Why Tune?

I was invited to give a keynote speech for the SHARE conference in Chicago. The theme of their "conference within a conference" was Capacity Planning and Performance Management. I chose the topic "Is Tuning Dead?" because many people seem to feel that it's easier to buy more hardware than manage the system. I don't believe that's true. I've taken the subject of that presentation and expanded it for my readers.

With the drop in hardware costs and escalating people costs, why not just upgrade your processor instead of tuning it? Many people are doing just that. So is there any reason to continue to have someone monitoring and analyzing the performance of your installation?

The answer is a definite yes! Let's look at how performance analysis has changed in the past few years, how it will change in the next few years, and why it's even more important today than at anytime in the past.

Introduction

This discussion is about tuning, performance management and capacity planning. The key question is whether it's cheaper to buy more hardware or to spend more money managing it.

It all comes down to the money. I want to address four items:

1. What does it cost you to tune or manage your system?
2. What can you save by tuning or managing it?
3. How do you justify it?
4. What does the future hold?

What Does it Cost?

To determine the cost of performance management and capacity planning (PM/CP from here on), you need to look at the cost of all of the resources. (PM/CP is also used to mean Performance Manager and Capacity Planner, the person doing PM/CP.)

- **Salary, Benefits and Training -** Salary and benefits for a PM/CP vary by location and availability. I have heard of ranges from $50,000 a year to $120,000 a year, with capacity planners at the higher end. Benefits and training must be added to that figure. A PM/CP who gets no training on the latest technology loses a lot of his/her value. More about that later, but ensure that training is considered one of the requirements of a good PM/CP. For my examples, I'll use $8,500 per month for a PM/CP. Your figures will vary.

- **Other People's Time -** Nobody works (effectively) in vacuum. Any PM/CP will need to have meetings with their users, managers, sysprogs, DBAs, application programmers, operators, etc. You should consider the time of others when determining how much the PM/CP effort is taking. Several sites I know use a factor of 25% of the PM/CPs time.

- **Computer Resources** - This can be quite large. However, most of the resources are free. If you consider that most data collection and processing is done at night when resources are available, the processing costs can often be ignored. If not (for example, you need to run 20 hours of processing to convert a file for future performance benefits), then you can compute the resources for that specific effort and subtract the cost of processing from the total savings. (More about that later.)

After taking these costs of a PM/CP effort into account, you should have some idea of the cost of such an effort. Just for this article, I'm going to use $10,000 per month per PM/CP.

What Can You Save?

Value of Resources Saved

The resources you can save by tuning or proper capacity planning are: CPU, software costs, storage, DASD, tapes, paper, staff time, elapsed time, user time, and customer satisfaction.

- **CPU Time** - You may already have a value for CPU time in your site. If so, you can skip this paragraph and use your value instead. Otherwise, you should try to quantify the cost of a CPU second/minute/hour/month at your site. That's often simply the cost of the CPU lease each month. Or you can take an industry average. As I mentioned in the last issue, Amdahl is now posting some of their list prices for CMOS processors on their Web site <**http://www.eShopAmdahl.com**>. Using their data, you should expect to never pay more than $2600 per MIPS for purchase or $100 per MIPS per month (on a 2-year lease). Most sites double that number to account for maintenance, environmentals, space rent, and support staff (sysprogs, operators, schedulers, tape/print operators, and managers) to $200/MIPS/month.

  Understand that CPU savings are only a savings if you need that processing time for something else. Saving 10 minutes of CPU time from a job that runs at 3 am in a site where the nightly batch cycle completes an hour before needed is not really ***saving*** anything! Saving it from peak period transactions in a CICS region that is constrained at 10 am ***does*** provide a savings. Be sure that the savings matters.

- **Memory** - Again, you may already have a cost for this. If not, you can use Amdahl's site again to find that memory storage costs $50,000 per 512 MB purchase or $4 per MB per month (over 2 years). Try not to pay more than this for memory.

