

Paper 287-28

The Bleeding Edge

The Effects of Hardware and Software Migration on the SAS® System

Charlie Bastnagel, Cigna Healthcare, Chattanooga, TN

ABSTRACT

When SAS projects and applications require the latest and greatest, their success or failure is laid open to the whim of technology's "bleeding edge." This quasi-obsessive need for change cuts through our corporate culture, holding little regard for whether that change is necessary or in the corporation's best interest. Knowing how and when to migrate is critical. Research, planning, testing, and benchmarking should be critical tasks in your migration efforts. Migration, by its very nature, exposes an organization to a heavy dose of risk. When you choose the road less traveled and ignore the risk, your project may just end up in a heap on the side of the road as the "bleeding edge" arrives via an OS upgrade, hardware platform migration, or SAS version upgrade. The purpose of this paper is to describe how changes in hardware platforms, software applications, and operating systems can dramatically affect the capabilities, reliability, and performance of the SAS System.

Topics: Hardware Platforms: PC and UNIX. Software: SAS, SAS/GRAPH®, Adobe Distiller, Tivoli Workload Scheduler, AIX Workload Manager, Apache, and Tomcat/Jakarta. Operating systems: AIX and Windows.

DISCLAIMER

The theories, opinions, and conclusions and feeble attempts at parody in this paper are expressed solely as the authors and in no way reflect the attitude of Cigna Healthcare®.

INTRODUCTION

Business and technology are becoming so entwined that at times the lines between them are indistinguishable. We are so consistently barraged with claims of new features and more speed that many customers find themselves in a constant state of migration in order to keep up. Their hand is being forced, but they are afraid to get off the train. The time frame for developing or enhancing a software product and then bringing it to market used to be measured in years and months, but is now measured in months and days. But something has been sacrificed and that something is STABILITY. Detailed design and testing have been glossed over so that vendors may bring their products to market sooner. Technology vendors have figured out that if they can wedge their foot into your door then the battle is won. The problem is that the marketing and the reality of a product's features rarely jive together.

To compensate customers for the down time and instability, vendors have given us Hotfixes. These little miracle workers are often sent in to fix problems that should have been discovered had the vendor done proper testing. Or in

some cases, the vendor knows about the defect, but cannot afford to miss the release date so the problem is sent to you. Hotfixes are at times being misused as a *de facto* method used to stage the release of a new or enhanced software product. But since the Hotfixes sometimes suffer from the same lack of design and testing as the product they are hotfixing, whole new problems are born that require more yet Hotfixes. This endless cycle of change can be symbolized by the Greek figure known as Ouroboros.



What does it mean? The serpent Ouroboros consumes its own tail in an eternal cycle of renewal¹. This is what our technology culture is doing to itself. Change begets change and so on...

Competition in the hardware and software arenas is fierce and it is being played out in advertisements, the courts, the Internet, and in the mailbox outside your home and in your computer. But vendors are not the only ones to blame, for we have become willing spectators in this obsession with change, subsidizing and ensuring it will go on. In the dining rooms of hotels, in conference rooms, and even at the dinner table, hordes of consultants and IT professionals participate in animated and heated discussions decrying the pros and cons of hardware platforms, programming languages, and which is the best operating system. There is always some new piece of information to be brought forth to keep the discussion alive *ad infinitum*.

For most of us, the PC we use at home is often light years ahead in technology than the ones we use at work. This is especially true with larger companies. Have you ever wondered why? Large Companies are particularly wary of the cost and risk associated with the latest in PC technology. Their applications would all have to be tested and there is just not enough incentive. The growth rate in PC technology has outstripped Corporate America's ability to keep up. But PC vendors know that eventually Corporate America will come knocking. PC Vendors have discovered that bundling new features into the HOME PC is the best way to force large Corporations to follow suit. If they can get a few large corporations using the new product, then others will have to follow suit if they want to be able to communicate with one another.

Here is a silly analogy demonstrating how this works. The scene: In an IT conference room a team of developers for ABC Company are discussing a new project. Johnny Developer remarks that if the company would install the

DESTABILIZER 2000 this project could be completed in nearly half the time. If his manager wants it done, Johnny Developer would be happy to go home and use his HOME PC or his LINUX box to complete the project by tomorrow morning. Hey, he's got a wireless router under the sink and his kids all are all certified CNE's. In the tradition of managers everywhere, Bob Keephisjob realizes that he cannot allow this to happen; for the simple reason that he cannot manage Johnny Developer if he is at home. But Bob recognizes that if he can accomplish the project in half the time then he will look good in his manager's eyes. ABC Company "needs" The DESTABILIZER 2000. They buy it, install it, run a few tests, and then using SMS distribute it only to discover that the DESTABILIZER 2000 has updated their MDAC version and now the application they bought to handle their inventory is no longer working. Oops!

