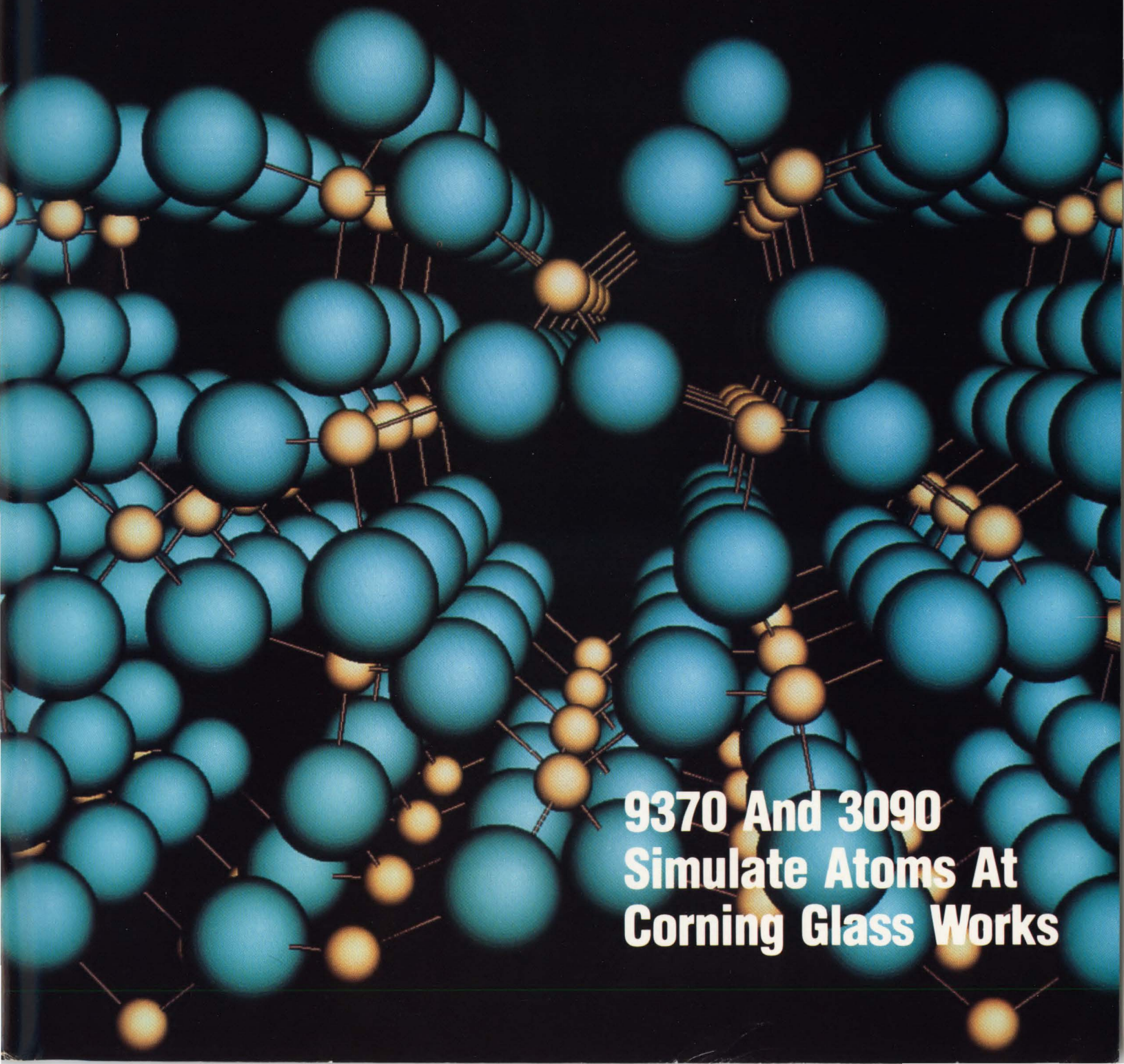
MAINFRAME JOURNAL

For Users of IBM System/370 Architecture & Compatible Systems

February 1989



9370 And 3090 Simulate Atoms At Corning Glass Works

VMCENTER II
Smart Economics.

For the new generation, a new choice in VM systems management.

For a company the size of PepsiCo, it takes a lot to keep the sparkle in your data processing operations. From Pepsi-Cola to Taco Bell, growing user demands have made strong central systems management essential.

Fortunately, as the challenge has grown, so has PepsiCo's ability to meet it—thanks to VM Software and the capabilities of VMCENTER II.

The results: a new performance standard for VM operations. A new level of convenience for data center staff. And a new ability to manage for the long term—knowing the short term is under control.

LEARNING THE VALUE OF RELIABILITY.
At one time, the company relied extensively on in-house software for data center management. Saving some money up front. But quickly paying the price in flexibility, reliability, and peace of mind.

Today, more and more operations are under VMCENTER control. And the results are apparent in everything from backup and tape management to system scheduling and accounting. Performance and reliability are up. Operator errors and administrative headaches are down. While problem resolution is faster, easier, and more accurate than ever before.

VMCENTER II. *THE* VM SYSTEMS MANAGEMENT TOOL FOR THE FUTURE.
This performance is impressive. But it's only the beginning of what VMCENTER II can do. From comprehensive management reports to the automated handling of a broad range of everyday operations, VMCENTER II does more for VM systems management than any other product on the market. Which is why it's the one essential complement to *all* your VM systems—from 9370 to 3090 to whatever the future may bring.

VMCENTER II. Standard-bearer for Pepsi's new generation. Smart solution for today. For more information, write or call:

800-562-7100
703-264-8000

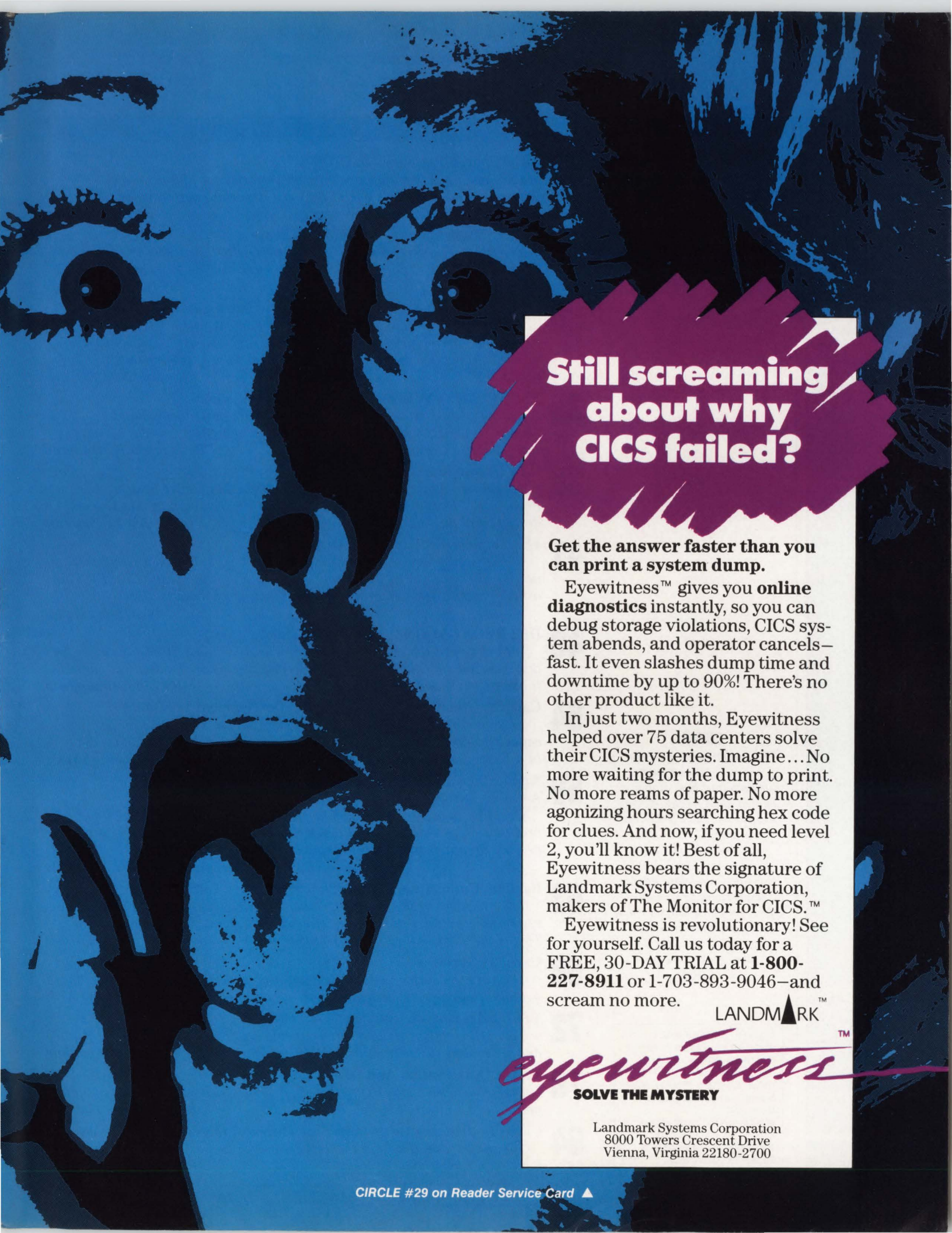
VM Software, Inc.
1800 Alexander Bell Drive
Reston, Virginia 22091

VM SOFTWARE INC.

The VM Experts

© 1987 VM Software, Inc.
1-MFJ-8902

CIRCLE #20 on Reader Service Card ▲



Still screaming about why CICS failed?

Get the answer faster than you can print a system dump.

Eyewitness™ gives you **online diagnostics** instantly, so you can debug storage violations, CICS system abends, and operator cancels—fast. It even slashes dump time and downtime by up to 90%! There's no other product like it.

In just two months, Eyewitness helped over 75 data centers solve their CICS mysteries. Imagine…No more waiting for the dump to print. No more reams of paper. No more agonizing hours searching hex code for clues. And now, if you need level 2, you'll know it! Best of all, Eyewitness bears the signature of Landmark Systems Corporation, makers of The Monitor for CICS.™

Eyewitness is revolutionary! See for yourself. Call us today for a FREE, 30-DAY TRIAL at **1-800-227-8911** or 1-703-893-9046—and scream no more.

LANDMARK™

eyewitness™

SOLVE THE MYSTERY

Landmark Systems Corporation
8000 Towers Crescent Drive
Vienna, Virginia 22180-2700

CIRCLE #29 on Reader Service Card ▲

CONTENTS

DEPARTMENTS

6 Editor's Comments

8 Tech Advisor

96 Product Update

98 Vendor Profile

FEATURES

13 **Local Shared Resources In CICS**
LSR has a positive influence on both performance and resource utilization and as such is one of the most important performance tools available in a CICS environment. ***By Ted C. Keller***

22 **Why COBOL Efficiency Still Makes Sense**
Techniques enabling an IBM mainframe COBOL program to run faster are presented with coding suggestions to motivate you to examine new techniques. ***By Harvey Bookman***

30 **DB2's Query Management Facility: One User's Experience**
The author's mission when she joined Diamond Shamrock Chemicals was to provide micro and mainframe user tools and support to complement existing production systems. QMF was found to be a user-friendly product. ***By Mary Jo Cater***

34 **ES/9370 And VM Generate Computer Simulations**
Corning Glass Works gains a cutting edge in glass modeling by combining the ES/9370 and the 3090 to simulate effects on the substances that comprise Corning's products. ***By Stephen H. Quanrud***

36 **ISPF Techniques**
Techniques that aid in making the use of the Interactive System Productivity Facility/Program Development Facility more productive are described in this article. ***By Jon E. Pearkins***

41 **ESA Implements Expanded Addressing**
Enterprise Systems Architecture implements expanded addressing through new facilities known as the Advanced Address Space Facility (AASF). This article presents more information about several of these facilities. ***By Michael Haupt***

48 **DB2 REORGS? Don't Waste Your Time!**
IBM designed the DB2 REORG utility to correct the problems caused by disorganized data. Read this article to discover what this utility is and how it can be improved. ***By Bill Cunningham***

54 **Capacity Planning: The Applications Development Viewpoint**
This author feels applications developers should contribute to the capacity planning process by finding and accounting for NBUs that relate in a linear fashion to computer resource consumption. ***By Michael Snyder***

58 **Applications Development:**
A Guide Through The Maze Of Tools And Methods
Organizations, not just system development managers, need to adopt procedures for weighing each new development tool's promise against each project's realities and for predicting obsolescence. ***By Robert Lorentzen and Paul Tinnirello***

63 **Banking Consortium Solves Disaster Recovery Dilemma**
The president of the data processing subsidiary of a $1.8 billion bank is responsible for smooth handling of data regardless of the circumstances. This is the story of how he did just that. ***By Beth Ellyn Rosenthal***

68 **Capacity Surfaces**
The capacity surface offers the analyst a vehicle for communicating large quantities of complex modeling results to decision makers. ***By H. Pat Artis***

72 **VSE Under The Covers: The Dispatcher And Scheduling System**
This article offers an in-depth explanation of the supervisor's interrupt and scheduling mechanism in the VSE system. ***By Eric L. Vaughan***

78 **VSAM Optimization And Design**
Free space selection, I/O buffer allocation, space allocation and index options are addressed in this article. ***By Eugene S. Hudders***

84 **Capacity Planning Using Statistical Theory And Performance Data**
The theories presented in this article are the basic mathematical premise for a total tuning philosophy. ***By Philip C. Davis, III***

February 1989

Volume IV Number 2

COVER:

The graphics on the cover represent a computer simulation of how atoms of molten glass might appear when cool. The ES/9370 at Corning Glass Works generated the model in partnership with a 3090. Computer modeling allows researchers to simulate effects on the substances that comprise their products. See article on page 34. Photograph courtesy of IBM.

