ABSTRACT

I recently participated in a project to design and performance test a client/server environment that would support statistical processing. The hardware and software mix included Oracle on one or more HP/UNIX servers as the DBMS for the data warehouse, SAS Software for analytic processing, and both the servers and Windows PC's as candidate sites for the computations and data serving. I had hoped to benefit from the experiences of others and do research and test every possible combination of cooperative processing and file serving. But I found only limited information in recent SIGO proceedings. Much has been written at this point, but these papers tend to be theoretical, describing what can now be done and generally how to do it. Little seems to be available that gives you some sense, with hard numbers, of how capable these environments are of supporting the kinds of tasks we hope to assign them to. So I felt I could contribute to the SAS community by publishing some of my own project-related research findings. This paper should be useful to SAS programmers considering a client/server system that combines access to large data files and heavy computational processing.

INTRODUCTION

My project concerns migrating a statistical processing operation from a legacy mainframe environment to a client/server environment which is supporting several related applications. These applications are all part of a new integrated clinical trials system, called CRISP, being developed by Merck Research Laboratories. The CRISP project was initiated by Merck in response to the challenge of handling large and complex amounts of clinical trials data for global registration of new drug compounds. The server side of the new platform involves HP9000 Series 800 servers running the HP-UX10 operating system, Oracle database software, and SAS Software. The client side involves high-end desktop PC's running Windows for Workgroups 3.11 and SAS Software served from a Windows NT applications server. There are several Oracle databases involved which together represent the data warehouse for all of the applications.

We recognized up-front that in a client/server environment, and especially one which involves SAS Software which can run on both the client and the server sides, that there would be several ways available to accomplish the variety of processing tasks at hand. Further, we understood that our statisticians are accomplished SAS programmers and that they would be free to use all of the ways available. Hence, we felt it was not realistic to choose one particular way for our users to accomplish their work. Personal preference and particulars of the moment would dictate the ultimate choice by given statisticians. So, our goal for the performance testing was to examine all the feasible processing configurations and:

- determine if the conceived hardware/software platform was adequate overall;
- see how this platform compared to the mainframe;
- offer guidelines on what configurations among the several on this platform would provide the best results for the processing tasks at hand.

This paper describes our experience designing and testing our new processing environment. In it, I will:

- describe the configurations identified and tested;
- present and discuss the test results;
- provide a series of guidelines to help one design and test a client/server environment.

CLIENT/SERVER ENVIRONMENT

Server Hardware and Software

The data is warehoused on the largest server in the mix, a HP9000, Series 800, Model T500 running the HP-UX version 9 operating system. This machine, which I will refer to as the data warehouse server, is nearly a mainframe in processing capabilities. The data resides in several Oracle databases on this machine. The Oracle release is 7.2.2. A second HP/UNIX server, relatively smaller but by no means a slouch, is dedicated to the statisticians. This statistical server is a HP9000, Series 800, Model K200 running the HP-UX version 10 operating system. It has two 100 MHz CPU's, 512 MB...
RAM, fast/wide SCSI drives and a floating point processor. Version 6.09 of the SAS System for UNIX is running on this machine. The SAS/ACCESS to Oracle product allows SAS programs to extract data from Oracle, and SQLNET is the "middleware" forging the connection from SAS on the K200 statistical server to Oracle on the T500 data warehouse server. SAS/CONNECT is also running on the K200 to make the connection between SAS on this statistical server and SAS for Windows on the client PC's.

When our performance testing project began, the T500 and the K200 were not yet available. We did, however, have available a HP9000, Series 800, Model G50 on which to warehouse the data under Oracle, and a HP9000, Series 800, Model G40 for the statistical processing. The G40 runs the HP-UX version 9 operating system. It has one 64 MHz CPU, 128 MB RAM, and fast/wide SCSI drives. It does not have a floating point processor. This is equivalent to a PC without a math co-processor. The G50 server has one 96 MHz CPU and 256MB of RAM.

