC workstation is an IBM PS/2 Pentium with 32MB memory and a 1.2GB IDE disk drive. The network is an IBM 16Mb/sec Token Ring. Routers are used to link multiple rings. The AIX servers are IBM RS/6000 models 370 and R21 with at least 128MB memory and at least 1GB SCSI disk drives. The Windows NT server is an IBM PC Server 704 (2 166Mhz Pentium Pros) with 132MB memory and 10- 2GB SCSI disk drives using software RAID level 0.

Software Tools

SAS Software tools will be used to perform the information gathering and storage. We are using SAS Windows version 6.11 TS40 and AIX version 6.11. The first tool is a SAS test program which will be used to produce the performance information. This test program contains 10 program modules used for creating datasets and output for a completed Clinical Trial Report. The ten program modules include:

- two analysis dataset creation programs
- two listing programs
- two summary programs containing report macros
- two graphics programs
- two statistical analysis programs

This cross section of typical programs will simulate normal SAS activity and use. The name of the program is ptp_test.sas and it will be stored in the PTP_TEST development programming directory.

The second tool is a SAS dataset to store performance information. The performance dataset will store program execution information about each of the ten code blocks within the test program, for each run of the test program. The information is described below.

<u>variable</u>	<u>Description</u>
platform	platform type (aix, nt, w95)
config	configuration (1, 1x, 2, 2x, 3, 4, 5)
type	test type* (u, u4, n, n4, s, s4) denote unstressed, normal or stressed
run_id	run identification number, a random number used to identify a program run
sequence	sequence number to identify a code block within each run (0, 1, 2, ...)
s_date	block start date
s_time	block start time
e_date	block end date
e_time	block end time
time_dif	execution time (sec) used by each code block

The third tool is a SAS macro to gather performance information and populate the performance dataset. The macro will be placed at the beginning of the test program and after each of the ten code blocks. This macro stores program execution timing information about each of the ten code blocks within the test program, for each run of the test program. The name of the macro is %ptptest and it will be stored in the general macro library. The macro is described below.

```
%ptptest( funct=, plat=, config=, type=, seq= )
```

where:

- funct= (init or write) to initialize program or write information to performance dataset
- plat= (aix, nt, w95) platform type
- config= (1, 1x, 2, 2x, 3, 4, 5) configuration type
- type= (u, u4, n, n4, s, s4) test type of unstressed, normal or stressed
- seq= (0, 1, 2, ...) sequence number to identify code block

The macro functionality is described below. The macro is called at the beginning of the test program with the “initialize” function. Afterward it is called with the “write” function at the end of each of the ten code blocks.

*See description of type in the Testing Procedure section.

Performance Dataset Description

Initialization call at beginning of program

```
%ptptest ( funct=init, platform=aix, config=1, type=u,  
          seq=0);
```

- 1) Create global macro variables for platform, configuration, type, run identification, start date and start time.
- 2) Create a six digit random number to populate run identification macro variable.
- 3) Populate the start date and start time macro variables.

Write call at the end of each block

```
%ptptest( funct=write, seq=1);
```

- 1) Create the work.ptp_test dataset.
- 2) Use the global macro variables for platform, configuration, type, run identification, start date and start time to populate the dataset variables.
- 3) Use the sequence macro variable to populate the sequence dataset variable. The value of sequence is 1 for block 1, 2 for block 2,
- 4) Create and populate end date and end time dataset variables. Subtract the (start date + start time) from (stop date + stop time) to calculate the time difference for the code block execution time. Populate the time_dif dataset variable.
- 5) Append the observation to the stored ptp_test dataset.
- 6) Use end date and end time dataset variables reset and populate the start date and start time macro variables.

Testing Procedure

This section describes the testing procedure which will be repeated for the various configurations on both platforms. The purpose of this procedure is to simulate three levels of computer and network utilization. We have defined the levels as unstressed, normal use and stressed. For each test the %ptptest macro parameters need to be defined as appropriate. All tests will be run in batch mode. Each test will be executed several times at all three levels to allow us to average the results for each level.

The first level is unstressed, where system resource utilization is low. This test will give us a baseline for best performance on each configuration. Test will be run after normal working hours. Each test program will be executed five times in sequence (no overlap of execution).

The second level is normal use. Test will be run during

normal working hours. Each test program will be executed five times in sequence (no overlap of execution) on two days for a total of ten test runs. This test is not applicable to the standalone configurations as the network is not being used in a standalone configuration.

The third level is stressed, where system resource utilization is high. Test will be run after normal working hours so it will not effect users. A software tool such as Response Probe or automated file copying will be used to simulate extreme network traffic. Each test program will be executed five times in sequence with no overlap of execution. It will then be run simultaneously on multiple workstations.