MAINFRAME JOURNAL© (ISSN 0895-5751) is published monthly by Thomas Publications, Inc., 10935 Estate Lane, Suite 375, Dallas, TX 75238, (214) 343-3717. Second class postage paid at Dallas, TX.
POSTMASTER: Send address changes to: MAINFRAME JOURNAL, P.O. Box 551628, Dallas, TX 75355-1628.

CACHE MAGIC™

With Cache Magic, improved VM system performance has never been easier. Enjoy a dramatic reduction in batch run times, in some cases by as much as 50%! Accelerate online response times, and complete more transactions in the same, or less time—all without the cost of additional hardware!

Cache Magic works its magic by storing your most active DASD files in an in-memory cache. Here they are accessed at processor speed, effectively eliminating I/O bottlenecks. And because Cache Magic automatically determines which files should be cached, performance gains are guaranteed, without investment of time and personnel.

Why not spend ten minutes with one of our representatives to find out what amazing things Cache Magic can do for you? All you have to lose are batch overruns, user complaints, and expensive hardware upgrades.

SDI®

20 Years of Software Innovation
1 (800) 537-1969 (415) 572-1200
In Europe: +31 10 436 12 77
1700 South El Camino Real, San Mateo, CA 94402

CIRCLE #76 on Reader Service Card ▲

EDITOR'S COMMENTS

Carol M. Hoag

December was the month of accidents of the bone-breaking, muscle-wrenching kind. First my eight-year-old son injured his knee and pulled the hamstring of his left leg at a roller skating party. The pediatrician sent him to the orthopedic surgeon and while no bones were broken, he was on crutches for a while.

Then, my teenage son was running backward during an athletic drill, hit a wet spot and started to fall. Thinking he could break his fall, he put his hand out. He broke his fall and a bone in his hand.

Sometime in between my sons' accidents, I sprained my ankle. Rather, I *thought* I sprained my ankle. It hurt like crazy when the accident happened, but I thought it was just bruised. The first inkling that it was more than a bruise came about two-and-a-half weeks later when I realized the injury was not getting any better. In fact, it was worse. You guessed it. I have a hairline fracture of my left ankle bone.

The only way I have peace of mind about such a series of bizarre accidents is because I have insurance. With insurance comes the knowledge that when an accident happens, I will be able to get the proper attention and institute a recovery plan so that I will be "up and running" in no time.

Insurance is to an accident as a recovery plan is to a disaster.

A disaster might be natural such as a fire, flood, earthquake or tornado. Also, it could be the result of a terrorist bomb. Closer to home, a downed computer could spell D-I-S-A-S-T-E-R.

An adequate recovery plan can restore computer services within 48-hours. It would involve two emergency locations provided to recover the computer system. First there is the fully-equipped disaster recovery facility from which regular daily computer processing runs. It usually is available for around six weeks. At this facility, the staff would load the operating system so it would be "up and running" when the subscriber to the plan arrives to load the application.

If recovery time is longer than six weeks, the plan provides for a shell facility in which computer operations and support staff would be available for about six months.

One subscriber to a fully-equipped disaster recovery facility described it as "sleep insurance." Two excellent reasons for being a subscriber to such a facility are to maintain the company's operation and its computer base.

In this issue is a story starting on page 63 about a banking consortium and how it solved its disaster recovery dilemma. It may be the impetus you need to urge your company to look into a recovery plan. In the event of an accident or disaster, we all need a plan to get us "up and running."

This issue also covers another area of planning — capacity planning. Capacity planning from the application development viewpoint starts on page 54 and is written by Michael Snyder. Phil Davis writes about capacity planning using statistical theory and performance data starting on page 84. We are particularly pleased to welcome Pat Artis back as he tackles the problem faced by performance analysts of how to effectively convey the results of their systems' performance on page 68. Planning pays off!

A membership that more than pays for itself in technical information passed on to users is one with GUIDE International. We have had an overwhelming response to Pete Clark's article, "GUIDE International: A User's Perspective," that appeared in the January, 1989 issue from readers wanting information about how to become a member of GUIDE. To do this, contact Margaret Miller at GUIDE International Corporation, 111 East Wacker Dr., Suite 600, Chicago, IL 60601, (312) 644-6610 Ext. 3216.

Carol M. Hoag

MAINFRAME JOURNAL
For Users of IBM System/370 Architecture & Compatible Systems

✣ ✣

Publisher and Editor-in-Chief
Robert H. Thomas

Associate Publisher
Martha Thomas

Editor
Carol M. Hoag

Copy Editors
Judy Beller
Pat Warner

Art Director
David Kramer

Assistant Art Director
Ken Buerer

Advertising Manager
Denise Thomas

Marketing Services
Sally Webb

Circulation Manager
Janice Porter

Assistant Circulation Manager
Nancy Crawford

Administrative Manager
Marian Davenport

Typography Coordinator
Dub Pope
Jaggars Chiles Stovall

Printing Coordinator
Steve Priester
American Signature Graphics

BPA Membership Applied For April, 1988.

SUBSCRIPTION RATES & INQUIRIES: Subscriptions are free within the USA and Canada. One-year foreign subscriptions are $96.

All subscriptions, remittances, requests and changes of address should be sent to MAINFRAME JOURNAL at 10935 Estate Lane, Suite 375, Dallas, TX 75238, (214) 343-3717.

MAINFRAME JOURNAL© is copyrighted 1989. All rights are reserved. Reproductions in whole or part prohibited except by permission in writing.

Why We're Betting a Million Lines of Code on the SAS/C™ Compiler.

At SAS Institute Inc., we've invested more than 10 years of research—and over a million lines of code—in the SAS® System, the world's leading data analysis software. So you can bet we left nothing to chance when we chose the C language for the next generation of our software.

We selected C for the portability it would bring to the SAS System, but weren't about to risk our code on just **any** mainframe C compiler. So we tried them all. When none could meet our exacting requirements, we created our own: the SAS/C compiler.

We Developed It.

Support It. Use It.

The SAS/C compiler set new standards for efficiency and technical quality, with:

- A source-level debugger that includes structure display, ABEND recovery, and debugger I/O exits for debugging specialized applications
- Reentrant object code
- Highly optimized generated code
- Use of standard IBM linkage conventions, with support for 31-bit addressing
- A CMS Rexx/TSO CLIST interface
- Support for signal handling including program checks and terminal interrupts, and non-standard signals such as timer interrupts and stack overflow
- Many built-in functions including string handling
- In-line assembler.

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA. SAS/C is a trademark of SAS Institute.

Copyright © 1987 by SAS Institute Inc.
Printed in the U.S.A.

And when we combined these features with outstanding technical support and frequent updates—both provided free—software developers everywhere took notice. The SAS/C compiler is now the market leader, installed in hundreds of commercial firms and academic institutions.

Test It. Compare It.

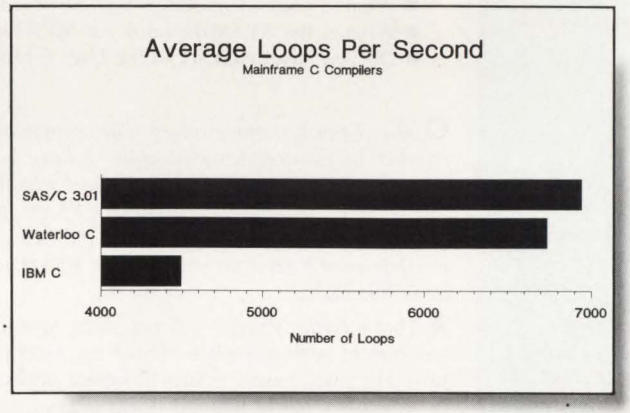
FREE for 30 Days.

We're betting you've set the same high standards. That's why we'd like to send you the SAS/C compiler, under OS or CMS, for a free 30-day evaluation. We'll also send you a free copy of a leading benchmark program. Compare our compiler with any other. Odds are, you'll choose the SAS/C compiler.

Just mail the coupon below. Or call your Software Sales Representative at (919) 467-8000.

SAS Institute Inc.
SAS Circle ☐ Box 8000
Cary, NC 27512-8000
Phone (919) 467-8000
Fax (919) 469-3737

Using a C version of the Dhrystone benchmark, the latest SAS/C compiler release produces the fastest code among the top 3 mainframe compilers. It even tops our own previous release by 35%.



I'd like to put the SAS/C™ compiler to the test with a free 30-day trial, and my free copy of the Dhrystone benchmark program. Give me the details.

Please complete, or attach your business card.

Name _____ Title _____

Company _____

Address _____

City _____ State _____ ZIP _____

Telephone _____

Mail to: SAS Institute Inc., Attn: CC, SAS Circle, Box 8000, Cary, NC, USA, 27512-8000

TECH ADVISOR

Q *I've inherited a system in which Generation Dataset Groups (GDGs) were created in an uncontrolled manner. Is there a way to inquire what GDGs are in the system now, the retention period and when the GDG was defined?*

A Getting the information you need is strictly a manual process. First, do an IDCAMS LISTCAT USERCATALOG to get the name of all of your user catalogs. Then do a LISTCAT ALL CATALOG (usercatalogname) for each of your user catalogs. This will give you the name, creation date and LIMIT (number of generations) of the GDGs.

Q *We run DOS/VSE as a virtual machine. Whenever someone leaves our IBM 3287 printer not ready, we get message DMKDID 546I. If not corrected by readying the printer, it causes us to have to re-IPL DOS/VSE. Any ideas?*

A From the information we have available to us, I am assuming your TP-monitor (CICS or something else) appears to be hung up on a missing interrupt from the 3287 that in turn hangs your entire system. (This type of error was common under BTAM by the way.) I would suggest looking at the SYSMIH value. Either set it to zero to ignore the missing interrupt or set the value high enough to allow you to ready the printer before the error occurs. Also, check to see if you have the MIH option specified in your user directory entry. If you do not have it specified, CP will not simulate an interrupt to the guest when a missing interrupt is detected.

We assumed some information in answering this question. Some additional information you may need to find the solution to this problem is:

- What is your release of VM?
- What is your release of VSE?
- What is the entire message?
  DMKDID546I has the following content:
  DMKDID546I Interruption (Pending/Cleared); (Device/Control Unit) RDEV, CSW csw, USERID userid
- What is your TP monitor, BTAM or VTAM?
- What is the SYSMIH value for MISC set to?
- Do you have MIH set in the User directory entry?

Q *May I put freespace on an index component to reduce the number of Control Interval splits I have been getting? I'm running MVS/XA using DFP 2.3 and I would like to cut down the number of splits on the index of a VSAM file that gets heavy inserts. I tried putting freespace on the index component when I redefined the file, but it looks like VSAM will not let me do it. Is this right?*

A That's right. VSAM will not allow you to define an index component with freespace. There are a couple of reasons for this. The main reason is that freespace applies to a percentage of an entire Control Interval. Each Index Control Interval contains just one record that VSAM makes seven bytes less than the size of the Index CI. VSAM will not reserve part of an Index Control Interval because then it could never fit an index record into one.

The other reason is that if VSAM allowed freespace on an Index record, then part of each Data Control Area referenced by the Index could not be loaded with data. You would in effect be applying freespace to each Data Control Area inadvertently. It is preferable to request the data freespace directly.

Instead of using freespace, we recommend expanding the size of each Index Control Interval when you define the file. Beginning with DFP 2.2, VSAM allows Index CI sizes up to 32K. (For VSE, you can go up to 8,192 for the index.) Simply specify an Index Control Interval size larger than the default value VSAM selected for you. Expanding the Index CI should reduce the splits and may also help with key compression problems and reduce the total number of index records and levels.

The disadvantages are that each index record will use more virtual storage when read into memory, rotational delay and I/O transfer times will increase and your new index size may not match available CICS LSR buffer pool sizes.

Q *What are IBM's plans for VM? What enhancements will be added in future releases?*

A I am sure there are many data processing professionals who have this same question on their minds. IBM has a way of keeping these things secret until IBM is ready to let us know about them. The best I can do is pass on to you some information and speculation that has passed over my desk or things I have heard at user groups.

Some of the things you could look for in the future are the following.

- The CLASS OVERRIDE table may be dynamically changeable in the future.
- A disposition parameter for the VM print queue similar to the VSE POWER parameter 'DISP=K' may be introduced in the future.
- A maintenance history file has been announced for VM/SP R6. This will be similar to the MSHP history file in VSE. This has been long in coming but will be a nice feature for VM.
- VM may be given the ability to cold start without clearing the queue areas.
- Look for more products to utilize GCS. Networking tools for VM could be big in the future also. As we all know, networking of different computers will be the challenge of tomorrow.