We felt it made complete sense to begin testing with only the G50 and G40 available at the server end for a few reasons. First, much of the preparation we would do on the G50 and G40 could be easily transferred to the T500 and K200 when they were available, so we were heading directly for our ultimate goal from the start. Second, some of the results could be scaled up to the larger machines before we even had them. Third, much of the testing would take place largely on the client end and could be done without distorting the results if the G50 and G40 played the relatively smaller server roles. I feel there is an important message here for anyone planning a client/server performance testing project. That is, there is much productive work that can be done prior to the arrival of the production hardware and software. This is an especially useful point because hardware in particular is often delivered later than we would hope, but its delay does not have to bring your project to a halt.

One more note about the K200 server (and its surrogate G40) is in order. We began our testing where multiple instances of a program being run simultaneously shared the same physical disk volume (i.e., disk spindle) and the same disk controller. Performance was quite poor, so we separated our multiple programs onto separate physical disk volumes which used separate controllers, and saw marked improvement. We decided from these results that in production we would distribute the processing for our drug projects across controllers and disk volumes. But we did test both ways, so I report the results of both ways later in the paper. And note that wherever I use the phrase "separate physical disks" I mean that these disks are also on separate controllers.

Client Hardware and Software

The PC's used for testing are Compaq PC's with Pentium 75 MHz and 100 MHz CPU's. These machines run the Windows for Workgroups 3.11 software over DOS 6.2. They have 700MB medium speed IDE hard drives, and 24-64MB RAM. The SAS System for Windows, version 6.10, is available both resident on the local hard drive and, in served mode, from an NT server.

Interestingly, the PC's CPU speed did not turn out to make much difference in performance when our test SAS programs ran on the PC's. This is largely because our programs are very I/O intensive and I/O was always the bottleneck, not CPU cycles. Thus, I will not distinguish between CPU speeds on the PC's as I discuss the results below.

Also, since our testing found very little performance difference between the network-served and local-served approaches to serving the SAS Software, I will not distinguish between these two approaches as I discuss test results below. I am not suggesting here that it is never important where SAS is served from. It may be in your case, and if so, it should be tested. However, in our case, the small time savings from local service was of little difference in the overall picture.

Finally, we found that performance peaked at 32 MB RAM when testing our programs on PC's in the "All PC (Standalone)" configuration. Thus, we decided to simply use the 32MB RAM machines as the PC clients when we conducted the client/server tests. Therefore this sized PC was used for all the test results below.

Network Connections

The statisticians operate out of two sites. The T500 and K200 servers reside at the same location, which is yet a third site. The WAN connecting the sites is varied. The two servers are both connected to one statisticians' site using a T3 line, which operates at 45 megabits per second. And they are connected to the other statisticians' site using a FDDI ring transmitting at 100 megabits per second. In addition, the two servers are connected to each other at their local site on another FDDI ring. Finally, the clients are on LANs that
connect to the WAN and that locally use "10BaseT" technology, which runs at 10 megabits per second.

It is important to note that we tested our client/server processing on the K200 statistical server from both statisticians’ sites and found no performance difference whether we were using the T3 line from one site or the FDDI ring from the other site. Hence, I will not distinguish this aspect of the testing any further as I present the results below.

Finally, recall that the model G50 server was the surrogate for the T500 data warehouse server during initial testing (and eventually all testing as discussed later), and the G40 server was the surrogate for the K200 statistical server during early testing. Hence, I note that the G40 statistical server was connected to the clients on a T1 line (at 1.4 megabits per second), and it was connected to the G50 data warehouse server on the same line.

I have included all these details about the network connections to be complete in my presentation. However, I do not mean to suggest by the amount of time spent on this information that the speed of these connections was critical to the performance results. In fact, I believe that the actual client/server configurations tested was more important. That is, the relationship of where the data was to where the number crunching occurred was paramount. And the disk and CPU capacity of the machines in question was also paramount. Any reasonable industry standard network connection speeds could not have the same impact on performance that these other factors did.