For many IT professionals it is becoming clear that cult of technology holds little regard for whether change is necessary or how it is accomplished, it just demands that change must occur. Why is this happening? Technology vendors must create new products and upgrade existing ones to create new revenue and to appear that they are staying competitive. They are on the NYSE and have an image to uphold. The slick marketing campaigns serve to convince you that your competitor has "it", and your company is left with the doubts of what will happen if they don't have "it". It is another variation of keeping up with the Joneses. The motto is now: "If it isn't changing and growing, then it must be dead." It has replaced that pearl of wisdom, "If it ain't broke, don't fix it."

Customers are so busy keeping up that at times they forget to ask a very important question? **"Do we really need to migrate?"** One tactic employed by the vendor's marketing person is to create "need" where none exists. They are very creative and can spin a mighty fine tale. I call them Marketeers. They are all smiles and confidence; and your migration questions are replied to in the affirmative or cast off into the land of no return with "I'll get back to you on that." Marketeers are so busy nodding their heads "Yes, it can do that!" that at times they begin to resemble bobble-head dolls.

But what are the risks? The reality is that downtime, lost user minutes, and lost revenue are just a poorly planned migration away. Small companies are better equipped to manage this risk. But large companies have much to lose and sometimes little to gain in keeping up with the latest technological trends. Distributing software to tens of thousands of employees can be expensive and is by its very nature filled with risk. When one application interferes with another, companies can be put into some difficult positions. In some instances, the red on the balance sheet is clearly the result of technology's bleeding edge.

In this paper I will show how changes in hardware platforms and software applications can dramatically affect the stability, capabilities, and performance of the SAS system and applications written in SAS. The focus of this paper will center on the lessons learned from two particular migration efforts:

- The migration from SAS 6.12 to SAS 8.2
- The migration from IBM S80 running Aix 4.3 to the new flagship IBM p690 running Aix 5.1

The lessons learned from these two migrations apply regardless of the hardware platform or software application. I will explore migration methodologies, tools, and benchmarking routines written in SAS that can help you in your migration efforts and in performance tuning. You and your company are the customer and have the right to decide when and if you should migrate. Tape it to your monitor, in case you start to lose your nerve when the Marketeer comes knocking.

The Migration to SAS Version 8 – One problem with huge consequences

Until SAS V8, migrating from one version of SAS was seamless. Version 6 lulled us into complacency. Going from SAS 6.08 climbing up the stairs to SAS 6.12 was so easy that it could often be accomplished without user's every knowing. At SUGI 26, I happened to be in the demo theater and heard a "SAS Presents" Paper on migration from SAS V6 to SAS V8. Within my company, it had been decided that all the platforms (Windows, AIX, and MVS) would migrate to SAS V8 at the same time. The presenter went on and on about how easy it was to convert to Version 8. SAS even recommended that you could run both versions simultaneously since V6 and V8 filename extensions were different. In this way you could stage your migration if you had doubts.

At the end of the paper, I asked the presenter about migration on the AIX platform. I had heard a rumor that migrating on AIX was more complicated under SAS V7 and wondered if it had been remedied in SAS V8. The presenter's face darkened and he relayed the information that on AIX SAS V6 Catalogs had to be transported into V8², meaning you take the V6 catalog create a transport file and then import it into SAS V8. My head was reeling. Why? We had hundreds of thousands of SAS catalogs. Also, unless you wanted to have 2 copies of each SAS dataset you would need to proc copy or rebuild your SAS V6 datasets in V8 format. I knew that would not be realistic, not with over 1 Tbyte of datasets involved. We only had enough space for one version of the data. I was stunned and as the reality began to dawn upon me as to what migrating to V8 would mean, I decided to retire to my hotel room to think about it.

At that time our SAS Server was an IBM S80 running AIX 4.3 with 12 CPUs, 16 gig of RAM, and over 1.6 Tbytes of DASD(storage space). We used SAS for Windows to connect to this server and to MVS SAS as well. My main concern was the AIX platform since it involved the most risk. With three different Datamarts on this AIX SAS Server requiring 24X7X365 access for our 400+ SAS users, I started wondering how in the world we could possibly migrate. The scope of the task was monumental. In my hotel room I began to think about the volumes and difficulties involved.