I would ask your IBM representative if he/she knows anything that can be passed on to you. Sometimes the representatives hear things that can be passed on to a customer.

Q *We use VM/SP Release 4. There are several programs available for CMS Release 5 (University of Waterloo) that were very useful. I need to convert these programs to SP. Has anyone already done these conversions? If so, where can I get copies? Alternatively, I need better information that IBM provides on the differences between the old CMS file system and the new.*

A The best way to find out if anyone has already converted these programs would be to contact the University of Waterloo or the program's author. Since they distributed the program tape, they may know if someone has already converted the programs you are interested in. It is our understanding that the University of Waterloo tapes contain the names and addresses of the program authors and the University of Waterloo contacts.

With regard to the second part of your question about the differences between the old CMS file system and the new, we believe IBM will be your best source of information on these changes.

If you decide to change the programs yourself, you should examine the source of these programs and determine if the author used standard VM calls and parameters or if he hard-coded the process. If standard calls and macros were used, you should be able to reassemble the programs using the VM Release 4 assembler. Should the programs contain non-standard calls, then you will have to rewrite the modules so they work on your system. The program author may be able to assist you if a rewrite is needed.

Questions are answered by consultants and instructors from Davis, Thomas & Associates (Minneapolis, MN), the largest technical services firm in the upper midwest. Please address your technical questions to: The Tech Advisor, MAINFRAME JOURNAL, PO Box 551628, Dallas, TX 75355-1628.

SYSTEMS SOFTWARE FOR VM/VSE DATA CENTERS:

Only one company covers the two, completely.

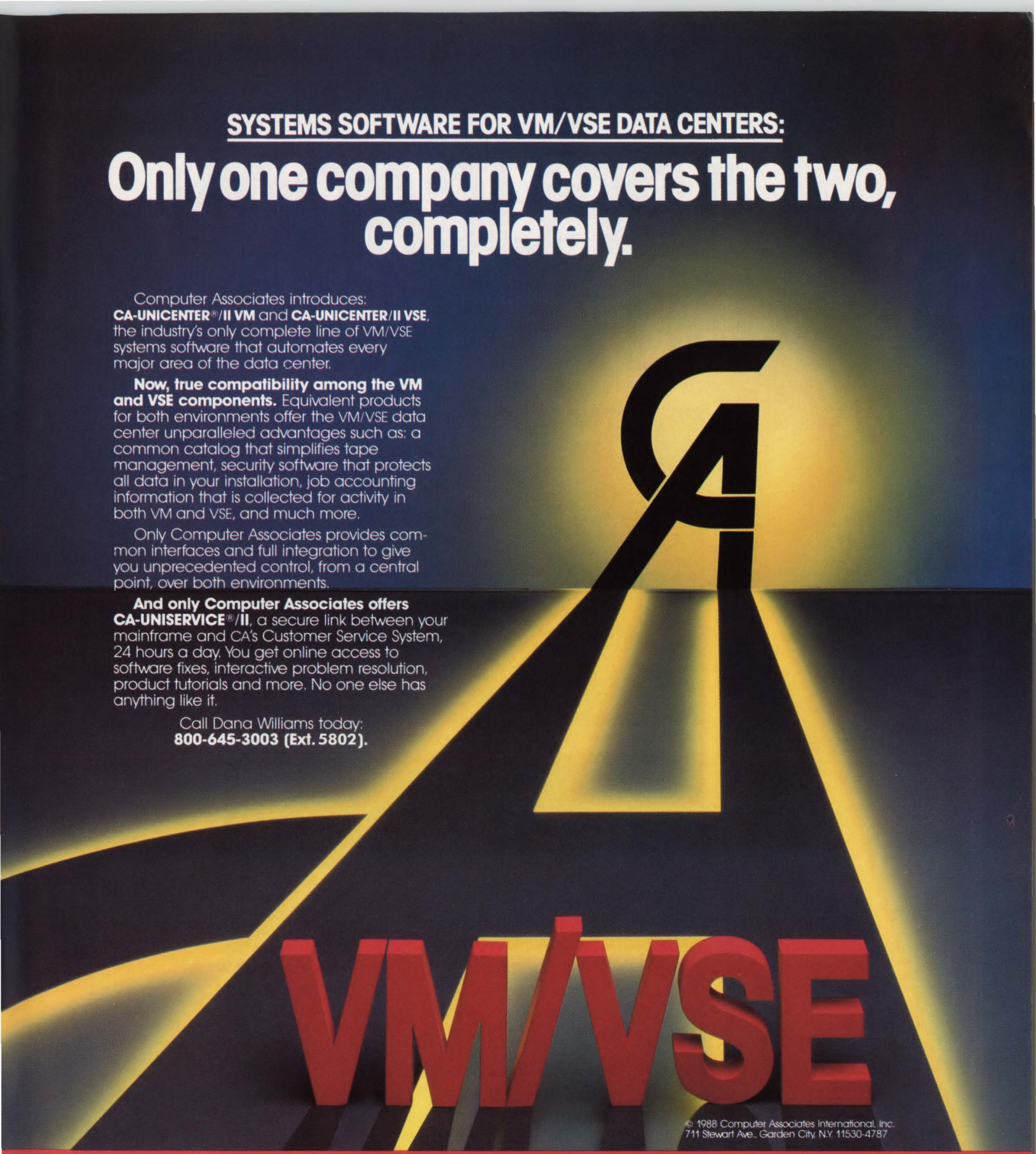
Computer Associates introduces: **CA-UNICENTER®/II VM** and **CA-UNICENTER/II VSE**, the industry's only complete line of VM/VSE systems software that automates every major area of the data center.

Now, true compatibility among the VM and VSE components. Equivalent products for both environments offer the VM/VSE data center unparalleled advantages such as: a common catalog that simplifies tape management, security software that protects all data in your installation, job accounting information that is collected for activity in both VM and VSE, and much more.

Only Computer Associates provides common interfaces and full integration to give you unprecedented control, from a central point, over both environments.

And only Computer Associates offers CA-UNISERVICE®/II, a secure link between your mainframe and CA's Customer Service System, 24 hours a day. You get online access to software fixes, interactive problem resolution, product tutorials and more. No one else has anything like it.

Call Dana Williams today:
800-645-3003 (Ext. 5802).



© 1988 Computer Associates International, Inc.
711 Stewart Ave., Garden City, N.Y. 11530-4787

COMPUTER ASSOCIATES™
Software superior by design.

- World's leading independent software company.
- Broad range of integrated business and data processing software for mainframe, mid-range and micro computers.
- Worldwide service and support network of more than 100 offices.

Resource & Operations Management • Financial • Banking • Graphics • Spreadsheets • Project Management

CIRCLE #105 on Reader Service Card ▲

APPLICATION INTEGRATION

Windowing is one way to integrate your applications with TPX.

End users can also benefit from interacting with multiple applications on a single, customized screen.

ONLY TPX MAKES IT THIS EASY

Session managers allow users to switch among applications, but what do users do when they need information from two or more applications at the same time? TPX, the world's leading session manager, now sets a new standard for session management by providing Application Integration—easily.

Many times every day your users need to access and update information from multiple applications concurrently. For instance, a service representative using a Customer Information application under IMS sees a customer's name, address and a summary of accounts. While viewing this data, the service representative needs to access detailed information from the Account History application running under CICS. Think of the increased efficiency and improvement in customer service if information from two or more related applications could be combined on a single, interactive screen.

TPX gives you two ways to integrate applications: windows and customized multi-application screens for end users. With windows, you can divide the terminal screen any way you want and easily interact with any session, as well as jump, scroll, zoom and do cut and paste between sessions. If that's not enough, you can customize your own screens by combining data from any number of applications and give your users a unified, standard view no matter what the existing application screens look like.

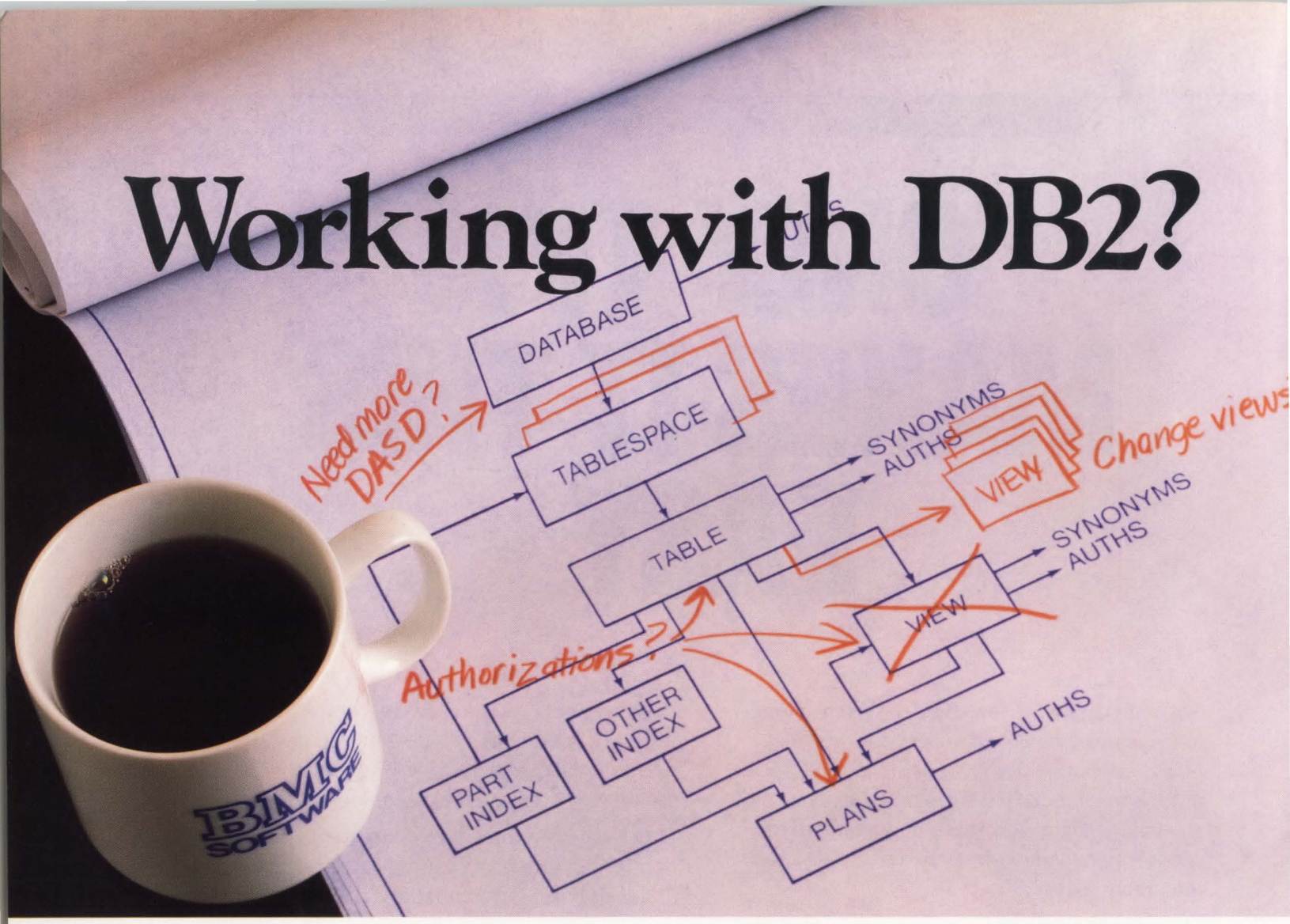
Either way, TPX integrates concurrent applications easily, increasing your users' productivity and your business' competitive advantage.

Return the attached reply card today for more information on how TPX 2.0 can help you integrate your applications in an MVS or VM environment. If someone else already returned the card, feel free to call **(800) 323-2600 ext 444** or **(412) 323-2600 ext 444** in PA.

DUQUESNE SYSTEMS
Two Allegheny Center
Pittsburgh, PA 15212

CIRCLE #1 on Reader Service Card ▲

Working with DB2?



Work with BMC

DB2 can be a lot more work than you expected with quite a bit less help than you need. But when you've got BMC Software's comprehensive set of data base administration tools—which include standard interfaces and integrated function—you can reduce your costs and make your work fast, easy and error-free.

DB2 ALTER—provides complete support for changing, copying and migrating DB2 data structures; includes data conversions, authorization-id switching and restart capabilities.

DB2 CATALOG MANAGER—gives quick and easy catalog information, execution of SQL DDL and DB2 utilities, audit logs and extended SQL function.

DB2 DASD MANAGER—controls the life cycle of physical objects with comprehensive space analysis statistics; also includes space estimation, AMS command and utility jobstream generation and action triggers.

DATA PACKER™/DB2—reduces DASD requirements for DB2 tables an average of 50% to 70%; reduces EXCPs.

DB2 REORG PLUS—reorganizes DB2 tables 4-10 times faster than the supplied DB2 utility; provides dual image copy and statistical history.