- **Software Costs** - These are fairly difficult to determine without a lot of effort, but it's very important that you do. Software costs may be the most significant savings you can achieve. These come in most frequently when you are able to delay an upgrade of your machine for some number of months. After you determine the increase in software charges you will pay after the next upgrade, you can convert that to the savings achieved for the delay. In some sites, that amount is equal to the CPU costs, while in others it may be double the CPU costs. Even if you simply use a percent of the CPU costs, please don't ignore this savings. It's significant!

- **DASD** - DASD storage costs between $.25 and $.50 per MB for three years (or $12,000 per terabyte per month). If you're paying more than that, it's time to look at other vendors.

- **Tapes, Paper, Staff time** - These costs might be reduced by a tuning effort, but will vary for each instance. Wait until you see what the savings are to quantify the amount. It should be easy enough to do.

- **Elapsed Time** - This is a tough one to quantify because it's very site-specific. For some sites, reducing the batch window might not produce any savings at all. You were finishing batch by 4 am and after tuning you finish by 3 am. All you did was give your system more idle time. If, however, the batch cycle was encroaching on your online startup and delaying customers, then a reduction in elapsed time could be significant. For some sites, reducing the batch window by an hour, for example, could delay a CPU upgrade for several months. That's worth something! There is also a subtle savings in CPU time and storage when elapsed time is dropped significantly. If the savings is due to reduced I/O response times, you'll often see a corresponding drop in CPU time. I think the value of elapsed time will vary for each instance and

should be evaluated during the justification period for each project.

- **Customer Satisfaction and User Time** - These more esoteric savings are difficult to measure. Since I've generally found enough savings in the hard resources to justify most efforts, I've seldom needed to try to put a dollar value on customer savings, but ***you*** may need to. The value will depend on a lot on the type of business you have, what the competition is, and how patient your customers are.

  I once switched banks because the ATM for my bank was never available at 5 am when I needed it. That was ten years ago. That's a lot of lost revenue from just one person. IBM tells of the story of Barnes & Noble, stung from the success of Amazon, who initiated and completed their online e-commerce application in nine months to successfully compete with Amazon. For them it was a matter of survival. Stories abound about customers leaving their brokerage firms, or banks, or Internet providers, or whoever because of poor response times. Customers today have more options available to them and less patience. You simply ***must*** provide responsive systems to maintain your current customer base and draw others to you. I can't give you a good estimate of this savings, but you may need to. If all else fails, ask your management how important each customer is.

What does this all mean? To simplify this - in order to justify the costs for your PM/CP, you would need to save the following:

- For CPU at $300/MIPS/month (this takes into account staff time and software costs), it would take a savings of 33 MIPS/month to justify a $10,000/month PM/CP.

- For memory at $2100/512MB/month, it would take a savings of 2.5 GB/month to justify one PM/CP.

- For DASD at $.40/MB/3years, it would take .8 TB/month for one PM/CP.

This says that the low cost of storage and DASD makes it much harder to justify the effort needed to reduce them. That doesn't mean that I/O tuning isn't worth it. Tuning I/O response times reduces CPU time and elapsed time and is almost always worth the effort. Reducing DASD space, on the other hand, is seldom worth it. There are always exceptions, and I have some shown below in our success stories.

Success Stories

It's extremely important to quantify the actual savings of each of your tuning or capacity planning efforts. In preparation for this article, I solicited stories from users that show the value of tuning and correct capacity planning. I've included most of them below with an explanation of what they did to achieve their savings. You'll notice that it's especially effective when a dollar amount can be specified.

I've included the full stories so you can get ideas about what might work in your site. As you read, you'll notice frequent references to Strobe. (I'm not on their payroll!) Strobe, by Programart (just announced as being acquired by Compuware), is a product which analyzes programs and shows where the time is being spent (either using CPU or waiting). The reason there are so many Strobe stories is that the product is not inexpensive and users must quantify their savings to justify the product. They've ***had*** to put a dollar amount on their efforts. Another reason I'm including the Strobe examples is because they show that simple changes often produce significant savings. Often small changes in COBOL options or buffering can produce very large savings in CPU time or elapsed time, even if you don't have a product to identify the cause of delay.

These stories should give you ideas on how to quantify savings for your management. Notice that the most effective examples are where dollar amounts are specified.

Contention - Big Time!