CONFIGURATIONS TESTED

As previously stated, there are several processing configurations which can be utilized in the new environment for statistical processing. All feasible configurations were considered prior to our performance testing. The configurations that were actually tested are listed below:

#1 -- Mainframe

This very large IBM mainframe environment is the existing production configuration, or the "legacy" environment. In this environment, the data is warehoused in an old mainframe-based database product. The SAS System, version 6.07 also resides on this machine, with no direct connection from SAS to the database. Data is extracted from the database into external files (EBCDIC) and then loaded into SAS data sets. This extraction occurs before the statistician get involved in the process. Statisticians then analyze the data by running SAS on the mainframe, using their PC’s as simple terminals to the mainframe with the help of Extra terminal emulation software. Note that multiple SAS programs can be submitted to run simultaneously here, since this is a multi-tasking environment.

#2 -- All UNIX Server

The K200 statistical server is used for all the SAS processing and it is the source for the SAS data sets, too. Data is first extracted from the T500 data warehouse server to the K200 using the SAS/ACCESS to Oracle product. The data is then analyzed using SAS programs that are submitted from the K200 to run on the K200. There is no network traffic involved in this configuration when the analysis programs run (except for the eXceed software for terminal emulation). That is, there is no client per se in a separate processing environment that is being fed results. Two other points are important. First, since this is a multi-tasking environment, the user can submit multiple SAS programs to run simultaneously. Second, since this a shared environment, any data stored here can be easily shared among multiple users without any additional effort to transfer it to a place others can get it from, as would be the case if it were stored locally on a PC.

#3 -- PC Client/UNIX File Server

The Windows-based PC is the platform where the SAS analytic processing occurs, and the K200 statistical server in this case is only serving data to be processed on the PC. Data is first extracted from the T500 data warehouse server to the K200 using the SAS/ACCESS to Oracle product. The data is then analyzed using SAS programs that are submitted on the PC to run on the PC. SAS/CONNECT and SAS Remote Library Services facilitate the data serving from the K200 to the PC. This is the only configuration that engenders substantial network traffic during statistical program execution. Two other points are important. First, since this is a single-programming environment, the user cannot submit multiple SAS programs to run simultaneously. Second, since this not a shared environment, any data stored here cannot be easily shared among multiple users. Additional effort is needed to transfer it to a place others can get it from.

#4 -- All PC (Standalone)
The Windows-based PC is used for all the SAS processing and is the source for the SAS data sets, too. Data is first extracted from the T500 data warehouse server to the K200 statistical server using the SAS/ACCESS to Oracle product. The SAS data sets are then downloaded to the PC from the K200 statistical server in a separate step. This is done either using SAS/CONNECT software and SAS proc download, or using SAS proc cport on the server to construct transport files and proc cimport to reconstruct SAS System files on the PC, with a third-party file transfer product used in between to bring the transport files from the server to the PC. The data is then analyzed using SAS programs that are submitted on the PC to run on the PC and input the data from the PC. Network traffic is thus engendered in one large, up-front burst to download the data, and no network traffic is involved during statistical program execution. Two other points are important. First, since this is a single-programming environment, the user cannot submit multiple SAS programs to run simultaneously. Second, since this not a shared environment, any data stored here cannot be easily shared among multiple users. Additional effort is needed to transfer it to a place others can get it from.

The Configuration Not Tested

Our identification of candidate configurations also included research into distributed, or true, client/server processing where all front-end work is done on the client and all job, or transaction, processing is done on the server. The results only are served back to the client when the work is done. With the SAS product, however, when jobs are submitted to the server using SAS/CONNECT and the SAS Remote Submit facility, the client is “locked up” from any further SAS processing until all the job’s results are returned to the LOG and OUTPUT windows. Thus, this configuration does not allow the user to process multiple SAS programs simultaneously.