SAS Dataset Migration – 350,000 SAS Datasets

Since we did not have an extra Tbyte of DASD laying around getting a suntan, I knew we would have to migrate the SAS datasets in place. We had over 350,000 SAS datasets weighing in at over 1 Tbytes of DASD. Many of

the tables were small, but hundreds of the SAS datasets were over 2 gig and some of them over 20 gig in size. Furthermore, due to an aggressive space management campaign we had over 50,000 files that were SAS datasets compressed under Unix. Uncompressed these files added up to another 500 gig. To convert them would require us to uncompress the V6 SAS dataset, convert it to V8, and then re-compress it. It would involve a lot of overhead and processing since compression and un-compression under AIX are extremely resource intensive. And since we run a 24X7X365 shop, I started to wonder, how long does it take to rebuild 1.5 Tbytes of SAS datasets and when would we have the time to do it?

SAS Catalog Migration – 150,000 SAS Catalogs

We were heavy users of SAS AF and our applications often stored various types of output in Catalogs resulting in Source entries, SCL Lists, and other catalog member types. SAS V8 could not even touch a V6 catalog on AIX. It gave you an error the instant you attempted to use it. This meant converting catalogs would require us to use SAS v6 and PROC CPORT to create a transport file and then call SAS V8 and use PROC CIMPORT to import it into version 8. It was complicated.

Migration Planning

Considering our SAS Server availability requirements and the volume of files, it was obvious to me that this would be a huge project and that we would have to write an application to handle all these migration requirements. That application would require testing and verification. We needed some tools to make this application manageable. We would also need several weekends where we would restrict access so that we could migrate files into V8 format. Since some files might get updated along the way, we would have to re-convert some of the files that changed during this timeframe. Since there was no way it could be done in one weekend, we broke the project up into phases by application.

Migration Tools

Some of the tools we developed for this migration were:

Filelist Macro

With over 500,000 files involved, I realized that we would need a filelist macro for both Windows and AIX. This macro would use the filename PIPE statement to issue the dir command on windows and the ls -l command on AIX to read in the list of files along with their size, creation date, and other attributes. It would need to recurse subdirectories and build a SAS dataset with the files in the path with the added benefit of applying a where clause to the dataset in case the migration team wanted to limit files included in the output dataset. The output dataset would then be used by the migration application.

V6toV8 Macro

The V6toV8 macro allowed the migration team to pass a path and a where clause to convert the SAS Datasets, compressed SAS Datasets, and SAS Catalogs from SAS V6 to SAS V8. This routine would call the FILELIST macro to get the list of files to convert and then begin the work of converting. For testing purposes it needed to be able to point to a new output area or convert a file in place.

Migration Problems

After the development of the tools we would use to do the migration, the first phase was to test our migration process and the applications on a test server with a subset of our data. We converted all of our SAS AF applications (over 200 SAS AF frames, and 1000 SCL programs) and had the testing team begin testing the applications. It so happened that our largest reporting application was left towards the end of testing. It was 2 months into the project that a problem was discovered. The reporting application involved creating graphical reports that displayed graphs and spreadsheets on the same page. Pages were combined into Postscript or PDF files resulting in packages with hundreds of pages. These reports went directly to our customers. They were our bread and butter. In SAS V8, the application was bombing.

This reporting application used a graphics language that I had developed during our migration from OS/2 to Windows some years back. Pre Year 2000, we had been using Lotus for DOS to create our automating reporting packages. Hundreds of pages of Lotus macros controlled the automation. We could not migrate this to Windows since LOTUS for DOS was being sunset within our organization. Nobody was excited about rewriting these macros in EXCEL. Our reports have to be extremely flexible and the resulting PDF packages grew in some cases to over 200 pages.

The objects on the outputted page (Graphs and Spreadsheets) vary by customer and the years of data they have. Rows may be removed or columns toggled on and off. During the migration from OS/2 to Windows, I wrote the Graph language using SCL, SAS Pattern Matching(rxparse), and DSGI (data step graphics interface). DSGI allows you to put anything on a SAS GRSEG entry: Tables, Graphs, Notes. Of course the only problem is that if I wanted a procedure that made vbar graphs with 3D bars, then I would have to program it. It took me a total of six months to develop this language. It was a new reporting language that gave us high quality graphical reports that had proportional fonts, something that SAS 6.12 did not provide on its own. To create a report that had a 3D pie chart and a spreadsheet you submitted something like the following:

```
%REPORT(

    options font=times height=2;

    title "My Report";

    PROC PIE data=work.mydata 3D;
        group by year;
        var paid_amt;
    RUN;

    PROC REPORT data=work.mydata;
        when desc='TOTAL' then font=timesb;
    RUN;

);
```

Flexibility could be added by putting using the SAS Macro language within the graphics language. What was the problem? In Version 8, SAS changed the graphics drivers to be compatible with the True Type Fonts. The result was that the fonts were 15% smaller than under SAS V6. Although I had worked with Tech Support in building the language, we were one of the few customers who knew and used DSGI.