For more information or to begin a 30-Day-Plus Free Trial of any or all of these products, complete and return this coupon or call BMC Software, Inc., **The Complete DB2 Company.™**

BMC Software, Inc.
P.O. Box 2002
Sugar Land, TX 77487-2002

1-800-841-2031

or call collect: 713-240-8800

BMC SOFTWARE

Contact me about:

Free Trial	More Info	
☐	☐	DB2 ALTER
☐	☐	DB2 CATALOG MANAGER
☐	☐	DB2 DASD MANAGER
☐	☐	DATA PACKER™/DB2
☐	☐	DB2 REORG PLUS
☐	☐	All BMC DB2 products

Name _____

Title _____

Company _____

Address _____

City _____ State/Prov. _____ Zip/P.C. _____

Phone _____

MJ

CIRCLE #103 on Reader Service Card ▲

Local Shared Resources In CICS

LSR has a positive influence on both performance and resource utilization.

By Ted C. Keller

One of the most significant performance tools available in a CICS environment is VSAM Local Shared Resources (LSR). LSR provides services that can allow savings of CPU processing, virtual storage utilization, working set size and physical I/Os. Many tuning techniques require trade-offs between resources — you often need to spend more of one resource to reduce utilization of others.

LSR, compared to Non-Shared Resources (NSR), can allow for improvements in performance as well as reductions in usage of all these resources simultaneously. In performance terms, this is one of the few true ''win-win'' situations. In most cases, LSR has a positive influence on both performance and resource utilization.

LSR and NSR are the two access techniques commonly used with VSAM. NSR is more common and is available without any special support. It is used primarily by batch programs and other less sophisticated systems. LSR is more complex and its use is usually limited to systems providing this special support. CICS provides support for both LSR and NSR. For CICS Releases 1.7 and beyond, LSR is used by the default.

NSR Processing

When a VSAM Keyed Sequence Dataset (KSDS) file is accessed *using NSR,* VSAM will need to read the highest level of index, any additional levels of index, the sequence set (which is the lowest level index) and finally the Control Interval (CI) containing the data. VSAM will reserve one index buffer and one data buffer for each string defined for the file. It will also reserve an extra data buffer for internal uses. If more index buffers are allocated than VSAM strings, VSAM will use these extra index buffers to hold the highest level index CI and additional high-level index CIs (if any).

On subsequent I/O operations, VSAM will look at the CIs in these buffers before attempting to reaccess high-level index records from the file. When only one additional buffer is allocated, it will be used to hold the highest level index record. Most small- to medium-size VSAM files have only this single high-level index record. Larger files may have additional levels of index and can benefit from additional index buffers.

When a sufficient number of index buffers are allocated with NSR files, the physical I/Os to read higher level indexes can be all but eliminated. Accessing data on the file will normally consist of locating the higher level indexes in memory, reading the sequence set CI from DASD and reading the data CI. If the specific index buffer that is allocated to the VSAM string assigned to read the sequence set CI *happens* to contain the correct CI (it was left there from the previous access *by that string*), the CI will be used without

reaccessing it from DASD. Similarly, if the specific data buffer that is allocated to the string assigned to read the data CI *happens* to contain the correct data, a read operation will be bypassed.

In most environments, the number of times the sequence set or data CIs will be found in the correct buffer will be so small that it is not worth considering. Furthermore, if no extra index buffers are allocated, each level of index will be read one at a time into the same buffer thereby guaranteeing that no I/O can be saved. Typically, additional index buffers will be allocated and each read access to a VSAM KSDS using NSR will involve two I/O operations — one to read the sequence set CI and one to read the data CI.

LSR Processing

When accessing VSAM files *via LSR,* one or more *pools* of buffers will be created. In most cases, several files will share buffers in a pool. Buffers in the pools are not permanently assigned to strings associated with specific files but can be utilized by different files in the pool. Each pool is divided into *subpools*. Each subpool contains a set of buffers of one particular size. CICS allows us to specify which LSR pool a file will use. Within each pool, data and index components are assigned to different subpools based on their CI sizes.

When a record is accessed using LSR, VSAM needs to look at each of the high levels of index, the sequence set and the data CI just as it did with NSR. With LSR, however, before assigning buffers or doing I/Os, VSAM will determine if any buffer in the proper subpool contains the CI to be read. If one does, data in the buffer will be used without rereading the CI from DASD. This feature is called *look-aside*.

With LSR it is not only possible to avoid reading high-level index records but in most cases it is probable that a high percentage of the sequence set CIs will also be located via the look-aside feature. With look-aside, it is not uncommon to satisfy 80 to 95 percent or more of all requests for index data using CIs already in the pool. The exact read/hit ratio will depend on a number of factors discussed below. (The term, *read/hit ratio,* will be used to denote the ratio of CIs located using look-aside to total read accesses requested.)

While it is likely that a fairly large percentage of index records will be in the pool when they are needed, a considerably smaller percentage of data CIs will be available in the pool. This is because there are usually many times more data CIs in a VSAM file than index CIs. Depending on data access patterns, LSR may provide various levels of I/O access savings for data CIs.

VSAM maintains the LSR pool using a *Least Recently Used* (LRU) algorithm. Whenever a record needs to be read from DASD, LSR will select the buffer that has been referenced least recently. This, in effect, tends to keep those buffers in memory that have been referenced most recently and that are most likely to be read again shortly. Those CIs that are referenced frequently (such as high-level index records) will tend to remain in the pool. Lesser-used data will normally need to be reaccessed from DASD each time it is needed. The amount of benefit LSR can provide depends on the size of the pool (the number of buffers) and specific activity against files sharing the various subpools in the pool.

A typical access to an active VSAM KSDS would proceed roughly as follows.

1. VSAM would attempt to read the highest level index record and then other higher level index records (if any). There is a high probability that all these CIs would be resident somewhere in the proper subpool and I/O operations usually will not be required. A read/hit ratio of 95 percent or better should be expected for high-level index data in a tuned LSR subpool.

2. VSAM will then attempt to read the sequence set CI. Depending on the size of the file, the number of buffers in the pool and the activity of other files sharing the pool, it is reasonable to expect that a fairly large percentage of sequence set CIs will be found in the subpool. This will be particularly true of files in which a high percentage of the activity falls in a relatively narrow key range.

3. Finally, the data CI will be accessed. On a relatively small file containing highly active reference data, 80 percent or more of the data CIs requested may be in the pool when they are needed. On larger files in which data tends to be reaccessed quickly (for example, a clerk working a number of transactions for the same city in a city file), it is also possible to have a good data read/hit ratio — perhaps as high as 70 to 80 percent. Larger files with truly random access patterns may have a much smaller data read/hit ratio because of the small probability that CIs will be in the pool when they are needed.

Of the three read operations necessary to access a VSAM KSDS file with one high-level index, LSR may often average less than one physical I/O per logical access. For small, highly active files, it is reasonable to expect a read/hit ratio of 90 percent or better overall. With similar files accessed via NSR, the high-level index could be retained in memory, but each access would average almost exactly two physical I/O operations (usually one for the sequence set CI and almost always one for the data CI).

While LSR could average as few as 0.3 physical I/Os per access, NSR could do no better than about 2.0. For larger files or files with more random access patterns, LSR would achieve a somewhat lower read/hit ratio and smaller percentage of I/O savings. And, for very large files with multiple high levels of index, LSR might achieve a combined read/hit ratio for index and data CIs of only 50 to 60 percent unless some tuning was done.

Limitations When Using LSR

Most specific limitations to the use of LSR have been eliminated by Release 1.7 of CICS and most VSAM files should now be eligible for inclusion in LSR. However, there are a few situations in which LSR will either cause problems or not perform as well as NSR.

Perhaps the greatest problem associated with placing a file in LSR is that certain programming practices can cause program lock-ups or data contention. When using NSR, a single task can access multiple records on a single file concurrently. The CICS command level interface will allow a task to both browse a file and perform updates or deletes on the same file without ending the browse operation. With NSR the VSAM update operation does not interfere with the browse.

With LSR, though, VSAM will first determine if the CI being accessed is already in the pool. If it is and is in use (that is, it is being held for an update or browse operation), VSAM will have the requesting task wait until the buffer becomes free. If that same task is holding the buffer, the task will become locked up. It will be waiting for the release of a resource that it owns itself and cannot free. The task will remain waiting until it is cancelled. This can lead to additional complications if other CICS tasks need any of the resources held by that task.

With LSR, it is important that applications refrain from accessing multiple records on the same VSAM file concur-

STROBE

MVS APPLICATION TUNING

Exploring the Boundaries of Space-Time

Basic questions about Space-Time remain unanswered today. Will the Universe expand forever? What, if anything, could retard its expansion?

What about your IBM MVS mainframe universe? Will your memory, disk space, hiperspace and CPU time requirements also expand forever? Is there anything you can do to retard this expansion?

Fortunately, you don't have to master Einstein's general theory of relativity to answer these questions. Programart's STROBE® Performance Measurement System tells you just where CPU time is spent within your mainframe address spaces. It enables you to quickly spot and correct inefficiencies that produce excessive resource demands.

With STROBE, you can:

- Tune your applications both during and after development
- Reduce online response times
- Shorten critical batch processing runs
- Postpone costly hardware upgrades

To learn how to keep critical applications within your space-time boundaries, call Programart today at (617) 661-3020.

PROGRAMART

1280 Massachusetts Ave. ■ Cambridge, MA 02138

STROBE will help you tune applications in CICS, DB2, IMS DB/DC, and other DB/DC environments, as well as batch processing programs.

STROBE is a registered trademark of Computer System Architects Incorporated. © Computer System Architects Incorporated 1988.

CIRCLE #33 on Reader Service Card ▲

rently. Even when records with different keys are accessed, if they both are located in the same CI, a lock-up will occur.

For these same reasons, it is important that CICS tasks retain control of LSR buffers no longer than necessary. Since no other task can access data in a CI until it is released, it is important to design applications with this in mind. It goes without saying that records read for update should be rewritten, deleted or released as quickly as possible. (This is also true for NSR processing.)

With LSR it is also important to prevent browse operations from holding a particular CI for prolonged periods of time. Some examples of situations to be avoided would include not ending a browse during segments of a conversational task or updating other files before ending a browse. In either case, records in the CI being browsed will be unavailable to other CICS tasks until the CI is freed.

Files that are browsed heavily may be better serviced by NSR than LSR. Browse operations under NSR may benefit from read-ahead processing. If extra data buffers are defined for a file using NSR (more than one data buffer per string plus the extra data buffer that VSAM reserves for its own use), the first task starting a browse operation will allocate all of the extra buffers for read-ahead processing. It will schedule multiple chained I/Os and attempt to fill the buffers with data before it is needed. Any other browse operations starting while these buffers are being used will use a single data buffer each. Thus, one string at a time can benefit from read-ahead processing with NSR. All other strings will use only one data buffer. With LSR, all browse requests use one buffer at a time and there is no read-ahead. But if the CI to be read happens to be in the pool already, it will not have to be reread.

Perhaps the most significant problem using LSR for files that are heavily browsed is that each CI read during a browse will always use the least recently referenced buffer in the pool. Browses spanning several data CIs will tend to fill the pool with those CIs being browsed since each CI will normally be read into a separate buffer. This can hurt the performance of other files sharing the pool by overlaying highly used data. For this reason, it is best to either not use LSR for files of this type or place them in a separate pool by themselves.

Control Area (CA) splits can cause performance problems in CICS, especially for files using LSR. When a CA is split, numerous I/O activities will occur, including reading and rewriting approximately half of the CIs from one CA to another. On a typical KSDS with a 4K data CI size residing on a 3380-type device, about 70 records will need to be read from the old CA and then copied to the new CA that will almost always be on another cylinder. With NSR, VSAM uses multiple buffers and attempts to read or write as much as a track at a time. This can significantly reduce the number of physical I/Os and the amount of DASD arm movement. With LSR, though, each record to be read and written requires a separate I/O operation. Since the two CAs are almost necessarily on separate cylinders, each I/O operation will involve DASD arm movement. Processing a CA split with LSR can take many times longer than with NSR.

NOW, CICS AND VSAM UNDER VM WITHOUT A GUEST OPERATING SYSTEM!

With VMCICS™ and VMVSAM™ from Unicorn Systems Company, you can develop and run your production CICS and VSAM applications directly under VM. And you can do so without compromising compatibility or performance. No resource hungry guest operating system is required. No hardware restrictions—from the IBM® 9370 to 43xx to 30xx and compatibles. No headaches.

VMCICS/Development System allows you to develop, test and debug CICS applications directly under VM with exceptional productivity. You can obtain true VSE or MVS CICS compatibility with the advantages of developing under VM/CMS. Your developed applications can be moved back to VSE or MVS for production or remain under VM using VMCICS/Execution System.

VMCICS/Execution System is a multi-user production CICS environment for VM which provides outstanding performance. CICS/VSAM applications easily port from VSE or MVS. And, under VMCICS/ES, they operate with improved stability—no more region crashes. In effect, users get their own "virtual region."

VMVSAM is a multi-user, shared file system for VM which provides full VSAM compatibility. Programs written in COBOL, Assembler or REXX can now share VMVSAM files. And VMVSAM supports concurrent sharing of files between batch and online programs operating under VMCICS/ES—with full data integrity.

Together, VMCICS and VMVSAM make it possible to move your CICS/VSAM applications to VM, the most popular and fastest growing IBM mainframe operating system. So isn't it time for you to say "no more guests"? Be *our* guest by calling toll-free today at **800-222-6974** (from California, **800-232-CICS**).

unicorn

Unicorn Systems Company
3807 Wilshire Blvd.
Los Angeles, CA 90010 (213) 380-6974

IBM Business Partner
Authorized Application Specialist

VMCICS and VMVSAM are trademarks of Unicorn Systems Company.