This configuration, which we might title the “PC Client/UNIX Processing Server” configuration, is most closely associated with our “All UNIX Server” configuration in that the SAS programs execute on the K200 UNIX server and these programs input data from and output data to the same UNIX server. The first important difference, though, is that in the “All UNIX Server” configuration the results (in terms of LOG and OUTPUT window contents) are served to SAS for UNIX windows that are part of the UNIX server environment. In contrast, in the “PC Client/UNIX Processing Server” configuration, the SAS programs are actually remote submitted from the SAS for Windows environment to the SAS for UNIX environment, and the results are served back to LOG and OUTPUT windows of SAS for Windows environment on the PC. This means that when the contents of the LOG and OUTPUT windows are large, as is sometimes the case in SAS processing, more network traffic is engendered this way, which can slow down processing. The second important difference is that multiple SAS programs can be executed simultaneously in the “All UNIX Server” configuration, while this cannot occur in the “PC Client/UNIX Processing Server” configuration.

So, we felt that the “PC Client/UNIX Processing Server” configuration would probably perform almost as fast as the “All UNIX Server” configuration when the results to be served were small, and it would be somewhat slower when the results to be served were large. But it would never out-perform the “All UNIX Server” configuration. This, when coupled with its inability to handle multiple SAS program submissions for simultaneous processing, led us to not pursue it as a feasible configuration during actual testing.

I feel there is another important message here for those considering a client/server performance testing project. That is, you do not have to actually test every one among the many possible configurations. Given your particular circumstances, certain configurations can be eliminated on paper before the testing begins.

STATISTICAL PROGRAMS TESTED

Although the statisticians run a large number of programs during the data analysis, these programs all fall into one of three groups:

- Data preparation Programs;
- Data Analysis programs;
- Oracle to SAS Extract Programs.

We felt it would be best to choose one particularly demanding example from each group for the testing. The program was considered demanding based both upon the amount of data it was reading and writing, and upon the amount of statistical number crunching it was doing with the data between input and output. The programs and data volume tested are described briefly in this section.

Data Preparation Program
This is a long-running, I/O heavy program that prepares the data for analysis. Many temporary data sets are spun off from the original data set input, and some of these data sets are subsequently re-merged with the original data set after their own processing. This is a classic "SAS File Management Processing" type of program which is heavier on SETs, SORTs, and MERGEs than on statistical procedures. The data preparation program tends to be run once, by one statistician, followed by several executions of the analysis program, by several statisticians, using the output data of the data preparation program as the analysis program input data. For our tests, this program input a 48MB data set, and its final output data set written permanently to disk was 43MB.

Analysis Program

This is a shorter-running, somewhat less I/O heavy program than the data preparation program. This program typically has one major data input step, followed by statistical proc oriented processing, and ending in a final table being generated in the OUTPUT window and to an external file. The printed table is not particularly large, perhaps one or two pages total. The analysis program tends to be run many times, by many statisticians, against the same overall data set, varying the parameter settings which control what data subset and what statistical proc options are used. For our tests, this program input a 43MB data set. It did not produce any permanent output data set, and its final printed table was one page in length.

Oracle to SAS Extract Program

As mentioned above, in this environment the data to be analyzed is warehoused in multiple tables of an Oracle database on the data warehouse server. A fundamental processing design choice must be made in such an environment. Will the analysis programs read the data directly from the Oracle database as it is needed, or will the data first be extracted from the database into permanent SAS data sets on the statistical server in a separate step, and then processed by the analysis programs from there? The circumstances of four related issues should determine the answer in most cases.

First, understand that there is considerable overhead incurred when SAS programs read data from Oracle tables versus from SAS data sets. I learned from a review of previous SUGI papers and discussion with SAS Institute technical staff that the time difference can be quite large, even approaching an order of magnitude, when SAS analysis programs read database tables directly. And, of course, the more data being read, the higher the price. So if you are essentially reading all the rows of a database table sequentially in order to analyze all of it, then you will pay the maximum price. You must determine if your analysis programs will be processing the data this way, or if instead they will be selectively retrieving only a (preferably small) subset of the rows.

In order to effectively use the above information, you must consider the second, third, and fourth issues. As the second issue, you must determine how often the analysis programs will be reading the same data. In many analysis situations, the data is re-read many times as the statisticians set and reset statistical and data set options and parameters while they explore the data. In such cases, there is no question that it will much more time efficient to extract the data first and save permanent SAS data sets, unless the third issue points in the other direction.