George Alan Esworthy of SAS Institute Inc. provided the following item. I've had a similar experience.

"If you're interested in covering the full spectrum of tuning experiences, I think this one may fit all the way at one end (you pick which end!).

"More than 20 years ago I went to work for a small insurance company in the Virginia suburbs of Washington, DC. They were a VS1 shop at the time, running an Amdahl 470V. I was the first real systems programmer they'd ever had. Previously, the operations manager did all software installation, and he'd just been promoted to manage a new project.

"When I arrived, the company was considering an expensive CPU upgrade because they were having real problems completing all the overnight batch work in time for the start of business the next day. On my first day there, I walked by the string of DASD and touched each drive. One was vibrating madly, obviously seeking all over the place. It was the SYSRES pack.

"I dug around, mapped the drive, and found that the sort and scheduler work area data sets were on that one volume, along with both the system and main application load libraries. The following Sunday, I distributed the SWAs and SORTWKs and moved the application library to another drive.

"On Monday, batch processing was all done by 10:45 p.m. We took the two midnight shift operators and moved one to second shift and the other we promoted into a junior programmer slot. The computer room was dark from before midnight to 7:00 a.m. It was another three or four years before they needed that CPU upgrade.

"It was the only time I've ever heard of a system that was tuned literally by a 'laying on of hands'."

Net Savings: One shift and delay of CPU upgrade for 3-4 years.

COBOL Options

Bruce Rogge of Compuware Corp. contributed the following:

"While researching a simple production job step that was timing out, I discovered that the Cobol program was compiled with both Trace and Flow activated. A review of our production library showed these options in over 70% of the modules. While great for debugging in test, or troublesome production, these options were just overhead in most production modules.

"Recompiling to remove Trace and Flow returned enough capacity to defer a processor upgrade by 8 months."

Net Savings: Delay of CPU upgrade for 8 months.

IDMS Tuning

A reader writes:

"We have a current production IDMS CICS application running on a IBM 9672-R46 which comes up each morning after the overnight batch run and is available to dealers in the stock exchange. If the system is up late, the users are unable to do their dealing. The cost of a late deal or a missed settlement can be enormous.

"Up to 6 months ago, the overnight batch was regularly finishing between 7 and 9 in the morning; on the occasions when there were problems with one of the jobs which demanded a rerun - which happens with monotonous regularity, perhaps once a week on average - the online system could come up much later than this, causing a big cost for the business.

"We have since brought the normal finish of the overnight batch stream to between 4 am and 5 am in the morning, and there is potential for bringing the time back even further; this has meant that even when there are problems with the batch stream, in the past two months we have always finished before 7 am.

"This was achieved by:

- Understanding the jobs that made up the critical job stream.

- Adjusting the schedule as necessary (there was for instance an extra hour's delay built in for the winter time variation between countries, that had never been taken out for summer time).

- VSAM tuning - use of batch LSR to reduce VSAM Index I/O; some jobs reduced from 15 minutes to 2 minutes elapsed.

- IDMS buffer pool tuning - to ensure that enough buffers of the right type in the right pool are available to the right job. This is a process that produces the best return for IDMS jobs, and is still ongoing. Some jobs have had hours knocked off them by correct use of buffers. One database was allocated to its own buffer pool; as a result, one million IDMS I/Os per day were saved.

- IDMS tuning - removing unnecessary internal locks on fields and removing unnecessary database calls.

- IDMS data base placement - separating the most active IDMS databases onto virtual disks of their own. This surprisingly had no measurable effect.

- Reducing the amount of data processed by running regular archives of old data. This knocked half an hour off one of the critical batch jobs.

"This has been achieved by having a regular performance meeting between application support, operations support, systems programmers, DBAs and performance/capacity people to discuss current performance problems and apply technical and organisational solutions to perceived problems; this was done with the encouragement and support of data centre management who have been pleased with the results achieved.

"Further action planned:

- Implement regular database reorganisations to reduce the amount of unnecessary I/O (the databases have *never* been reorganised).

- Further implementation of batch LSR for VSAM file tuning.

- Further IDMS buffer pool tuning.

- Further blocksize tuning."