CIRCLE #44 on Reader Service Card ▲

During the time that VSAM is processing a CA split, the MVS TCB requesting the split is suspended. In other words, while a CA split is occurring, no other work gets done in that CICS region unless VSAM subtasking is being used. If VSAM subtasking is being used, the subtask would be suspended limiting access to all VSAM files in that region. While CA splits are always troublesome, they are much more so with LSR. As a general rule, files should be designed to eliminate CA splits if at all possible. If CA splits cannot be almost totally avoided, it might be best to exclude the file from LSR processing.

VSAM files defined with share-options of "4" should not be used in LSR. (In fact, they probably should not be used in CICS.) Share-options "4" is used in a crude sort of way to allow sharing of VSAM files. It requests that VSAM reaccess each level of index and data from DASD each time data is retrieved. When share-options "4" is specified, neither look-aside nor read-ahead processing will be done. There is no reason to have such files in LSR since CIs in buffers can neither be reused nor benefit from look-aside.

Tuning LSR Processing

Usually when LSR is first used, several VSAM files are placed in a pool and everything runs fine. In most cases, there are several improvements compared to NSR. For one thing, there is usually a considerable decrease in the amount of virtual storage needed for the pool (which is not that important any more in an MVS/XA environment since the pool can reside above the 16MB line), a slight reduction in working set size, a reduction in the number of physical I/Os and often a reduction in total CPU utilization (especially for DFP Releases 2.3 and later). Typically, the improvements of LSR over NSR have been significant enough that little concern is given to making it perform even better. There are a few techniques, though, that can improve LSR performance and/or help balance resource utilization.

The data and index CI sizes of VSAM files are quite important when using LSR. To begin with, data CI sizes should be different from index CI sizes for all files sharing a pool. Since LSR pools are subdivided into subpools based on buffer size, having different sizes for data and index CIs will force them into different subpools. This will prevent data CIs (that commonly have poorer read/hit ratios) from stealing buffers holding index CIs (that usually have good read/hit ratios). This need is particularly important when one or more files in the pool is more active than other files or when much browse activity is occurring.

Since there is almost always a better read/hit ratio for index CIs than for data CIs, when data CIs are allowed to steal index buffers additional physical I/O activity will occur. If at all possible, it is good to protect index CIs by isolating them from data CIs. It only takes one highly active data file stealing index buffers to lower the performance of the entire pool.

There is also value in matching data and index CI sizes with allowable LSR buffer sizes. The only buffer sizes supported by LSR are 512, 1,024, 2,048, 4,096 and multiples of 4,096 up to 32,768. CIs of other sizes will use the next larger

XA-RELO / XR-RELO

CICS Performance Optimizer

As you may have already discovered, converting to XA does not necessarily mean your CICS performance and storage problems are over. Now it's time to let the powerful features of XA-RELO provide the solutions. (XR-RELO provides similar solutions for MVS/SP).

- Enjoy faster response times
- Improve internal throughput
- Eliminate all CICS storage compressions
- Eliminate virtual storage constraints
- Eliminate Short-on-Storage conditions
- Increase the Dynamic Storage Area (DSA)
- Eliminate all program fetches from the CICS load library during execution
- Reduce system I/O and WAIT time
- Allow all programs and mapsets to reside in the XA address space without any recompiles or modifications, including macro level programs
- Easy to install ... less than 30 minutes without any system modifications or program changes

— Call now for a free trial —

(800) 542-7760 • FAX (205) 833-8746

Quantum International Corporation

"Superior Solutions"

CIRCLE #63 on Reader Service Card ▲

buffer size. While some other CI sizes may have other storage/access advantages on certain media types, they will waste buffer space in LSR.

Since any CI resident in memory will use an entire buffer, it might as well be filled with data to improve the chances that data will be in the pool when it is needed. Unless there are strong reasons to do otherwise, data and index CI sizes should be used that match LSR pool buffer sizes.

On 3380 geometry, 4,096 is generally considered an ideal data CI size for most *randomly* retrieved VSAM files. This size provides a good balance of service and space utilization on the device. Prior to DPF Release 2.2, when VSAM could use only physical block sizes of 512, 1,024, 2,048 and 4,096, it was considered good practice to choose data CI sizes when defining a VSAM file and allow VSAM to select an appropriate index CI size. At that time, VSAM was just about forced into assigning index CI sizes of 2,048 or 4,096 for most VSAM files defined on 3380-type devices. DFP Release 2.2 changed the structure of VSAM by allowing physical records to be the same size as CIs. This change allowed more flexibility in VSAM record geometry but significantly increased exposure to another problem.

When a data CI size of 4,096 is requested with DPF Release 2.2 and beyond and index CI size is not also specified when the file is defined, VSAM will be free to choose smaller index CI sizes. An index CI size of 1,536 is frequently chosen by VSAM. Using information about the file, the type of DASD upon which it was being created and the length of the key, VSAM will select the smallest appropriate index CI size based upon standard key compression algorithms. (VSAM compresses the keys stored in index records to conserve space.)

Unfortunately, when key compression does not occur according to standard patterns, the index CI size is sometimes not large enough to map all of the data CIs it controls. This leads to unusable space within the file and to premature CA splits. When VSAM was forced to use a 2,048 index CI size instead of 1,536, the additional space in the CI took care of most unusual key compression patterns.

A data CI size of 4K and index CI size of 2K are preferred for randomly accessed VSAM files on 3380-type devices, especially when used with LSR. This will provide good DASD service, make good use of space in the LSR pool, keep the index and data CIs in separate subpools and, in most cases, allow full utilization of space allocated to the file. In cases in which a 2K index CI is not sufficient either because of larger keys or poor key compression, you can either increase the data CI size to 6K or 8K or you can increase the index CI size to 4K. (If the index CI size cannot be 2K, it might as well be 4K since it will use a 4K buffer anyway.)

It might be wise to avoid allowing both the index and data CI sizes to be 4K since there would be no way to place them in separate subpools. Unless the device or its path is overly busy, it probably would be better to increase the data CI size to 6K or 8K and allow the index CI size to remain at 2K. A 6K data CI size would be better than an 8K CI size if DASD space is important (with 6K, you can get 42K of data per 3380 track versus 40K with an 8K CI size). An 8K data CI size would be used to avoid wasted space in the LSR pool, especially if the DASD device was not overly busy and/or was supported by DLSE.

Another way to improve LSR performance is to use multiple LSR pools. Beginning with CICS Release 1.7 on MVS/XA systems, CICS allows the creation of up to eight separate LSR pools. The placement of files into separate pools can improve the read/hit ratio for files in each of the pools.

One of the most important things to consider when planning LSR pools is data and index CI sizes. As just mentioned, it is important to ensure that no file in a pool has the same data CI size as any other file's index CI size. It is also worthwhile to separate files with good read/hit ratios from ones with poor read/hit ratios. While this will probably not help the files with poor hit ratios, it will prevent them from robbing buffers that are likely to be needed again quickly.

Highly active files may warrant special planning. Files with high levels of activity and good data read/hit ratios are probably good candidates for having their own LSR pool, especially if their performance is critical. With only eight LSR pools and perhaps hundreds of files, it is usually not practical to give files their own pools, regardless of how active they are. When we cannot give this type of file its own pool, it may be possible to place its data CIs in a separate subpool. This can be done by using an unusual data CI size (such as 12K) that is not being used by any other file in the pool. This might be especially beneficial for small files with high data read/hit ratios.

However, care should be taken when larger data CI sizes are used. They can have a negative impact on the I/O subsystem. If the use of a separate subpool eliminates most of the read I/Os associated with a file, the larger data CI size will probably have little impact unless there is also a considerable amount of write activity. Unless the volume upon which the file resides and the associated paths are all 30 percent or less busy (higher path activity can be allowed on DLSE architecture) *and* there is a significant improvement in the read/hit ratio, larger CI sizes should be avoided. If the separate subpool does not materially improve the read/hit ratio, it is probably better to continue to use a smaller data CI size.

Files with high levels of activity and poor data read/hit ratios should probably share a pool by themselves. Even if look-aside will do little for the data CIs, with an appropriately sized index subpool, a reasonably good read/hit ratio can still be achieved for index and sequence set CIs. For these files it is especially important that data and index CI sizes be different.

Files involved with extensive browse activity may occasionally be good LSR candidates, but they should usually be placed in a separate pool or subpool. If the length of a typical browse request involves only one or two CIs, VSAM processing may not be materially different than for randomly accessed files (from an LSR perspective). The advantage of leaving such files under NSR is that one application (and only one!) at a time could benefit from VSAM's read-ahead feature.

The advantage of using LSR is that some CIs might already be in buffers. Depending upon the particular read/hit ratio, there might be significant real I/O savings, especially for files experiencing shorter browse requests.

Prior to DFP Release 2.3 (and on all non-XA releases), LSR may have used more CPU resource than NSR. At that time, LSR look-aside routines would scan *every buffer* in a subpool before ever assigning a buffer for an I/O operation. If the correct CI was in the subpool, look-aside would scan an average of half of the buffers in the subpool before finding the correct one. If the CI was not in the pool, the entire pool would be scanned. Of course, if the CI was in the pool, the entire path length for scheduling and servicing an I/O would be eliminated. Larger pools required more CPU (one reason why

smaller pools used to be recommended). Higher read/hit ratios saved more of the CPU associated with scheduling I/Os. Depending on the size of the LSR pool and the read/hit ratio realized, LSR could have used more or less CPU than comparable NSR file processing.

DFP 2.3 introduced a new LSR buffer management technique called *VSAM hash buffer management*. This changed the way look-aside accessed buffers in a subpool. With this technique, VSAM look-aside routines use a hashing algorithm to calculate the address of buffers in the pool instead of searching through the pool sequentially. The amount of processing required to locate a record is now substantially less regardless of the size of the pool.

With DFP 2.3, it is no longer necessary to create separate LSR pools to reduce CPU utilization. In fact, the opposite is true — the trend is now to make LSR pools as large as possible (within limits of storage) to improve the read/hit ratio for the pool. Larger pools can improve the read/hit ratio and reduce the number of physical I/Os requested for files in the pool. It can also reduce the CPU overhead needed to schedule and service the associated I/O events. Systems not troubled with paging problems or real storage constraints can now reduce CPU usage by selectively increasing the sizes of various LSR subpools to improve read/hit ratios.

Determining the optimal size of each LSR subpool requires consideration of a number of factors. The primary reason for increasing the size of an LSR subpool is to improve its read/hit ratio. The relationship between the number of buffers in a subpool and the resultant read/hit ratio is somewhat "exponential" in nature. That means that the read/hit ratio will improve rapidly at first but improve less rapidly as more buffers are added.

The higher the read/hit ratio for a particular subpool, the more buffers that must be added to improve the ratio. In other words, it will usually take more buffers to increase the read/hit ratio for a subpool from 90 to 95 percent than it did to improve that same subpool from 70 to 75 percent.

Data access patterns are the major factor influencing read/hit ratios. Files for which data is reaccessed quickly will show greater benefits from increases in pool sizes than files with more random access patterns. The number of buffers necessary to achieve a given read/hit ratio is related to the likelihood that records will be accessed multiple times in a relatively short period of time. In some situations subpools with 20 to 30 buffers will achieve excellent read/hit ratios (90 to 95 percent or more) while subpools of 200 to 300 buffers may achieve less than 60 to 70 percent. The differences are entirely related to the access patterns of files sharing the subpool.

Of course, it is assumed that tuning for CI size and pool selection has been done before adjustments are made to pool sizes. It may be hard to compensate for poor file design or CI size selection simply by providing additional buffers.

It should be reasonable to expect read/hit ratios of 90 percent or more for subpools used exclusively for indexes. Increasing the size of these subpools until a 90 to 95 percent read/hit ratio is obtained is usually the easiest way to improve overall performance.

Since DFP 2.3, when LSR's look-aside processing was improved, LSR has become one of the best methods of improving performance and extending the life of CPU-constrained processors. On processors with expanded storage, if read/hit ratios can be materially improved by creating large subpools, total CPU utilization will usually be lower, even if additional paging results. As long as the page faults generated by an increased working set size can be satisfied from expanded storage, the CPU savings associated with improved read/hit ratios should be greater than the CPU usage associated with satisfying page faults. While larger LSR pools will increase working set size and paging rate, the net result can still be a reduction of total CPU usage along with performance improvements resulting from replacing real I/Os with satisfied look-aside requests.

When we discuss expanding the life of a CPU-constrained processor, we normally talk about two primary aspects: the reduction of total CPU utilization and the ability to achieve acceptable service levels at higher levels of CPU utilization. By exploiting LSR and expanded storage, we can do both. Much larger buffer pools will probably become common as we attempt to make better use of processor resources.