The third issue concerns how often the data is updated in the database environment, and the price you will pay for potentially working with non-current analytic data and for additional storage, if you extract into permanent SAS data sets. If the database is being updated as the analysis occurs, you would ideally want your analysis programs to process directly from the database and save these costs. However, thinking back to the second issue, the time cost to the statisticians during analysis may be too great, especially in an interactive data analysis situation.

And finally, you must determine where the database resides vis-à-vis where the SAS analytic processing will occur. If it is on another machine and requires a network transfer to move the data to the SAS processing machine, this too might raise the time cost too high to not extract the data first into permanent SAS data sets in a separate step.

I am sure you can see that there is no answer as to which is the best choice here. It depends entirely on a careful analysis of your circumstances. And to make the choice even more difficult, the tradeoffs are sometimes not clear-cut, requiring more analysis and even time tests before the final choice is made. I do recommend that you spend an adequate amount of time, though, on this issue, as the choice you make will be a very fundamental one. It quite possibly will underpin the eventual success or failure of your new processing environment.
In our situation, the choice was to extract first into permanent SAS data sets. And since all the rows of a given database table would be needed for analysis, we wrote a SAS macro which allows the user to input the name of the one or more database tables to be extracted at the same time. The program is centered around PROC SQL, and relies upon the SAS/ACCESS to Oracle product which resides on the statistical server. It uses this product via the "SQL Passthru" approach. This is the program for which results are reported later.

DATA PREPARATION PROGRAM RESULTS

The data preparation program results are presented in Table 1. Note that since this program was tested using both the K200 and the G40 as the statistical server, as discussed above, both sets of results are reported here. The results indicate that the production version (K200 server) of the client/server environment improves performance over the legacy mainframe environment. An important caveat, though, is that each project must be assigned its own physical disk(s) for this performance gain to be realized. Highlights of the Table 1 findings are in bullets below:

- The data preparation program requires about 33 minutes on the mainframe. When the "All UNIX" configuration is run on the G40, this program requires about 62 minutes, which is nearly twice as long. In contrast, the "All UNIX" configuration on the K200 runs in 13 minutes which is well more than twice as fast as on the mainframe.

- The data preparation program requires about 92-108 minutes on the PC, depending upon the configuration tested. And it probably cannot run much faster on this PC without a significant change to the PC's I/O subsystem. Even then, processing time would probably still be measurably slower than the 33 minutes currently obtained on the mainframe and the 13 minutes obtained on the K200.

- When two data preparation programs are run simultaneously on the mainframe, there is no performance degradation over the single-instance execution. Multiple-instance degradation is nearly fully avoided in the "All UNIX" configuration, as long as the users are located on separate physical disk volumes of the HP machine. In contrast, significant degradation does occur on both the K200 and the G40 servers for multiple (two) instances of the program when the users are running on the same physical volume. Processing time more than doubles (from 13 min. to 32 min.) on the K200, and about doubles (from 62 minutes to 121 minutes) on the G40.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Single Instance of Program</th>
<th>Multiple Instances of Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Mainframe</td>
<td>33 min. (avg 7 runs)</td>
<td>no measurable change from single-instance</td>
</tr>
<tr>
<td>2. All UNIX - G40</td>
<td>61.8 min (avg 4 runs)</td>
<td>121 min (avg 2 runs) - same physical disk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>55.5 min (avg 2 runs) - separate physical disks</td>
</tr>
<tr>
<td>2. All UNIX - K200</td>
<td>13.0 min (avg 5 runs)</td>
<td>31.6 min (avg 4 runs) - same physical disk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16.6 min (avg 4 runs) - separate physical disks</td>
</tr>
<tr>
<td>3. PC Client/UNIX File Server</td>
<td>107.7 min</td>
<td>not possible with single Windows session</td>
</tr>
<tr>
<td>4. All PC (Standalone)</td>
<td>92 min (avg 3 runs) + 15 min to download data = 107.6 min</td>
<td>not possible with single Windows session</td>
</tr>
</tbody>
</table>

Table 1
Data Preparation Program Test Results (Single and Multiple Instance Processing)