Net Savings: 4 hours off batch cycle; unquantified savings from better user availability.

Strobe Success Stories

Amy Bethke of Programart has lots of good stories about their application performance management product, Strobe:

"Every year, Programart conducts an Application Performance Awards contest and invites our customers to submit their before/after success stories and STROBE Profiles.

"Last year, I believe total $ savings reported by entrants were over $10 million (and that's just those that quantify savings in $), run time savings were ~8,000 hrs/year.

"Here's a brief description of some of the case studies from recent contests:

1. "During an annual performance review of an IMS/DC and COBOL-based application, this company's performance team found that CPU time per transaction had recently doubled. Without improvements, the organization knew they would be unable to meet their holiday peak workload with existing capacity.

   "STROBE Performance Profiles from a measurement of the application revealed that SVC011 (TIME) was consuming 86% of the region's CPU time. Using this information, the

team identified that the CPU time used to time-stamp data records had increased as a result of a recent CPU upgrade. The problem was related to a time-stamp processing operation used to prevent duplicate records. The program was doing a COBOL ACCEPT TIME statement until the second changed. The faster processor, from the recent upgrade, was able to execute this loop twice as many times per second, thus using more MIPS. A 50 ms wait was inserted into the procedure, forcing execution 20 times per second instead of 80,000 times per second.

"As a result of these changes, CPU consumption in the online regions was reduced by 96% and the number of MIPS used by the application during peak periods was reduced from nearly 74 to less than 1. The organization was able to process the annual peak workload with existing capacity and postpone their next CPU upgrade by 9 months.

Net Savings:
Delay of CPU upgrade for 9 months.

2. Application: Weekly batch processing application that uses DB2, IMS/DB, and COBOL.

   Problem: Job stream was finishing a full day behind schedule.

   STROBE analysis: STROBE Performance Profiles identified opportunities to improve wait and reduce CPU time. Several improvements were made to the application including file buffering parameter changes, and changes to SQL statement access paths. In one situation, STROBE identified two company-authored routines that were substituted years ago for standard COBOL READ and WRITE operations. Written to save on DASD space, the routines were consuming excessive quantities of CPU time. The company restored standard COBOL operations. These same routines were buried in over 3000 of the company's other application programs. Annual estimated savings for all changes are $4.07 million.

Net Savings: $4.07 million annually.

3. Application: ADABAS region within a corporate credit application.

   Problem: Response time had gradually degraded over a period of months and was impacting timely delivery of credit reports.

   STROBE analysis: STROBE Performance Profile reports revealed that a majority of wait time and I/O was attributed to access of ADABAS command log files. Investigation showed that these logs were used for billing customers and occasional troubleshooting. The company decided to switch off command logging and use high-level database statistics instead to bill end users. For troubleshooting, the command logs could be switched on for solving ad-hoc problems. As a result of changes, credit reports were produced on time. Annual savings were estimated at over $980,000.

Net Savings: $980,000 annually.

4. Application: DB2 and Natural-based batch processing application that processes insurance policy data.

   Problem: The job stream was running for nearly 40 hours before it was canceled. The company estimated that the job would need to run for 40 days in order to complete processing.

   STROBE analysis: STROBE Performance Profiles pinpointed 2 DB2 SQL statements that were using over 76% of the job step's CPU time. Indexes were added to the DB2 tables being accessed, and the plans were rebound. As a result of these changes, the entire job completed in ~3 hours instead of the projected 40 days. Overall savings were calculated to be $3.45 million.

Net Savings: $3.45 million.

5. Application: COBOL batch processing application.

   Problem: The job step, which processes over 6 million input records, was running for 7 hours and using over 6 hours of CPU time.

   STROBE analysis: STROBE Performance Profile reports revealed that over 96% of the CPU time was being consumed within two two-byte sections of a user program. Several changes were made to reduce CPU consumption, including changing the sequence of data input records, and changing program statements to use more efficient instructions. Because of these changes, elapsed time was reduced by over 6 hours and CPU time decreased by nearly 6 hours. Annual savings were estimated at $2.1 million.

Net Savings: $2.1 million annually.