Of all the bugs we found with SAS V8 this one was the most devastating. The Graph Procedures I had developed (Vbar, Hbar, Pie, Line Plot) had to draw every component of the graph, and my offset formulas, which provided spacing between the various components of a Graph ended up generating negative numbers with the reduced driver font sizes under SAS V8. The graphs looked inside out. If someone needed a Tapeworm Plot, well I now had software to do it. The spreadsheets would come out smaller but at least they came out. All of the graph procedures I had developed bombed. It was both a personal and a professional blow. This change was not documented anywhere that I could find. SAS confirmed the mistake, but I was still left with a huge dilemma.

I had spent six months of my life writing the graphics language in 1999. When printed out and stacked, the various SCL entries and methods (Oh how proud I was of it's object oriented design) created a pile of paper that was over 5 inches thick. It was now 2001 and we were going to start the migration to SAS V8 very soon. All the other problems we had found with SAS V8, SAS had helped us to overcome. SAS sent out HOTFIXES almost daily for a while. The graphics driver problem remained and so did my dilemma. I started to wonder how this all got so complicated. Why were we migrating in the first place? I could no longer remember.

Should I fix my graphics language or should we try to convert the reports into ODS? In the end neither of those solutions was viable. My estimates showed that it would take 3 months to rewire the graphics language and at least 6 months with a team of 4 to rewrite the reports in ODS which was at that time an unknown quantity for us. Furthermore, I would not have time to work on this for I was being pulled to work on an exciting new project that required some immediate Research. The V8 migration team would have to go forward with their testing and application conversion while I started the design research for this new application.

As I was soon to learn, this application would also be a reporting application, but would require even more flexibility. Furthermore, it required new types of graphs that I did not have in my graphics language. The anticipated volume of reports (hundreds of thousands) would require a whole new AIX server just for this application. After a month of discussions I found myself back where I started. It became clear that this new reporting application required more flexibility and graphical requirements than the graphics language could supply in its current state. I would either have to have to fix the graphics language and add a vbar with cylinders, which the new application required, or I would have to abandon it and use ODS, which meant a lot of research and development. It was a huge risk with a huge price tag for failure. I chose ODS for the new application and it turned out to be the write decision.

Before I could start developing on the new project my boss convinced me to find a quick and easy way that the graphics language could work just for that one reporting application. This reporting application would be sunset in the future anyway. The graphics language worked in SAS 6.12 but the users and servers would soon be SAS v8. My solution was that this application would run in SAS V8, and when it got time to produce the reports it would then write a SAS program that created the reports using SAS 6.12. SAS 8.2 would call SAS 6.12 using %sysexec for the purpose of building the reports. SAS 6.12 would be retained on the AIX server until the application was sunset. It was not a perfect solution, but it worked.

In the end, it took over a year for us to completely migrate to SAS V8. Most of that time was spent in developing the tools and testing the migration and building the migration plan. It was not a year's worth of work, but it was a year nonetheless. The testing, patience, and planning was well worth the time, I know this because our users experienced very little downtime during the migration.

LESSONS LEARNED IN MIGRATING FROM SAS 6.12 TO SASV8

- As I have shown, one small change in SAS or any other application can have far reaching implications in your applications. Even after reviewing the Changes and Enhancements guides that SAS provides there are still hidden bugs that could have a dramatic impact on your applications. The change in the SAS/Graph Drivers with fonts that were 15% smaller under SAS V8 is a perfect example of this. The reality is that many vendors are relying on their customers to participate in the testing of their software. They cannot possibly begin to know all the different ways their customers will use their applications.
- Large companies should resist migrating to the first release of a new version of SAS. For example we waited until SAS 8.2 before we moved to SAS V8. Wait until the .1 or .2 release before you migrate. This will ensure that many of the bugs in the .0 release will have been resolved by time you start your migration.
- Create Migration Tools that allow you to automate as much of the migration as possible. The migration tools that we developed specifically for the SAS V8 migration gave us the ability to test the migration and saved us enormous amounts of time. When major migration efforts come your way, their reusability will save you time and effort.
- Before you migrate, take the time to do a Risk Assessment Analysis. Customers with large numbers of SAS users and SAS Applications have the most risk. They have the most to gain and the most to lose. The unfortunate side effect of our SAS V8 Migration experience is that migration to a new version of SAS is now met with extreme reservation. I have to remind everyone that the SAS V8 migration was the only one we have had difficulty with in the last 9 years. But people have a short memory and maybe that is for the best. For the risks that migrating to version 8 exposed

made us realize how much we have to lose when things do not go as planned.