The main message from these results is that this program generates a lot of I/O, as was already known. The I/O bottleneck problem seen is not surprising, and its sensitivity to the number of physical disk volumes is also not surprising. Simply put, the test results indicate that it is important to spread usage among separate physical disk volumes to maintain good performance. Heavy contention for the same disk volume appears, in fact, to be quite costly to performance. These findings are important information to help best configure the K200 and plan use of it. But I will wait to discuss this further until I have presented the test results from the analysis program, too.
Finally, I note that multiple executions of the data preparation program are not possible when running on a PC with Windows for Workgroups and DOS because the operating system does not support this. Hence, I have no findings to report for the two PC configurations and compare with the "All UNIX" configuration.

ANALYSIS PROGRAM RESULTS

The analysis program results are presented in Table 2. Note that since this program was tested using both the K200 and the G40 as the statistical server, as discussed above, both sets of results are reported here.

The results indicate that no configuration in the client/server environment improves performance over the legacy mainframe environment for analysis program processing, but for single instances of the program the "All UNIX" configuration does perform substantially the same as the mainframe. Processing time for the "All PC (Standalone)" configuration is also not significantly different from the mainframe; however, the PC does not allow for multiple jobs to be run simultaneously, and it requires a large amount of time to pre-download the analysis data from the server to the PC. Multiple jobs can run simultaneously in the "All UNIX" configuration, but there is serious performance degradation as more jobs are added into the mix. In contrast, there is no visible degradation on the mainframe.

At this point it must be stated that the results of this multiple-instance testing reflect the worst-case scenario and may represent conditions more severe than those the statistical users will generally encounter. If the results are indicative of the amount of degradation the statisticians actually do incur during processing on a regular basis, then there are avenues which can be further investigated to increase performance on the K200. Thus, processing volume conditions and performance will have to be closely monitored during production use until this is determined. Highlights of the findings in Table 2 are presented here:

- Processing one instance of the analysis program on the "All UNIX" configuration requires about 1.3 minutes using the K200 and 1.6 minutes using the G40. This compares to 0.6 minutes for the current mainframe environment. This is about a 100% to 150% increase for the one execution of the program, which equates to an actual wall clock time increase of about 40 seconds on the K200 and about 60 seconds on the G40.
- It seems puzzling that the G40 without a floating point processor could run the analysis program in a nearly comparable amount of time to that of the K200 with a floating point processor. We have not yet fully investigated this. But we feel it might be related to the fact that the analysis program also has a sizeable amount of I/O which is a measurable percentage of the total time required for the program to execute. Since the G40 and K200 both have fast-wide SCSI drives to handle this I/O, the G40 may be able to perform close to the K200 when running this particular program.
- The "PC Client/UNIX File Server" configuration requires much longer to run the analysis program (about 10.5 min) because of the time required to move the data over the network to the PC every time the analysis program runs.
- The "All PC" configuration requires only 1.5 minutes to execute the analysis program, which is actually comparable to the UNIX server’s time, but it also incurs an additional 9.5 minutes to pre-download the input data from the UNIX server.
- Multiple instances of the analysis program can run on the existing mainframe without any noticeable degradation from the single-instance time. (Analysis program execution time can vary, though, with other processes running on the mainframe at the same time, but this was not systematically tested for).
- In contrast, there is performance falloff during multiple-instance testing for the "All UNIX" configuration. The degradation starts quite small on the K200 but grows larger with the number of instances. One job requires 1.3 minutes. When the jobs are evenly distributed across two physical disk volumes, two jobs average 1.4 minutes each, 10 jobs average 5.2 minutes each, and 20 jobs average 7.2 minutes each. Degradation is even greater when the jobs are all running on one physical disk.
- Multi-instance processing is not possible on the two configurations that involve the PC because they use the Windows for Workgroups with the DOS operating system, which does not support multiple SAS jobs running simultaneously.
### Analysis Program Test Results