Real Believers

Tom Follen of Medical Mutual of Ohio sent in the following notes.

"Performance Tuning is not Dead (at least not here).

"Listed below are some examples...

1. Batch tuning. A person on my staff won an award from STROBE for some batch tuning examples.

   Summary: Used STROBE to analyze a batch job that processed IMS databases. The job was running **14+ hours** and not completing.
   Changes: Restructured the PSB to access only the segments needed, added VSAM buffers, added Hiperspace buffers.
   Result: Job completed in **26 minutes**.

2. Online example. A new application went into production. It was a client server application that used the mainframe as the data server. It accessed DB2 through CICS. Application logic resides on the Server. At one point the CICS/DB2 piece used **50+ more MIPS than projected**. This swing occurred literally overnight.

   Tools used: STROBE and SAS/MXG. We regularly track DB2 accounting/trace data and used this to isolate the problem DB2 plans. (Sum DB2 trace data by plan for a day or hour and sort in decreasing order by CPU time or GETS. This gives the worst plans so you can attack them in order of worst offenders first).
   Changes: Combination of application logic changes (actually this is an IEF generated application so we had to make appropriate model changes). Also added DB2 indexes as needed.
   Result: Within 2-3 days were able to buy back most of the 50 MIPS and subsequent tuning put the application on track."

Net Savings: 50 MIPS daily.

DB Conversion

Norman Hollander of Southern California Edison contributed this neat savings:

"During a large database conversion project (IDMS to DB2) and application platform upgrade, I had a chance to find a few opportunities.

'During the Q/A process, it was determined that this upgrade was going to take 43 hours per day to complete. It didn't take rocket science to point out that there really was only 8 hours available per day. Batch started around 9 pm, and the on-lines needed to be up by 5 am (for east coast offices). It was further decided that Saturdays would be used for catch-up processing, and that the on-lines may not be available on the weekends. Previously it was available for 8 hours on Saturday for each coast.

"Dividing the on-line files into east coast and west coast versions, and some other scheduling processes got the batch stream down to around 11 hours. But no buffer, 'just in case'.

"Using Strobe, I discovered a few things:

- Along with DB2, there were a bunch of VSAM files (no RLS yet) used for history-type and customer-type records. Strobe pointed to much wait time and CPU time used in Catalog and Open/Close/EOV. The application was opening a file, reading a record, updating a record, and then closing it. Plus, NO buffering, or Batch LSR. Needless to say that fixing the application code to NOT do all of the file management, defining the right BUFNI and BUFND for VSAM files, defining the right number of BUFNO for QSAM tape files, and putting the update files into BLSR with some hefty buffering, was a significant help.

  Specifically, 13 hours down to 3 hours and 23 minutes on a nightly basis. This enabled the application on-lines availability on Saturday (the return of 8 hours). It was estimated that the potential loss of revenue was $75,000 per hour. That's $600,000 per week, or $31.2 million per year. Even if the estimate was off by 50% of the value of the on-line on Saturday, the tuning effort was by far worth it.

  This was at a large Savings and Loan about 5 years ago. 1994 dollars are probably far cheaper than 1999 dollars.

Net Savings: $31.2 million annually.

- On top of this throughput issue, we had a DASD opportunity. This same application was estimated to need a terabyte of EMC DASD (estimated cost from 1994 was $1.2 million for 2 boxes). From Strobe reports there was an oddity on how the VSAM files were allocated. The Record Sizes were set at 2048 with a CISize of 4096. The applications didn't take into consideration the 10 bytes that VSAM needs in the CI, so about half of each CI was wasted. Reducing the record size by 10 bytes (the 2048 was there for future functions - they were only using 1200 bytes per record), reduced the DASD need by 1/2 (around $600,000), and we no longer needed 2 EMC boxes."

Net Savings: $600,000.

CICS Improvement

René Bélanger of DaimlerChrysler was running out of capacity due to a fifty percent growth in some CICS regions, plus a new application was coming onto the system. They used Strobe to tune CICS regions and found 35 opportunities to tune their regions. They started in on a series of changes, measuring as they went. They were able to reduce CPU utilization by 5% the first week and 15% within three months. Their prime shift capacity tripled and TSO response time was reduced from 1.3 to .15 seconds. During that time, CICS usage was cut in half even at high transaction volumes. They were able to delay a CPU upgrade by 4 months saving $820,000 plus software costs (over $1,000,000 total).