- When migrating and testing, your most important applications should come first. Make a list of your applications and rate them based on how important they are to your company. Do some contingency planning on what would happen if that application broke. Don't assume anything. Test all of the features. Many companies find it helpful to have a group of systems and business users that are formed into a testing team. They have scripts they follow to ensure that tests are run the same way each time.
- Before building your Migration Timeline make sure you have tested your migration process. How long does it take to convert a 5 gig SAS Dataset from V6 to V8? It is important to know things like that and build in overhead. How many files can your server comfortably convert in a multi-threaded environment. Migration can consume vast amounts of resources. One of the migration risks is that you will overload your server. It can happen. Adequate testing of your migration process will help you find your thresholds.
- Do not believe everything you read. Confirm your doubts with SAS Tech Support and then test for yourself. If you don't agree with what SAS Tech Support tells you, ask them to refer it to the developers.
- Be careful of what you hear at SUGI. SAS users often come back from SUGI full of vim and vigor. It is an exciting time but many return forgetting that some of the features they are most excited about are actually experimental. The quest to implement a new version of SAS to get a new feature should not blind you to the risks that other features in the new version of SAS may break your existing applications.
- Everyone has different priorities. Often SAS is at the mercy of the vendors that their software depends on. Each of these vendors in turn has their own priorities. The migration pathway for SAS on AIX was made difficult due to design conflicts between SAS V8 and the AIX operating system. SAS has different deadlines than AIX, Sun, or Microsoft. For SAS V8 and AIX, it was a case of bad timing. SAS and IBM had different priorities and they were unable to resolve this conflict before SAS V8 was to be released. In the end, the customer pays the price. But if you let your vendors know what it cost you, you might be surprised how quickly they will respond the next time around. SAS has been very responsive to my complaints about the difficulties in migrating to V8. I wrote an emotional criticism about SAS V8 in late year 2000 and I am still getting calls from concerned parties at SAS wanting to know if they have addressed all my concerns yet. They let me vent and then they actually started addressing the migration process.

Migrating to the IBM P690 Regatta Server

As I mentioned in the SAS V8 migration above, part way through our V8 migration, a new project came up that

required SAS V8 and a new AIX server scaled to handle the anticipated load. Hundreds of thousands of reports would need to be generated in an automated system that at times involved querying files in excess of over 100 gig. We had to use these large files as our source for building a Relational Datamart. Talk about load! After much planning and discussion, the IBM P690 running AIX 5.1 was chosen to be our new platform. All the press was good. Some claiming it was the fastest Midrange server on the planet. The risk was anticipated as low.

The P690 Hardware was configured as follows:

- Disk Storage: 2.6Tbytes of IBM Shark DASD
- Processors: 12 X 1.3 ghz processors
- RAM: 16Gig (We had actually had a total of 32 with 16 held for later use if needed.)

The Software relative to the project

- SAS 8.2
- Aix 5.1 operating system
- Apache Web Server
- Jakarta JSP Engine
- Adobe Distiller 3.1

BENCHMARKS

Once we had the P690 up and running, the plan was to benchmark it against the IBM S80 that was our other SAS Server. I had written a Benchmark Macro in SAS that had proved very helpful in comparing various platforms. The BENCHMARK macro worked for Windows, OS/2, UNIX(all flavors), and even the mainframe. It built its own data of a size the user controlled and then executed a series of SAS Steps against that data keeping track of the amount of time it took for each step and storing these in a SAS Dataset called a PDB(Performance Database). There was no limit to the size of the Test data that it would build. More importantly the BENCHMARK macro only required that you had SAS and that it was in your SASAUTOS Macro path. The steps were as follows:

- step1=Make TEST Data
- step2=Make Formats from Test Data
- step3=Make Main Table
- step4=Sort Main Table
- step5=Query without Index
- step6=Index Table
- step7=Query with Index
- step8=Build View
- step9=Query with View
- step10=Read from Table
- step11=Create Report
- step12=Copy Table using OS Command
- step13=Delete Table using OS Command
- step14=Copy Table using Data Step
- step15=Delete Table using Proc Datasets

We would run an apples to apples comparison on both servers while both boxes were quiet, meaning that nothing but the benchmark routine was using the server.