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Single Instance of Program</th>
<th>Multiple Instances of Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Mainframe</td>
<td>0.6 min (avg 10 runs)</td>
<td>no degradation observed</td>
</tr>
<tr>
<td>2. All UNIX - G40</td>
<td>1.6 min (avg 2 runs)</td>
<td>same physical disk:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 by 2 users: not tested</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 by 2 users: 1.4 min (avg 2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 by 1 user: not tested</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 by 1 user: 3.4 min (avg 5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 by 1 user: 5.6 min (avg 15)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 by 2 users: 8.8 min (avg 10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 by 2 users: 13.3 min (avg 20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>separate physical disks:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 by 2 users: 1.7 min (avg 2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 by 2 users: 3.5 min (avg 5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 by 1 user: not tested</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 by 1 user: 3.4 min (avg 5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 by 2 users: 5.1 min (avg 10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 by 2 users: 9.6 min (avg 20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 by 2 users: not tested</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 by 2 users: 2.3 min (avg 8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 by 2 users: 3.4 min (avg 5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 by 2 users: 5.6 min (avg 15)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 by 2 users: 8.8 min (avg 20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 by 2 users: 13.3 min (avg 20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a = staggered start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b = simultaneous start</td>
</tr>
<tr>
<td>3. PC Client &amp; UNIX</td>
<td>10.4 min (estimate with FTP method for the download)</td>
<td>not possible with single Windows session</td>
</tr>
<tr>
<td>File Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. All PC (Standalone)</td>
<td>1.5 min (avg 4 runs) plus 9.5 min to download Analysis data = 11.0 min</td>
<td>not possible with single Windows session</td>
</tr>
</tbody>
</table>

**Table 2**

DATA EXTRACT PROGRAM RESULTS

The Oracle software and source database resided on the HP9000, Series 800, Model G50 server during the testing. This UNIX server machine had one 96 MHz CPU and 256 MB of memory. The SAS software and the destination SAS data sets were on the HP9000, Series 800, Model G40 server. This UNIX server had one 64 MHz CPU and 128 MB of memory. Additional tests of this program were not run on the production K200 statistical server.

The performance results are presented in Table 3. The columns are labeled “Scaled-up” because the numbers were calculated by applying a linear scale-up factor to the actual test results. This was done because the amount of data available in the Oracle database during testing under-represented the data volume we expected in production. Since our early test results indicated that the relationship of data volume extracted to time required was generally linear, we simply applied a linear scale-up factor to reach volumes that were meaningful to us.
In our environment, the rate of extraction for a low volume of data (4.5 MB) was 2MB/minute. It rose to about 5MB/minute for 48MB of data, and leveled off at about 6MB/minute for higher volumes. In our tests, a small volume of data such as 4.5MB required about 2 minutes to extract. Our modest volume of 48MB required about 9 minutes. And our largest volume of 426MB required 69 minutes.

It is clear from these results that the extraction process from database to SAS data sets can be time consuming for large volumes of data. These results will hopefully add meaning to the discussion above in the section titled “Oracle to SAS Extract Program”.

<table>
<thead>
<tr>
<th>Scaled-up Total Size of SAS Data Sets Created</th>
<th>Scaled-up Real Time (Wall Clock)</th>
<th>Scaled-up CPU Time</th>
<th>Transfer Rate in MB/Minute</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6 MB</td>
<td>2.3 min (avg 2 runs)</td>
<td>0.17 min</td>
<td>2.0</td>
</tr>
<tr>
<td>48.6 MB</td>
<td>9.3 min (avg 2 runs)</td>
<td>0.79 min</td>
<td>5.2</td>
</tr>
<tr>
<td>137.9 MB</td>
<td>22.7 min (avg 2 runs)</td>
<td>2.03 min</td>
<td>6.1</td>
</tr>
<tr>
<td>223.3 MB</td>
<td>36.7 min (avg 2 runs)</td>
<td>3.31 min</td>
<td>6.1</td>
</tr>
<tr>
<td>426.1 MB</td>
<td>68.9 min (avg 2 runs)</td>
<td>6.55 min</td>
<td>6.2</td>
</tr>
</tbody>
</table>

Table 3
Oracle Extract Program Test Results

CLIENT/SERVER CONFIGURATION GUIDELINES

One can take a step back from all these results and see a series of guidelines to help anyone who is designing and testing a client/server environment, particularly one that is replacing a legacy mainframe environment. I present my series of guidelines below:

- You can start the project and even the actual testing before all production hardware arrives. First, much time is needed to design the configurations to be tested. Second, time is needed to assemble the programs and data to use during testing. And third, if other hardware is available, much of the work to set up the test environment on that hardware will not be lost as you later transfer to the production hardware.