The savings were achieved by a combination of the following changes:

- Tuned DB2 - Improved efficiency of SQL calls within the application; Added alternate indexes where appropriate.

- Tuned VSAM - created new LSR pools in one region to segregate files; Adjusted the overall LSR buffer pool allocations; Added more strings to individual files where appropriate.

- Tuned CICS – Created long-running mirror tasks to gain performance benefits associated with re-using CSMI tasks (the CICS mirror tasks) cross-region. Within one region, approximately 90% of the tasks are CSMI tasks.

Net Savings: $1,000,000.

How Do You Justify It?

As you can see from these examples, a very small change can produce significant savings. Also, the ability to quantify the amount of savings makes a very powerful and compelling justification for future projects. You can use the savings to justify the cost of a full-time performance analyst or capacity planner, education for them in the way of classes or conferences, or tools, such as Strobe (or a similar competitive product called InTune from BMC). Or a subscription to the TUNING Letter! One person used one performance tip from our newsletter to justify its purchase for the next 2500 years!

The point is, you need to be able to justify your time with a dollar amount. First, identify all of the types of savings: CPU time, DASD, elapsed time, staff costs, etc. and then apply a dollar amount to them. If you required prime time computer resources to complete your project, you may need to reduce the savings by the cost of the project itself. Determine how many months the savings applies to - a month, six months, two years? Document this and let management know about it. This simple step could help you get approval for the next projects you want to tackle. Most of the time you'll be very surprised yourself at the savings.

Don't think that a small change is worthless. Another example from **Norman Hollander**: "One CICS transaction that was tuned only saved fifteen seconds from the original. Except that 15-20 tellers in 500 branches executed this transaction every hour. So 20 times 500 = 10,000 transaction per hour, times 15 seconds = 150,000 seconds every hour. Over an 8 hour day, that's 5.5 hours!" Now, that's worth it!

Trust me on this. Try it once and you'll understand the power of adding a financial summary to any project.

I once tried to get my boss to put me on commission based on the savings I could achieve. He was smart enough to turn me down. Otherwise I could have retired at 30!

The Future

So what does the future hold for today's performance analyst and capacity planner? Change, and more work than you can handle!

Continued Tuning

First, in most installations, the amount of work on MVS is increasing. That means a continuation of tuning and optimizing the current systems that are running on MVS. It helps to understand the basics of tuning so you can easily respond to any emergencies as they arise. Most people keep a set of known techniques they can try at any time.

Here's the list from **Norman Hollander** of Southern California Edison's bag of tricks:

- "Buffering wand for VSAM and non-VSAM ('the best I/O is NO I/O')
- "Storage Pixie Dust for DB2 applications (so many shops scrimp on a fairly cheap resource)
- "IDCAMs Incantations (With all the options, the obvious is bound to go wrong with CI-this, Free-space-that; and isn't AIX another OS?)
- "Magic Strobe (If you're trying to boost Main Frame Vendor stock, let the application keep opening and closing files, or calling for the date every 30 milliseconds)
- "Miscellaneous Monitors (You'd be surprised by how fast a system can wait when DB2 has a bottleneck, or MQ keeps looking for its transaction queue in the wrong place)"

Notice Norman's first item - buffers. This is one of the easiest and most effective ways to tune any type of work, from batch to online. It reduces CPU time, elapsed time, user response time, and storage occupancy. My favorite way to tune!

One of the things you may want to do is to identify what tools you already have available for analysis. Even products like charge back systems can give

you invaluable reports for identifying important jobs to tune.

I'm a real believer in "Top xx Lists." 'xx' can be 10, 50 or 100, depending on how much time you have available. I would look for jobs taking the most CPU time or the most I/Os or the most tapes. These are always good candidates for tuning. Another good list is the highest CPU or highest activity transactions in your favorite online subsystem.