The IBM S80 was configured as follows:

- Disk Storage: 1.6Tbytes of striped EMC Symmetrix DASD
- Processors: 12 X 400 mhz processors
- RAM: 32Gig (only 16 gig was configured in the beginning. We would add more as needed.) use should we need it.

We configured SAS the same on both AIX servers with each server having a memsize=256m and a sortsize=240m. We ran all sorts of benchmark configurations (4jobs with 25gig files, 25jobs with 4 gig files etc...) but to properly stress the system we attempted to simulate the load on the system by having the BENCHMARK routine run 200 jobs at a time building and processing a file of 50m. The results were shocking were so contrary to our expectations that I re-ran the entire Benchmark Stream several times. The following results were found:

- Benchmark after benchmark showed that the IBM S80 was outperforming its big brother by a factor of 3 across the board. People were getting panicky.
- The P690 could not even complete the 200 job X 50m file benchmark. It ran out of paging space. This meant that this benchmark not only used up 16gig of RAM but 8gig of Paging space as well. This made no sense. When I did the math (200jobs X 50mb memory limit) I came up with a little over 13 gig. How could it possibly be using at least 24gig?
- AIX 5.1 did not work well with files over 2 gig. You could not delete, copy, or move a file over 2 gig.

At this time, the System Engineers and myself forged a much closer relationship. This was critical to solving the problems and we all assumed there was a problem. The System Engineers had the ties to IBM and I had ties to SAS. While they worked with IBM, I contacted SAS and was put into contact with the CTC (Customer Technology Center) who gave me several employees from IBM who reside at the SAS Institute and were there to help me. It was a great relief to know that they were there.

IBM was very responsive and started issuing hotfixes to solve the 2 gig limits and some of the minor problems we were finding. But there was so much pressure in my opinion these hotfixes were churned out too fast for proper testing. One hotfix would fix a certain problem but then a new problem was created. We were caught in the spiral of Ouroboros(the serpent consuming its own tail).

Luckily we had built in six months for setup and testing. We ended up needing it. I knew that IBM would eventually resolve the 2 gig issues and we would get to the bottom of the AIX stability. What I could not understand was why we could run the 200job benchmark on the IBM S80 but could not on the P690. In the end, one of the System Engineers recommended trying various memsize sortsize settings to see if we could get the benchmark to finish and to stop paging.

Trial and Error is sometimes the best solution and in this case it proved golden. Through trial and error we discovered that if we set the memsize to 64m and made the sortsize 16m we not only got the benchmark to run on the P690, but it was now more than twice as fast as the IBM S80 results. We then changed the IBM S80 SAS configuration to use the new memory settings and we saw to our surprise that it improved in performance as well. The P690 was still nearly twice as fast, but we saw improved performance by using these new memory settings. When we need more memory we try to keep the sortsize at a 1:4 factor in relation to memsize.

Here are the benchmark results for the 200Job with 50 m file benchmark with memsize=64m and sortsize=16m

IBM S80 – 60 minutes
IBM P690 – 32 Minutes

In particular we saw most of the improvement with PROC SORT and PROC SQL with order by clauses. So what was causing all the paging on the P690. It turns out that running SAS 8.2 which is 32bit on AIX 5.1 which is 64 bit results in extra memory requirements as each 32bit word is expanded to 64bit. We expect that this problem will go away once we migrate to V9. There are still a few patch issues (see below) but by and large we are happy with the P690. More importantly, SAS and IBM showed that they valued us as customers and helped us to resolve our issues.

LESSONS LEARNED IN MIGRATING TO THE IBM P690 REGATTA SERVER

- No one node on a IBM P690 can use more than 50% of the Total RAM. We discovered this the hard way. So if you bought a P690 with 64gig of RAM and carved it up into 3 nodes, the max any one node could have is 32G.
- The minimum number of processors for a node on an IBM P690 is 2.
- If you think you are going to potentially need to expand memory later on, save your company hundreds of thousands of dollars and make sure you purchase RAM in multiples that allow you to add new cards. Otherwise when you decide to expand, you will be forced to replace all your memory cards. Save yourself the money up front.
- When things go wrong, try to keep your composure and do not alienate your systems or vendor support staff. You will need them and a positive relationship to resolve these issues. It is easy to overreact when you think about the money, but when you purchase a new server just recently put on the market there is a large amount of risk involved. An itsy bitsy bug can have a large footprint. So keep your vendor informed of your deadline and maintain pressure, but keep your composure.
- In my first SAS class, users often asked the instructor John McCall "Which is faster...?" His result was a frustrating, "Benchmark it!" It is the best advice I have

received regarding SAS. Benchmarking and Testing will allow you to speak with confidence when you do your project planning. They also help you to know the limits of your systems and how to design your applications accordingly.