- There are typically many processing configurations available in a client/server environment, but you do not have to test every conceived one. Some can be eliminated on paper as simply inferior to others. This can save much time.

- Include the legacy system (if there is one) in the testing. The results become more meaningful this way.

- When I/O is a bottleneck on a shared server, as is often the case with SAS processing, distributing the processing across several disk volumes may improve performance immensely. It can also help to keep the SAS Software on yet another volume.

- A powerful server can often outperform a legacy mainframe for single instances of a program; however, this may not be the case for multiple instances. The server may seriously underperform the mainframe in this case. Thus, do not undersize your server by relying on benchmark comparisons between server and mainframe that use only single instance test results. It is very important to factor multi-processing into your test design. (The problem can be mitigated though, by having the multiple instances of the program use separate disk volumes, as mentioned just above).

- If a large amount of your processing is I/O intensive, then offloading the processing to a PC, even a powerful PC, will probably not yield acceptable results; the PC’s CPU capability may be impressive, but even when using a high-end disk subsystem, it probably won’t compete with the performance of a system geared towards heavy I/O processing such as a UNIX server or a mainframe.
Host Systems

- Also, client/server solutions that separate where the data resides from where the number crunching CPU processing occurs, (such as when offloading the processing to a PC and serving the data), will pay a heavy price to move the data to the number crunching machine. This will be true even on networks with reasonably fast transfer rates. This will also be true whether the data is downloaded in a prior, separate step, or dynamically by the number crunching programs.

- So, when processing large amounts of data, (e.g., tens of megabytes worth or more), try to configure a solution which keeps the data close to the number crunching, preferably on the same machine.

- It is hard to fully replicate the actual processing environment while testing; thus, you may not be able to draw final conclusions from your tests. The greatest challenge is in estimating the mix of multiprocessing, and then replicating it during testing. The challenge is even greater when you understand that even when running multiple instance tests, the impact of a literally simultaneous start for the multiple programs versus a slightly staggered start can make a measurable difference.

- Thus, a good idea is to test worst case multi-tasking scenarios, and then make sure you are in an expandable environment, which fortunately a client/server environment usually is.

- Your SAS analysis programs may be processing data warehoused in a database environment. If so, it is usually not immediately clear whether it is better to have the SAS programs extract the data dynamically or extract it in a separate prior step which saves permanent SAS data sets. Nonetheless, this is a very fundamental choice which may have a lot to do with the ultimate success of your chosen client/server configuration. Thus, the circumstances of your analytic processing should be considered carefully, as described in detail earlier in this paper.

REFERENCES


SAS, SAS/CONNECT, and SAS/ACCESS are registered trademarks of SAS Institute, Inc. in the USA and other countries. Windows and Windows NT are trademarks of Microsoft Corporation. ® indicates USA registration.

ACKNOWLEDGMENTS

I wish to thank Mr. Frank Palmieri of the Applications Development group at Merck Research Laboratories for his informed and critical review of my work. Mr. Palmieri is Project Leader for the project reported on herein. I also wish to thank Dr. Philip Loftus, Executive Director of Research Information Systems at Merck Research Laboratories, for his equally critical review. Dr. Loftus has executive directorship over this project. Their ideas and suggestions have been quite valuable to my paper. However, any errors or other shortcomings in the paper remain mine.

For more information about this paper, contact the author:

Barry R. Cohen
Planning Data Systems
PO Box 666
Ardmore, PA 19003
610-649-8701

Mr. Cohen is an independent information systems consultant, specializing in application development, object oriented programming, and client/server solutions. He has been using SAS software since 1980 in a wide variety of industry applications, including a focus on the pharmaceutical industry.