Measuring LSR Effectiveness

Standard CICS statistics produced at shutdown give a picture of how the pool performed overall. The statistic, "Num-

XA-VSAM / XR-VSAM

Improve VSAM performance as much as 90%

- **Cuts hours off daily production time**
- **Significantly reduces I/O and WAIT time**
- **Analyzes and dynamically tunes the performance characteristics of all VSAM datasets**
- **Automatically provides optimum VSAM buffer management for maximum efficiency**
- **Eliminates dedicated resources now used to tune VSAM datasets**
- **Requires no program or system modifications**
- **Easy to install . . . less than 30 minutes**

— Call now for a free trial —

(800) 542-7760 • FAX (205) 833-8746

Quantum International Corporation

"Superior Solutions"

CIRCLE #15 on Reader Service Card ▲

ber of successful look-asides,'' is the number of requests satisfied directly from the subpool. The statistic, ''Number of buffer reads,'' indicates the total number of records actually retrieved from the I/O subsystem. The read/hit ratio can be determined by dividing the total number of requests satisfied from the pool by the total of that number and the total number of read I/O requests. This statistic gives a picture of how well the pool performed for the entire time CICS was processing. If more precise information is needed for a specific period of time, these same statistics can be produced in separate interval reports.

Information on read/hit ratios for specific files is not available directly from CICS but is available using various CICS performance monitors. One vendor's monitor provides some satistics that are particularly useful. It indicates the total logical I/O services requested for each file (that is, read, write, add, delete and browse requests) and the number of EXCPs actually scheduled. For LSR files, these numbers are a good indication of the read/hit ratios for a file unless there is a lot of browse activity against the file. Some monitors analyze LSR pool performance without identifying specific files.

Service times may also be shown either by file or by subpool. While low service times usually reflect good read/hit ratios, they can be influenced by a number of factors such as the amount of browse activity, the type of DASD and whether the DASD is being cached. Service times are clearly the ''bottom line.'' Most I/O tuning is done to improve the average delay in providing data to applications.

Other Considerations

When VSAM Subtasking (VSP) is being used, CICS sets up a separate MVS task to service VSAM I/O requests. This allows most of the processing associated with VSAM I/O requests to be processed under a separate TCB. In multi-processor complexes (dyadic, quadratic and so on), VSP is used primarily to relieve CPU-constraint in overly busy CICS regions. When VSP is used, all VSAM I/O requests will be serviced by the VSAM subtask removing some processing burden from the main CICS TCB. Since VSP adds a considerable amount of overhead, it is not normally used unless a CICS region has a CPU demand of *80 percent* or *more*. (CPU demand is defined as the sum of the time a CICS region is using the CPU plus all other time it is attempting to use the CPU but cannot, such as when it is waiting for page faults or CA splits or for the availability of a processor being used by other higher priority MVS tasks.) Of course, VSAM subtasking is not recommended for use on a uniprocessor.

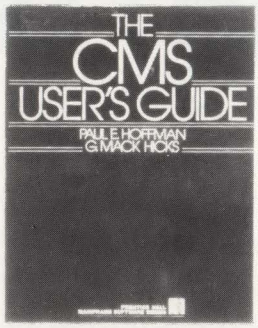
The use of VSP in a heavily LSR-oriented environment may be somewhat counterproductive. When VSAM files have high read/hit ratios, read accesses will be serviced quickly without scheduling much I/O and associated processor utilization will be minimal. Without VSP, service can be almost immediate. With VSP, requests must be shipped to the subtask for processing. This can add unnecessary delays and overhead in VSAM processing.

When LSR is used extensively, the need for cache controllers can be reduced considerably. Generally speaking, files with reasonable LSR read/hit ratios become poor cache candidates. The only I/O requests fowarded to the I/O subsystem are the ones that failed look-aside. Since LSR provides a type of pre-cache, the read/hit ratio within the cache controller will often not be high enough to justify the use of cache.

Conclusions

In most CICS environments, LSR should be the vehicle of choice for accessing VSAM KSDS files. While there are a few restrictions on its use, it will normally provide the best service and make the best use of available resources. With the enhancements to the look-aside feature provided in DFP Release 2.3, LSR has become a powerful tool that may be used to meet both performance and capacity objectives.

A Major New Addition to the Prentice Hall Mainframe Software Series...



THE CMS USER'S GUIDE
Paul Hoffman & G. Mack Hicks

Useful to both computer novices and more advanced users, this concise guide to the hugely popular Conversational Monitor System (CMS) outlines both the commands and the concepts behind them. Describes Release 5 of CMS and some common variations, plus an overview of previous releases. Covers both standard and nonstandard configurations, compares CMS to other operating systems, and details the most widely used commands, use of disks and files, and IBM applications.

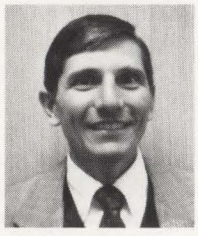
1989, 176 pp., paper 013-187725-9, $24.95

Available at better bookstores or direct from Prentice Hall at 201-767-5937. Attention corporate customers: for orders in excess of 20 copies to be billed to a corporate address, call 201-592-2498.

Prices subject to change without notice.

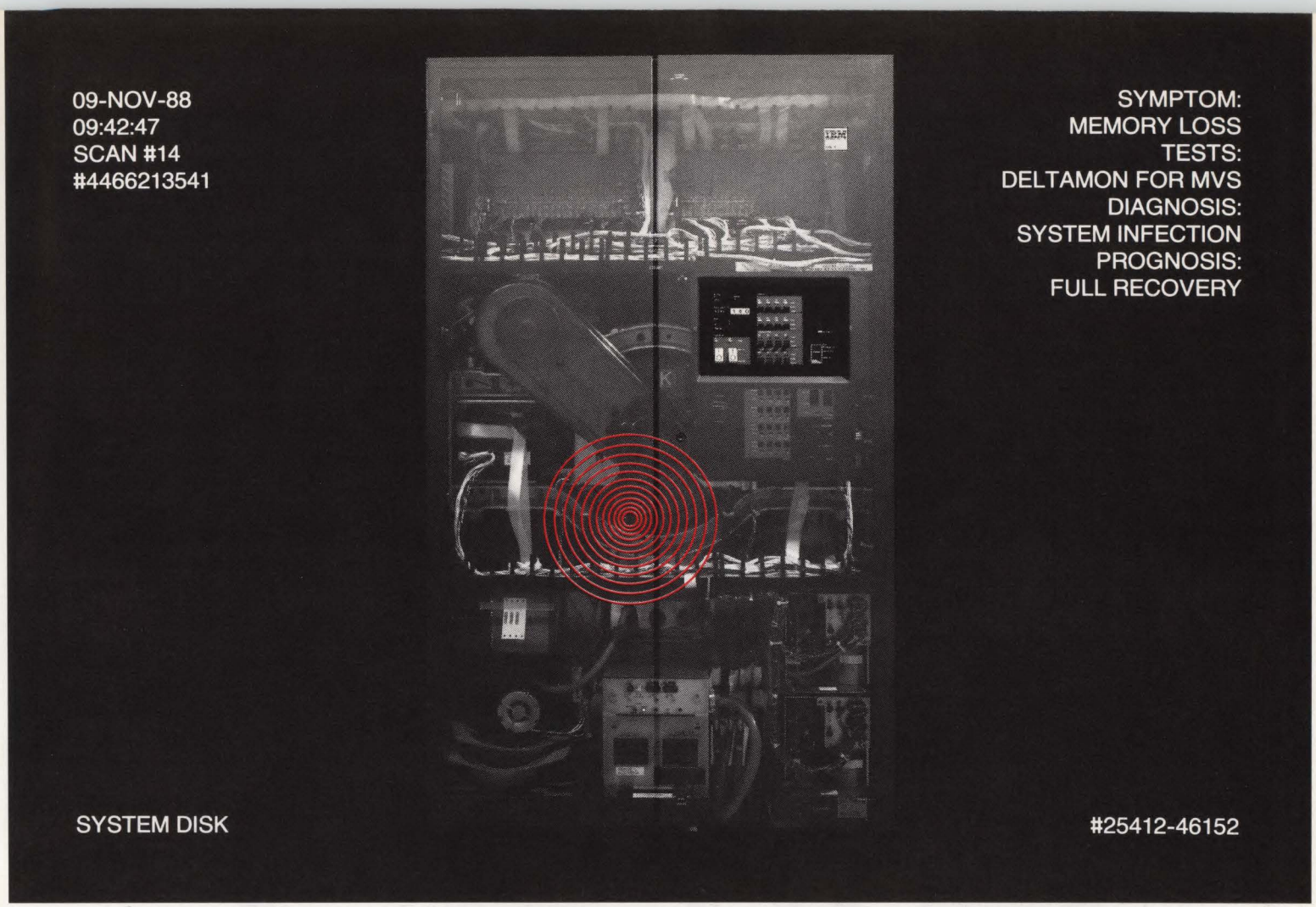
PRENTICE HALL
Simon & Schuster
Higher Education Group
Englewood Cliffs, NJ 07632

ABOUT THE AUTHOR



Ted C. Keller is the manager of a group responsible for CICS systems support, performance management and capacity planning at Yellow Freight System, Inc., 10990 Roe Ave., Overland Park, KA 66211, (913) 345-3274. He has worked in various data processing jobs for more than 21 years.

CIRCLE #106 on Reader Service Card ▲



Can you find it before it infects your whole MVS system?

Something's wrong. Suddenly. The files you were working on earlier are totally gone. Vanished. And your whole user network can't get access to the online applications. Something's different in your system. Something's changed.

But how do you find it? And how much time do you have before the MVS system's disabled? Is it something in your hardware, software, data sets? Is it viral? Just where do you start to look?

Look at DELTAMON® for MVS, change detection software from Candle. DELTAMON can tell you exactly what has changed in your system...and who changed it, whether it's a data center staffer or a user on a PC tied into the mainframe. It can pinpoint precisely any system alterations from hardware configuration to changes in applications, the security system, SMP, and data sets, including authorized libraries.

Then, once DELTAMON collects that information, it provides you with detailed reports of every single one of those changes, so you get a thorough audit trail that tracks all activity. It can just as easily verify scheduled changes — the changes you expected to be made — so you know they were made when they should have been.

You have a choice. You can take a chance and hope you'll be able to find changes before they damage your system. Or you can install DELTAMON and fix problems before they spread.

To learn how DELTAMON for MVS can protect the health of your MVS system, call Terry Forbes or your Candle Account Representative today, toll free, at **1 (800) 541-8515**. And make a change for the better.

!Candle®

CIRCLE #120 on Reader Service Card ▲

Why COBOL Efficiency Still Makes Sense

By Harvey Bookman

These frequently used adages are becoming more and more uncritically accepted by data processing management. "Computers are now so fast that it does not matter how inefficient your program is" . . . "Concentrate on the application, not the program" . . . "Spend your time on a structured program, not a technically efficient program." The saying used to be, "It is important to write efficient code." Why has this idea fallen out of favor in recent years?

It is indisputable that efficient code is at least sometimes necessary. As companies grow, so does the amount of data they process. When the rate of increase of data processed exceeds the rate of increase in the speed of computers, additional processing time becomes critical. If a production run cannot finish between the time a company ends daily processing in the evening and begins the next day's activities, a company cannot function. Furthermore, even if a job can process in this time frame, any programming problems become major ones. The schedule may be too tight to allow for a rerun.

Efficiency can actually be the deciding factor between life and death. I have a friend that has written programs designed to automatically perform defensive maneuvers of the F16 fighter jet. Imagine a missile coming at your jet and your sudden realization that your program is running slowly due to inefficient code. You had better hope that you can get your parachute and bail out more efficiently than your defensive program was processing.

Efficient programs certainly save a company money. At the end of 1984, I spent more than a month updating COBOL programs that ran on an IBM 3081. By using techniques detailed in this article, I was able to save between 15 and 95 percent of the CPU time for each program. This saved a major insurance company more than $30,000 each time the programs were executed which was a few times per year. One program that originally ran in more than three hours CPU time was changed to run in less than one hour while another was changed from running more than 15 minutes to less than one minute.

Efficiency is unfortunately considered something that when added to a program makes the program more convoluted and

harder to maintain. While this is the case in some highly technical pieces of code that strain the limits of COBOL, this code is rarely added for efficiency alone. It is a fallacy that efficient code must be difficult code. One often overlooked valuable result of efficient code is that the programming logic is often easier to follow. Consider the example that locates a month name based upon the numeric value of the month (see Figure 1). Which of the code segments do you find easier to read and understand?

It is quite apparent that efficiency is not always of primary importance. If a programmer writes code that will run only once, a small program that runs quickly or a large program that processes a small amount of data, extra time spent in making the program run faster is not cost-effective. Yet, with a proper understanding of the concepts of COBOL efficiency, the basics of which can be learned in about two days, you can learn to write efficient code as quickly as you can write inefficient code. Although it may be hard to break bad habits, it is much easier than you think to code efficiently.

Figure 2 illustrates two different pieces of code that serve the same function. Although the code segments look quite similar and would take approximately the same amount of time to write, one of the segments runs about three times as fast as the other. Reading this article should make it quite apparent as to which is the faster running code.

It often pays to scrutinize existing code and improve its speed. A COBOL program of 1,500 lines that has already been written can be reviewed and updated to run faster in less than a day. I have rarely, if ever, seen a program that cannot be made to run at least 15 percent faster. One quick approach is to install a program monitor and determine the exact points in your system where excess CPU time is being utilized. These are the points at which you can concentrate your efficiency efforts.