What has changed in traditional tuning is the lack of usefulness of bit-twiddling. Changing a few lines of code to save .01% of the runtime may not be worth it. Changing the size of a record to save a track of disk space definitely isn't worth it. It's too bad that wasn't true when we abbreviated our year fields to two digits!

Because of the lower prices for hardware, the reductions must be more substantial to be able to justify the effort to reduce the resources. I really believe that it is far better to buy more memory than to mess around with reducing memory usage. You might tell me that your machine is too old and the storage is maxed out (i.e. you've added all you can and are still out of memory). If that's true, I would recommend that you look into the newer CMOS machines that can often be had for less than you're paying today for maintenance and environmentals on that old machine. Look into it - you'll be surprised.

Here are a few other recommendations:

- Learn to avoid the common mistakes: overconfiguring LPARs, leaving debug options turned on in production, under-specifying buffers, etc.

- Run systems at 100%. The old guideline of running at 85% to provide good online response time is no longer true. OS/390 is designed to run effectively at 100%. In fact, once you go to goal mode, WLM will strive to run the system at 100%.

- Use memory for tuning your CPU - add buffers, internal sort work areas, data spaces, etc. to reduce I/O.

- Capacity planners must understand the impact of tuning changes to fully understand the available capacity on their machine.

- Both performance analysts and capacity planners should be intimately familiar with the intricacies of software pricing. Delaying a processor upgrade can save you thousands of dollars in delayed software costs.

- Reach out and touch someone - via email, local conferences, networking among your peers. One of the best ways to learn more about tuning is from others who have fought the same battles.

- Use IBM's extensive library of performance material. The IBM redbooks have the majority of performance information, although sometimes the regular manuals might have entire chapters on performance tips. The 600-page TCP/IP manual comes to mind! Research performance APARs and informational APARs. IBM's Web sites are also becoming repositories for much performance information today. The USS (UNIX System Services) pages, for example, give a large amount of performance-related information.

- Share your knowledge with others at your sites. When I used to return from a class or conference, I would put together a 1- to 3-hour presentation for the operators, one for the application programmers, and one for the sysprogs to share the neat things I found. All I ever asked was that if they tried something to let me know, so that I could quantify it for my management. This technique allowed me to get justification for others to attend the conferences as well.

- Document, document, document! Things will be ***so much*** easier if you do.

New Work

The biggest change for PM/CPers is the appearance of new workloads they will need to plan, manage, and tune. Most installations will be (have already) installed one or more of the following on their mainframes: WebServer, Java, Lotus Domino Notes, ERP applications (PeopleSoft, BaaN, SAP

R/3, etc.), ported UNIX applications, e-business, e-commerce, business intelligence (data warehousing), CORBA, etc.

These new workloads are the reason that I think education is more important today than it ever was. Whether you go to specific performance classes or attend user conferences like SHARE and CMG, training and knowledge of these new workloads will be a necessity for future performance management and capacity planning.

What's different about these applications is that they don't behave like our traditional legacy systems. They have very sporadic resource requirements and a high need for rapid response. The resource requirements can also be phenomenally large. These large new workloads will put additional pressure on your current workloads.

Even if you don't plan to put a WebServer on your S/390, you will probably end up with a WebServer on some platform which will need to access your mainframe databases, thus increasing the load on your system.

This scenario leads to even more difficulty for PM/CPers because determining response times and anticipating capacity requirements in a distributed system is much more difficult than managing them on a single system. Distributed systems will definitely keep you busy!

Reporting

More data centers are now putting their reporting on the Web, typically on their Intranet. The process can be automated. It may take a bit of effort to automate the process to put your reporting on your Intranet, but the benefits are extensive. Imagine, if you will, you management (all levels of your management) being able to log on whenever they want to look at the performance of the system for yesterday, last week, last month, or even over the last year. If they want more information about a particular system, they can drill down and get it. Presto - the reports display correctly to dozens of people at their convenience every day - automatically! I wish this had been available when I was doing data center reporting and waiting for the colored printers that printed at the speed of molasses.

I think that Intranet reporting for data center reporting, performance management, and capacity planning is one of the most interesting opportunities for PM/CPers today.