- Using SAS for Benchmarking gives you a great advantage over your System Administrators who are often unknowledgeable about your applications and are limited to benchmark scripts. Using SAS allows you to leverage the language and data engine of the SAS system to build a robust Benchmarking Application. Even if you are benchmarking Software other than SAS you can use SAS with the %sysexec, %systask, or filename PIPE features to construct complicated benchmarks that will allow you to find out how your application with all of its components performs under various scenarios. I have an open source version of my SAS Benchmark routine that I will distribute to those who attend the presentation of this paper.
- Striping the DASD, especially the work file system and any file system that you anticipate concurrent I/O is a great way to improve performance. If you don't know what striping is then read the papers shown in the reference section. It is important to stripe your disks when the file systems are initially built and that the SAS buffer size is a multiple of the stripe. It is much more difficult to do it later when there is a lot of data on the system. Your AIX System Administrator will thank you for being so thoughtful.
- The Benchmark process is an opportunity to forge a much closer relationship with your System Administrators. In many of the larger companies, there is a heavy emphasis on process and roles. Each group does one thing and they are supposed to do it very well. During migrations and in benchmarking, the various groups have to come together.
- Having a good project manager armed with a good project plan is critical to successful migrations.
- Under AIX, our Memory Tuning Magic Formula turned out to be memsize of 64m with sortsize of 16m. SAS has found that on the IBM P690 the greater the memsize setting the poorer the performance. It's counter-intuitive and should be the other way around but it is not. Part of the problem is running 32 bit SAS 8.2 on a 64 bit platform(Aix 5.1) These new memory settings helped us to get rid of SAS Abends caused by Segmentation Violation errors. For the last eight years we have been plagued during heavy use with the mysterious Segmentation Violations. It's been by our company's side for so long we decided to give it benefits. No Rhyme or Reason to it. An average of 5 tickets a year and still nobody could get it to go away. These memory settings were an unexpected solution to this problem. *Your own Magic Memory Tuning Formula may vary. Offer void in OHIO.*
- A Benchmarking routine that is self-contained is able to help you prove that your tuning is working. Can your system handle files larger than 2 Gig? Did the striping of your SASWORK file system give you any improvement? Is your OS/2 File Server faster than your new NT File Server. Which is faster MVS or AIX? All of these questions can be answered with a BENCHMARK routine. It allows you the ability to prove how your performance has improved or degraded.
- It is important to Benchmark when the System is Quiet, but it is equally important to Benchmark while the system is under normal use. It's the only way you are going to be able to make sure that your applications can co-exist with one another
- The side effects of using open source software or creating complicated home grown solutions instead of purchasing off the shelf products can expose your company to excessive risk when changes arrives via your OS upgrade, hardware platform migration, and SAS version upgrade. If you use open source software, make sure you have internal support or be prepared to support it yourself.
- You can use your benchmark routine as a load indicator. By gathering baseline values of how long our benchmark runs when the box is not being used, moderately busy, and when it is getting thrashed; you provide yourself with a barometer for controlling the load on your systems.
- Installing Adobe Distiller 3.1 on AIX 5.1 requires that you change the makefile so that Distiller thinks the server is running Aix 4.3.
- With the latest AIX 5.1 patches on the IBM P690 there is a problem with large numbers of concurrent threads calling Adobe Distiller 3.1. This problem showed up after a patch we received in October. You will get a error in the Distiller Log reporting that a Signal 4 was recieved while attempting to build the PDF file. To get around this problem, create a SAS Macro or script that keeps attempting to distill the file until it gets a 0 return code or reaches a loop limit. Since we put this fix in, we have not had any problems with this bug.
- With the latest AIX 5.1 patches on the IBM P690 there is a new problem that involves the performance monitor Topas. As of press time, It is reporting incorrect memory and paging numbers and can take up to 10 minutes to start up. IBM has issued a patch that is supposed to resolve this and we will be testing it in early 2003.

Migrating to Version 9

We have started planning and testing for our migration to SAS V9. Am I scared? You had better believe it, but with the two difficult migrations under our belt, I am at least confident about how to go about the migration process. We are expecting this migration to go much smoother. Ever the optimist I am.

According to SAS, V9 for AIX will only run on 64bit, which means that you will have to migrate to AIX 5.1. It appears that as long as you do not make use of V9's support for longer format names, you can use SAS V8 and V9 datasets interchangeably. The file extension is the same.