Perhaps you are convinced that the efficiency of a COBOL program is important but it can be accomplished by one of the fine COBOL optimizer programs on the market. An optimizer can only make the best out of the code it is given. It cannot rewrite the logic. While it may increase the speed of a poorly written program, this is no substitute for well-written code that is optimized.

Following is a description of a number of techniques to enable an IBM mainframe COBOL program to run faster. While the list does not specify all methods, it serves as a starting point from which to proceed. The coding suggestions serve to motivate you into examining new techniques of your own. By using the COBOL Procedure Map (PMAP) that contains the Assembler language listing of a program, you can extend your ability through the study of the Assembler code produced for different methods of COBOL coding. You can also experiment by running two different programs that produce the same results and by comparing the speed of each.

FIGURE 1

```
DATA DIVISION.
WORKING-STORAGE SECTION.
77  MONTH-NUMBER                    PIC 99.
77  MONTH-NAME                      PIC X(9).
01  MONTH-TABLE.
    05  FILLER                      PIC X(11)  VALUE  '01JANUARY   '.
    05  FILLER                      PIC X(11)  VALUE  '02FEBRUARY  '.
    05  FILLER                      PIC X(11)  VALUE  '03MARCH     '.
    05  FILLER                      PIC X(11)  VALUE  '04APRIL     '.
    05  FILLER                      PIC X(11)  VALUE  '05MAY       '.
    05  FILLER                      PIC X(11)  VALUE  '06JUNE      '.
    05  FILLER                      PIC X(11)  VALUE  '07JULY      '.
    05  FILLER                      PIC X(11)  VALUE  '08AUGUST    '.
    05  FILLER                      PIC X(11)  VALUE  '09SEPTEMBER'.
    05  FILLER                      PIC X(11)  VALUE  '10OCTOBER   '.
    05  FILLER                      PIC X(11)  VALUE  '11NOVEMBER  '.
    05  FILLER                      PIC X(11)  VALUE  '12DECEMBER  '.
01  FILLER REDEFINES MONTH-TABLE.
    05  MONTH-ENTRY OCCURS 12 TIMES
            ASCENDING KEY IS MONTH-TAB-NUMBER
            INDEXED BY MONTH-INDEX.
        10  MONTH-TAB-NUMBER        PIC 99.
        10  MONTH-TAB-NAME          PIC X(9).
PROCEDURE DIVISION.
    . . .
    SEARCH ALL MONTH-ENTRY
        WHEN MONTH-NUMBER — MONTH-TAB-NUMBER (MONTH-INDEX)
            MOVE MONTH-TAB-NAME (MONTH-INDEX) TO MONTH-NAME.
```

```
DATA DIVISION.
WORKING-STORAGE SECTION.
77  MONTH-NUMBER                    PIC 99.
77  MONTH-NAME                      PIC X(9).
01  MONTH-TABLE.
    05  FILLER                      PIX X(9)  VALUE  'JANUARY   '.
    05  FILLER                      PIX X(9)  VALUE  'FEBRUARY  '.
    05  FILLER                      PIC X(9)  VALUE  'MARCH     '.
    05  FILLER                      PIC X(9)  VALUE  'APRIL     '.
    05  FILLER                      PIC X(9)  VALUE  'MAY       '.
    05  FILLER                      PIX X(9)  VALUE  'JUNE      '.
    05  FILLER                      PIC X(9)  VALUE  'JULY      '.
    05  FILLER                      PIC X(9)  VALUE  'AUGUST    '.
    05  FILLER                      PIC X(9)  VALUE  'SEPTEMBER'.
    05  FILLER                      PIC X(9)  VALUE  'OCTOBER   '.
    05  FILLER                      PIC X(9)  VALUE  'NOVEMBER  '.
    05  FILLER                      PIC X(9)  VALUE  'DECEMBER  '.
01  FILLER REDEFINES MONTH-TABLE.
    05  MONTH-ENTRY OCCURS 12 TIMES PIC X(9).
PROCEDURE DIVISION.
    . . .
    MOVE MONTH-ENTRY (MONTH-NUMBER) TO MONTH-NAME.
```

Arithmetic Efficiencies

The IBM 370 family of mainframe computers performs fixed point arithmetic using one of two formats: packed decimal or binary. Packed decimal fields are represented as COMP-3 and binary fields as

COMP in COBOL programs. While packed decimal arithmetic may be done in memory, binary arithmetic always requires at least one register. This means that it will often take more instructions to do computations on a binary field than on a packed decimal field. However binary instructions are faster and usually more than make up for the additional instructions.

Zoned decimal fields are defined in COBOL as display numeric: a numeric field with neither COMP-3 nor COMP associated with it. If arithmetic calculations must be performed on the field, the hardware requires that the field be converted to either packed decimal or binary, thus increasing the amount of time necessary for the code to execute.

Arithmetic calculations should be made between fields of a similar type whenever possible, packed decimal to packed decimal or binary to binary. Whenever a calculation is made between a packed decimal and a binary field, the compiler will be forced to convert one of the fields (that can be to either packed decimal or binary depending upon the exact circumstances), thus using extra computer time.

An often overlooked efficiency is that calculations should be performed upon fields that have the same number of decimal places. It is not important whether one field is longer than the other, only that the number of places on the right side of the decimal point be the same. When this is not the case, the compiler has to create code to move one of the fields into a work area where it will be given the same amount of decimal places as the other field. This may also cause an additional move if the work area field must then be moved into the resulting field.

It should be understood when calculations are to be performed upon fields of different descriptions or differing number of decimal places, one of the fields can be moved to a WORKING-STORAGE field that matches the description of the other field. This technique will improve the speed of the program if the field is used multiple times. However it is not useful when a field will be used only once since the COBOL compiler will make the conversion just as efficiently as an added MOVE statement.

An understanding of packed decimal and binary arithmetic on the IBM 370 leads one to analyze the lengths that are best for each type of data as well as whether signs should be used on fields. Packed decimal fields allow two digits for each byte except the last that contains one digit and a sign. Therefore, a packed decimal field must always contain an odd number of digits. If a packed field is defined to the COBOL compiler as having an even number of digits, the compiler must generate extra code to strip out one of the digits.

Binary fields are handled as either halfwords (that can hold any four-digit number) or fullwords (that can hold any nine-digit number). Any larger field must be manipulated by the compiler to produce proper results, even though the hardware does not specifically support an instruction for this function. Binary fields with more than nine digits should be used with caution. Although some large binary numbers can be handled effectively by the compiler, many times the fields are internally converted to packed decimal. While COBOL allows only 18 digits for a numeric field, the hardware allows 31 for a packed decimal field.

Beware of using large numeric fields regardless of their internal representation. For instance, if a division is done using two eighteen-digit fields with differing numbers of decimal places, a COBOL internal subroutine will be called to do a division similar to the long-hand way it is done on paper. The division may take as long as the rest of the program.

Arithmetic should also be done between fields that are defined with a sign: an S in the data definition. COBOL does all arithmetic as signed whether or not it is requested as such. When a field is not defined as signed, the compiler must insert an extra instruction to remove the sign.

IF Statements

Many programmers fail to code the IF statement efficiently. It requires a certain amount of understanding and finesse to code eloquently. Take for example the code in Figure 3. The second segment of code is an attempt to be faster, yet accomplish the same function as the first segment. If CHECK-FIELD is always numeric, the second version will run much faster and is fine. However if CHECK-FIELD is not always numeric, while the second segment looks equivalent to the first, it may produce different results! A value of 2X will *not* make the first IF statement true but will make the second true.

What is most important to you ?

- ❑ Programmer Productivity
- ❑ Effective Use of Machine Resources
- ❑ Responsiveness to End Users

Call **800-336-2432** for information about:

MAGEC®

"A Better Way To Develop Better COBOL Applications"

MAGEC Software™ • P. O. Box 260319 • Plano, TX 75026
800-DD-MAGEC • 214-248-0823

VSE • MVS • VM/CMS • PC

CIRCLE #99 on Reader Service Card ▲

Alphanumeric rather than numeric compares should be used whenever possible, since they are faster. They also will not cause abends when invalid data appears in a field. Note that signed display numeric fields are the least efficient fields to compare. The compiler must first convert the fields to packed decimal so that a numeric compare can be done.

One valuable piece of knowledge of 370 hardware is the fact that *immediate instructions* exist where the operand field of a single character is placed into the instruction itself rather than into data storage. This enables an IF or MOVE statement operating on a single constant character to run about three times as fast when the field is coded as a constant in the statement rather than have the field defined as a constant in WORKING-STORAGE. Defining frequently checked indicators as one-byte fields with values of Y and N instead of YES and NO should not detract from the clearness of a program while increasing its speed.

It should be mentioned that using an 88 level field in an IF statement is neither more nor less efficient than comparing a field to a constant. In either IF statement in Figure 3, code of equivalent speed would have been generated had the program used the 88 level rather than the constant values. In the first case, six comparisons must be done while in the second, only two.

Complex IF statements that contain a number of AND and OR conditions can be speeded up simply by rearranging the order of the condition checks. The rule for efficient comparisons is as follows: in statements with many OR conditions, put the conditions with the greatest chance of truth first; in statements with many AND conditions, put the conditions with the greatest chance of truth last. The rule can be applied to nested IF statements as if they were multiple AND conditions. If you give the rule a bit of thought, it will become clear why it is true.

FIGURE 2

```
DATA DIVISION
WORKING-STORAGE SECTION.
01   MONTH-NUMBER                PIC 99.
01   FINISHED-INDICATOR          PIC XXX.
PROCEDURE DIVISION.
     ...
     IF MONTH-NUMBER IS EQUAL TO 12
       MOVE 'YES' TO FINISHED-INDICATOR.

-----------------------------------------------------------------

DATA DIVISION.
WORKING-STORAGE SECTION.
77   MONTH-NUMBER                PIC XX.
77   FINISHED-INDICATOR          PIC X.
PROCEDURE DIVISION.
     ...
     IF MONTH-NUMBER IS EQUAL TO '12'
       MOVE 'Y' TO FINISHED-INDICATOR.
```

FIGURE 3

```
DATA DIVISION.
WORKING-STORAGE SECTION.
01   CHECK-FIELD                 PIC XX.
     88  CHECK-VALUES            VALUE '01' '02' '03' '04' '05' '06'.
PROCEDURE DIVISION.
     ...
     IF CHECK-FIELD = '01' OR '02' OR '03' OR
                      '04' OR '05' OR '06'
       DISPLAY '***IS EQUAL***'.

-----------------------------------------------------------------

DATA DIVISION.
WORKING-STORAGE SECTION.
01   CHECK-FIELD                 PIC XX.
     88  CHECK-VALUES            VALUE '01' THRU '06'.
PROCEDURE DIVISION
     ...
     IF CHECK-FIELD > '00' AND < '07'
       DISPLAY '***IS EQUAL***'.
```

Table Handling

Optimizer programs can do a great deal to improve the performance of table handling in COBOL programs. Yet there are a number of technical points to consider to increase program speed when processing tables whether or not your program will be optimized. Most COBOL programmers can recite that it is better to use an index than a subscript mainly because it is a common interview question. What most programmers do not realize is that not only are absolute subscripts (subscripts defined as a constant in the statement using it) the most efficient form of subscripting; but also, they are more efficient than indexes and no less efficient than using a field that is not in a table! This is because when an absolute subscript is used, the compiler makes the exact determination of where the field is. At run time, the object code does not have to perform calculations to locate the field in the table.

While indexes are more efficient than subscripts, many, if not most programmers, do not use indexes often. Perhaps it is because they do not have the ambition to learn how to use them properly and how to find them in a dump.

If a subscript is used, it is always internally used as a binary integer. This is

the reason that subscripts should be defined as COMP whenever possible.

PERFORMs and GO TOs

When I teach that PERFORMs are inefficient and should be avoided where program speed is a major concern, it never fails to amaze me when at least one person jumps up and complains that I am attempting to destroy structured programming concepts. One would think that I was a KGB agent who had infiltrated the United States to destroy programming through my teachings.

Take a look at Figure 4. It is not hard to tell which segment of code is easier to read as well as many times faster. Unless a PERFORM is in-line, as can be done in COBOL II, you have to find the performed paragraph making it harder to follow the code. Since PERFORM statements may be harder to follow than in-line code, why not avoid them when possible? This becomes of added value when the code is performed millions of times and the added overhead of a PERFORM is eliminated.

One rarely used but extremely efficient statement in COBOL is GO TO DEPENDING ON. Regardless of how many possible paragraphs are the object of the GO TO clause, the code never takes more than 10 or 11 instructions to branch to the correct paragraph. This is because COBOL does not generate a list of IF statements but utilizes a branch table that uses the DEPENDING ON value as its subscript. When analyzed there are often cases that can use the GO TO DEPENDING ON statement, such as a program that peforms 50 different routines depending upon a numeric state code.