The three levels of stress testing will be repeated running the test program four times simultaneously on each of the client rest platforms: AIX, Windows 95 and Windows NT. This will be performed to simulate normal programmer use, where multiple programmers are editing programs and running several others at the same time. The stress levels will be denoted as u4 for unstressed, n4 for normal use and s4 for stressed.

Table 2, on page 5 describes the levels of stress to be run on the various configurations.

Results and Discussion

At the time this paper is being written we have only preliminary results. Complete results will be distributed and discussed at the presentation. The preliminary results include normal stress level benchmarking only. The results are summarized numerically (Table 3) and graphically (Figure 1) on page 6.

Table 3 contains summary statistics describing the total time for each execution of the test program on each configuration. Figure 1 is a bar chart of the mean time for each configuration.

In reviewing the preliminary results, our initial thoughts are:

- 1) The Windows NT standalone and client/server configurations are faster than the corresponding AIX standalone and client/server configurations.
- 2) The network and network topology have an effect on performance, at the most a two fold performance degradation. This effect is not as large as we initially thought it was. Network stress effect is still unknown.
- 3) The Windows 95 configuration results are inconclusive at this time.

Biography

Paul Gilbert is Vice President of DataCeutics, Inc., a consulting group specializing in software solutions, system integration, programming and support in the areas of clinical data management and statistical reporting. His fifteen years' experience includes clinical data management, implementing and maintaining Clintrial, designing SAS based biostatistics reporting systems, managing SAS based NDA programming support, SAS/PH-Clinical implementation and CANDA integration.

Steve Light, a Project Manager at DataCeutics, Inc., is responsible for SAS report macro development and biostatistics/clinical SAS programming support. His twelve years of experience with clinical information systems include SAS systems development and validation, NDA driven project management, SAS/PH-Clinical implementation, programming and clinical data management.

Andy Siegel, the CANDA Project Manager at Astra USA, is responsible for implementing SAS/PH-Clinical as the data review module of the Astra USA CANDA tool. His twelve years of experience with clinical information systems include SAS systems development and management, CANDA project development and management, and programming.

Shylendra Kumar, the LEAD SAS Programmer at Astra USA Inc., is responsible for SAS/PHClinical implementation and programming, and SAS system administration. His six years experience include academic research, data analysis, SAS/PHClinical implementation, programming, network and hardware support for SAS software.



Table 1
Test Platform and Configuration Specifics

Configuration	Client	Server (network location)	SAS software	SAS work library	SAS data library	SAS Output
AIX 1*	AIX workstation	AIX server (routed)	client	client	server	server
AIX 1x	AIX workstation	AIX server (local)	client	client	server	server
AIX 3	AIX workstation	N/A	client	client	client	client
AIX 4	N/A	AIX server	server	server	server	server
NT 1	NT Desktop PC	NT server (local)	client	client	server	server
NT 1x	NT Desktop PC	NT server (routed)	client	client	server	server
NT 2	NT Desktop PC	NT server (local)	server	client	server	server
NT 2x	NT Desktop PC	NT server (routed)	server	client	server	server
NT 3	NT Desktop PC	N/A	client	client	client	client
NT 4	N/A	NT server	server	server	server	server
NT 5	NT server 2	NT server (routed)	server 2	server 2	server	server
W95 1	Win 95 Desktop PC	NT server (local)	client	client	server	server
W95 3	Win 95 Desktop PC	N/A	client	client	client	client

*current standard configuration

Table 2
Levels of Stress to be run on the Various Configurations

Platform & Configuration	Levels of Stress					
	u	n	s	u4	n4	s4
AIX 1	✓	✓	✓	✓	✓	✓
AIX 1x	✓	✓	✓			
AIX 3	✓			✓		
AIX 4	✓			✓		
NT 1	✓	✓	✓	✓	✓	✓
NT 1x	✓	✓	✓			
NT 2	✓	✓	✓	✓	✓	✓
NT 2x	✓	✓	✓			
NT 3	✓			✓		
NT 4	✓			✓		
NT 5	✓		✓	✓		✓
W95 1	✓	✓	✓	✓	✓	✓
W95 3	✓			✓		

Table 3
Mean total execution time by configuration - All configurations
 Overall summary statistics - Total execution times (tests 1-10 combined)

Configuration	Mean \pm STD	Median	Min	Max
AIX1	967 \pm 100	942	863	1174
AIX3	796 \pm 47	772	759	868
NT1	604 \pm 6	605	595	612
NT1X	631 \pm 22	622	613	680
NT2	559 \pm 10	556	549	580
NT2X	709 \pm 20	703	682	740
NT3	464 \pm 12	458	457	486
NT4	265 \pm 6	264	259	277
NT5	552 \pm 6	554	545	557
W95 1	796 \pm 22	794	764	841
W95 3	422 \pm 18	424	394	439

Figure 1