To get started on this, check out your PM/CP tools to find out what they can provide. For example, SAS's IT Service Vision comes packaged with tools to provide the reports on a Web site. Look at what others are doing. Some of the nicest work I've seen is by **Linwood Merrit** of Trigon Blue Cross Blue Shield. Lin is the project manager for EWCP at SHARE. Figure 3 and Figure 4 show two of the reports available to his management and users. His SHARE session 2560, "Web-based Reporting" should be out on the SHARE Web site in a few weeks, and includes sample code for automating the reporting process. A similar session he gave at CMG is currently available at <**http://www.cmg.org/handouts/linmerritt.zip**>. His paper can be found on the CMG proceedings. Lin even publishes his SHARE trip report on his Intranet Capacity Planning Web site for everyone to share. Neat idea!

Summary

Obviously, performance tuning is alive and well. I see too many sites where the application of well-known tuning techniques has saved millions of dollars to believe that it's simply cheaper to buy more hardware.

I do believe that bit-twiddling to get another ounce out of the machine is long-past. I also believe that to stay current in the industry, you must develop new talents and an understanding of the new workloads and new tools. ■

Figure 3 - MVS Night Batch Window

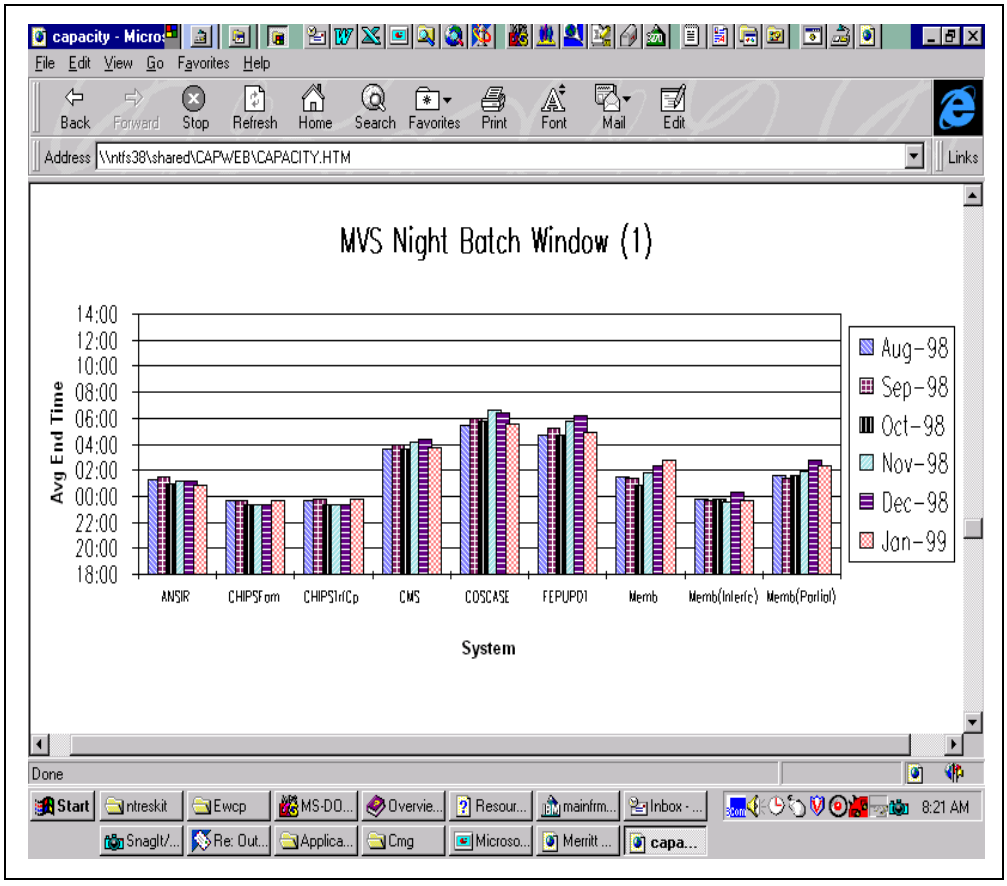

Figure 4 - MVS CPU by Day