SAS Catalogs however; still need to be transported from V8 to V9.

For a very important document dealing with Processing 32-Bit Version 6, 7, or 8 SAS Files with 64-Bit SAS Version 9 see the web site:

<<http://www.sas.com/rnd/base/topics/convert9/>>

CONCLUSION

In this paper I have attempted to demonstrate the various pitfalls that arise when migrating your hardware or software. Proper planning and risk assessment can help prevent long-term outages should things go wrong. Patience is important when they do go wrong. Let your vendors know when you are displeased. I have also hoped to demonstrate that creating a set of Software tools for Benchmarking can help you perform the all important testing for when you do migrate.

REFERENCES

¹ Chris McCoy. "What is the ouroboros?"
<http://www.dragon.org/chris/ouroboros.html>

² SAS Institute Inc. "**V6 AIX catalogs are not compatible with V7/V8**"
<<http://www.sas.com/service/techsup/unotes/SN/000/000069.html>>

RESOURCES

Here are some papers and web sites worth reading.

For an excellent paper for tuning the SAS system read: Tom Brown. "**A Practical Approach to Solving Performance Problems with the SAS System**" *Proceedings of the Twenty-Seventh Annual SAS® Users Group International Conference*. March 2002.
<<http://www.sas.com/rnd/papers/sugi27/SolvingPerformance.pdf>>

For an excellent discussion of how SMP Unix Servers work read:

John E. Bentley. "**SAS Multi-Process Connect: What, When, Where, How, and Why**" Paper 269-26 *Proceedings of the Twenty-Sixth Annual SAS® Users Group International Conference*. March 2001.

Clarke Thacher, SAS Institute Inc. "**Tuning the SAS® System for UNIX and Tuning UNIX for the SAS System**" Paper 294-25 *Proceedings of the Twenty-Fifth Annual SAS® Users Group International Conference*. March 2000.

Jeff LeSueur. "**Minimizing Development Risk While Maximizing Warehouse Performance on UNIX® Systems**" Paper 120-25 *Proceedings of the Twenty-Fifth Annual SAS® Users Group International Conference*. March 2000.

Kimberly J. LeBouton, Thomas W. Rice. "**Smokin' With UNIX Pipes**" Paper 103-25. *Proceedings of the Twenty-*

Fifth Annual SAS® Users Group International Conference. March 2000.

Maria L. Moore "**Using the SAS System System under AIX Effectively**" Contact SAS Support for more on this paper.

Margaret Crevar, Leigh Ihnen, Gary Mehler. "**Getting the Most Out of Your High-End UNIX and NT Server with SAS®**"

Paper 268-26 *Proceedings of the Twenty-Sixth Annual SAS® Users Group International Conference*. March 2001.

If you are migrating to Version 9 read the following paper: SAS Institute Inc. "Processing 32-Bit Version 6, 7, or 8 SAS Files with 64-Bit SAS Version 9"
<http://www.sas.com/rnd/base/topics/convert9/>

IBM White Papers related to SAS and Performance
<http://www.sas.com/partners/directory/ibm/whitepap.html>

Technical Note discussing Catalog incompatibility from V6 to V7+
<<http://www.sas.com/service/techsup/unotes/SN/000/000069.html>>

Platform Partners
<http://www.sas.com/partners/programs/platform/index.html>

IBM Partner Site
<http://www.sas.com/partners/directory/ibm/index.html>

IBM White Papers related to SAS and Performance
<http://www.sas.com/partners/directory/ibm/whitepap.html>

Optimizing SAS Performance on AIX
<http://www.sas.com/partners/directory/ibm/optimize.html>

SAS Technical Tips
http://www.sas.com/service/techtips/ts_qa/

SAS Administrator Resources
<http://www.sas.com/service/admin/unix/>

For Information on AIX and other IBM products:
<http://www.redbooks.ibm.com/>

ACKNOWLEDGEMENTS

I would like to thank the Customer Technology Center at SAS and Torre Devito of IBM for helping us during our migration to the IBM p690. I would also like to thank Alec Fernandez at SAS for putting up with my rants and raves during the SAS V8 migration. And I would especially like to thank SAS Technical Support. They are an indispensable resource for me.

TRADEMARK CITATIONS

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. Other brand and product names are registered trademarks or trademarks of their respective

CONTACT INFORMATION

For more information contact the author at:

Charlie Bastnagel
Cigna Healthcare
1111 Market Street, Stop BR2A
Chattanooga, TN 37411

Email: cebast@bellsouth.net