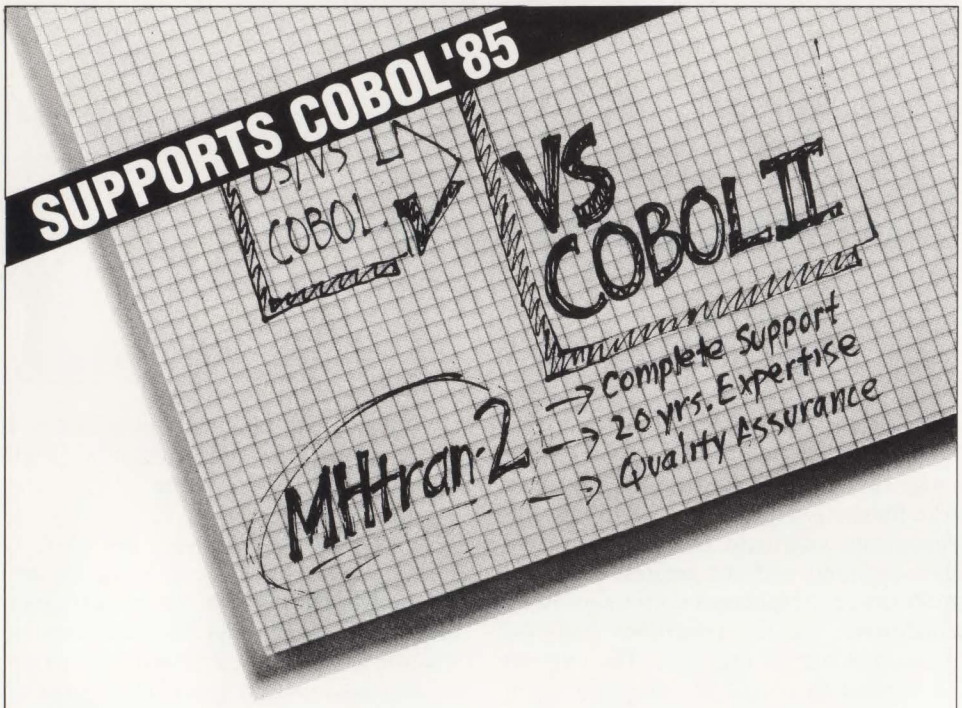
File Processing

Fixed-length records should be used rather than variable-length records whenever possible. Avoid using the OCCURS DEPENDING ON clause to define variable-length COBOL data items whether they are record descriptions or WORKING-STORAGE fields. Every time the object field of the OCCURS DEPENDING ON changes, COBOL takes time to recompute the length of the data item.

When variable-length records with a wide range of lengths are used, one extremely efficient technique is to use the APPLY WRITE-ONLY clause in the I-O-CONTROL paragraph. While it may come as a surprise to many, COBOL often does not fill up a variable-length block before writing it. As an example, take a program that uses a file containing two record sizes: 100 bytes and 10,000 bytes. Assume that the block size of the file is 10,008. While it may seem that many 100-byte records will be placed into a block before it is written, this is normally not the case.

After COBOL puts a variable-length record into a block, it checks whether the largest record that may be written by the program can fit into the current partially-filled block. If it cannot, the current block is written out. In our example, if a 100-byte record is written into an empty block by a program, COBOL checks whether a 10,000-byte record will fit into the partially filled block. It cannot, so the block is written. If the program continually writes 100-byte records, each will be placed into its own block!

I/O takes a considerable amount of computer time and when records are not blocked, the processing time can increase significantly. To prevent this occurrence, the APPLY WRITE-ONLY clause in-



If you're planning the pivotal move to VS COBOL II...

Don't migrate alone.

MHtran-2 translates both batch and CICS programs from OS/VS COBOL to VS COBOL II. Completely. Accurately. Automatically.

And MHtran-2 guarantees you what no other translator can — the MHtran-2 migration experts who support your VS COBOL II migration from beginning to end. Every step of the way.

For information on a free 30-day trial, call 201-934-0022

MHtran-2

YOUR COBOL II TRANSLATION SOLUTION

PRINCE SOFTWARE PRODUCTS P.O. Box 804 Mahwah, New Jersey 07430 201-934-0022

CIRCLE #101 on Reader Service Card ▲

The greatest career advancement you could make this year isn't on the job.

It's in an Amdahl systems education course.

Let's face it, as a systems programmer your greatest asset to your employer is your knowledge. When you want to move up in the company, it's not who you know, it's what you know that counts.

Yet, in the ever-changing world of systems programming, keeping your knowledge current and up-to-date is no easy task. You can wade through dull textbooks, try to follow complicated interactive video courses or attend classes taught by instructors more interested in selling products than teaching knowledge.

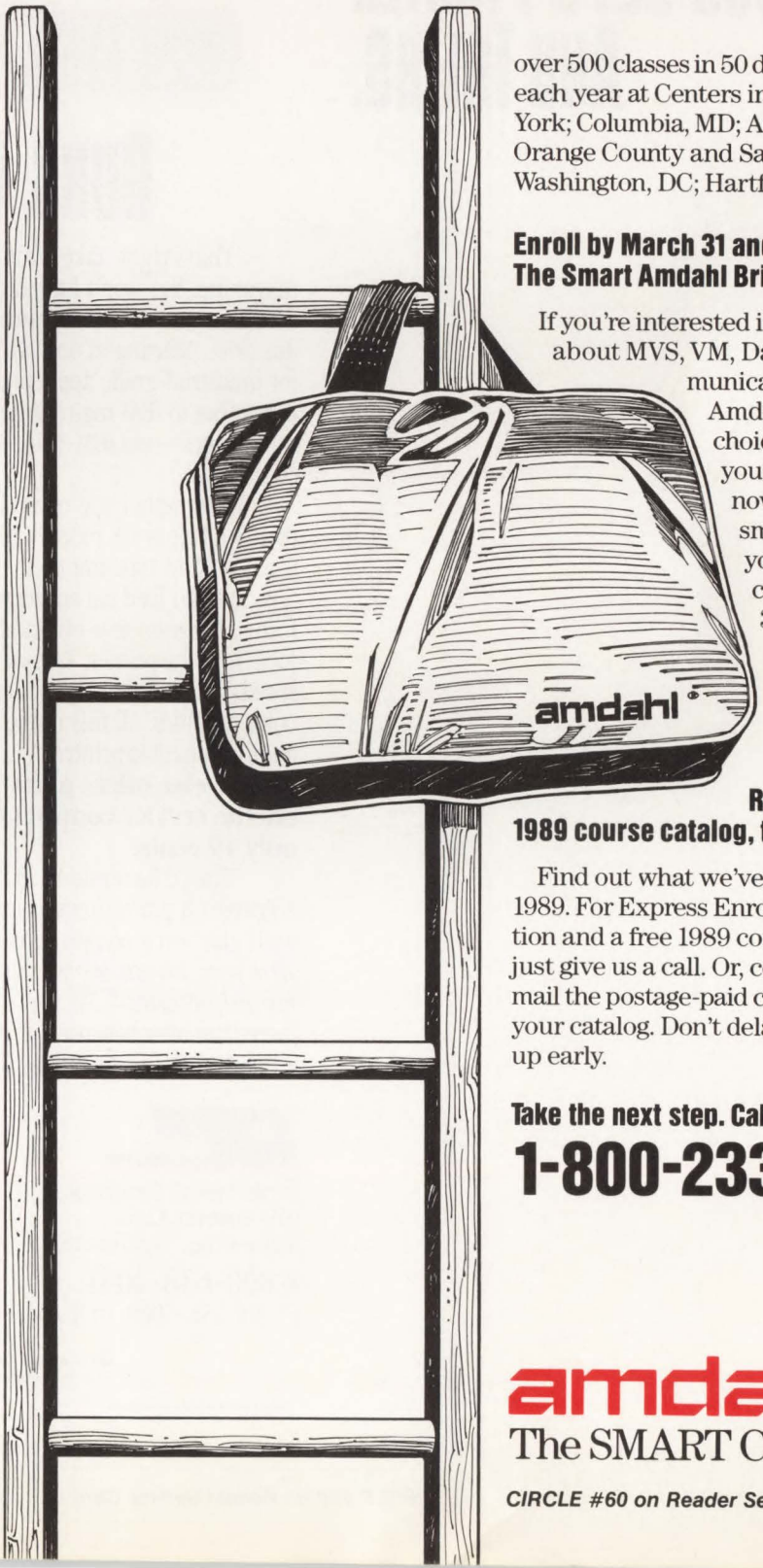
Or you can take an Amdahl systems education class. Our programs are designed to bring you up to speed in an intensive, hands-on environment.

Professional, individual instruction in all 370 system architectures.

Amdahl is the only education service that caters to all 370 system architectures, *regardless* of vendor hardware. And we're the only center staffed by veterans in the field who bring a technical superiority and unparalleled expertise to every course. Plus, we keep our classes small to encourage an open forum where information is readily exchanged.

Fully-equipped Education Centers nationwide.

All Amdahl courses are taught in full-service Education Centers located all across the country with on-site labs for hands-on application. We schedule over 500 classes in 50 different subjects each year at Centers in Chicago; New York; Columbia, MD; Atlanta; Houston; Orange County and Santa Clara, CA; Washington, DC; Hartford, CT.



Enroll by March 31 and get The Smart Amdahl Briefbag FREE.

If you're interested in learning more about MVS, VM, Data or Communications Systems, Amdahl is the smart choice for you and your company. And now, it's even smarter—when you attend a course by March 31, 1989 we'll send you The Smart Amdahl Briefbag FREE.

Request a FREE 1989 course catalog, today.

Find out what we've got going in 1989. For Express Enrollment information and a free 1989 course catalog, just give us a call. Or, complete and mail the postage-paid card attached for your catalog. Don't delay, classes fill up early.

Take the next step. Call today:

1-800-233-9521
ext. 132

amdahl®
The SMART Choice

CIRCLE #60 on Reader Service Card ▲

At $1,595 The Major Difference Between Our 3270 Printer And Theirs.

Now 50% Faster!
And with optional color graphics.

Ours Exists.

That's right. Take a minute and flip through this magazine. You won't find another mainframe printer that gives you as many features for anywhere close to this price. Starting at only $1,595, the Prima-CX is designed for industrial-grade applications, not PC-type printing. It attaches to IBM mainframes as well as 4300 and 9370 mid-ranges — via IBM 3174, 3274, and 3276 cluster controllers.

So whether you're producing memos or listings, there's a Prima-CX model (80- or 136-column) that's for you. The IBM-type front panel puts control at your fingertips. You can feed cut sheets without removing continuous forms. Program tear-off and RPQ options. Cancel print jobs from the printer. Take advantage of the 324 cps draft speed. You'll also enjoy the new optional GDDM and PSS color graphics. All this in the smallest footprint of any coax-compatible printer.

No one else offers a first-year, on-site service contract for only 49 cents.

They'd be afraid to. But, buy a Prima-CX professional printer and we'll give you a comprehensive, *first-year, on-site* service contract for *only* 49 cents!* We can make this incredible offer because our design ensures maximum reliability. Contact us today for complete details.

PSC

Printer Systems Corporation
9055 Comprint Court
Gaithersburg, Maryland 20877

1-800-638-4041
(1-301-258-5060, in Maryland)

IBM is a registered trademark of International Business Machines Corporation. Referenced model numbers are the products of International Business Machines Corporation.

*Subject to PSC terms and conditions.

CIRCLE #50 on Reader Service Card ▲

structs the compiler to fill a variable-length block to its maximum capacity waiting until the record length of a WRITE statement is known rather than predicting whether the next record might not fit into the current block. While some coding restrictions must be followed when the clause is coded, it is well worth using.

Compiler Options

What if someone told you that they might be able to make your COBOL program run faster without changing a line of code? Although it sounds like a trick, it is fact that programs will run faster or slower depending upon the compiler options chosen. This has nothing to do with a compile taking four or five times as long because one uses options like PMAP and SXREF on the first compile rather than SYNTAX.

The NOTRUNC option is rarely understood which is not surprising because of the way it is described in the IBM manual. It allows a program with COMP fields to run faster. Furthermore, depending upon how the program was coded, it may actually be a necessary option for the program to run properly. As mentioned earlier in this article, binary fields require a halfword for COBOL fields defined up to four digits long and a fullword for fields up to nine digits long. A halfword really holds up to 32,767 while a fullword up to 2,147,483,648. While a halfword or fullword COMP field in COBOL is defined as having a limit of 9,999 or 999,999,999 respectively, the actual numbers that may be put into the fields are larger.

The TRUNC option assures that when arithmetic is performed on a COMP field, the result does not exceed the maximum value that the field is defined as, adding extra machine code to do so. It is rare that a program wishes to ensure that a field exceeding its defined number of digits be truncated to the defined size. A more common practice is for a COBOL program to require that a field not be truncated because it is holding an address, as is the case in CICS programs. I see no reason that every company not make the NOTRUNC option the default for all compiles.

I would also like to mention a new option that has been added to the COBOL II compiler and will become important as companies convert. The SSRANGE option ensures that subscripts used in a program do not exceed the maximum entry number of a table. While this option is long-awaited and will prevent storage overlays as well as protection exceptions when programs are tested, the machine code added by the compiler will make the program run considerably slower. While useful in development of programs, the option should be avoided in the production environment.

While it is in vogue to accept as normal an environment of programmers who do not understand the implications of their own code relying upon the wide array of debugging and optimization products, the programmer who lacks the detailed knowledge of COBOL will also be lacking in productivity. It should be realized that anyone who suggests that programmers attempt to use a programming language without the understanding of those important intricacies of the language are asking the programmers to stumble around in the dark. The time spent by a programmer to understand a language, rather than blindly code it, will end up as cost efficient since coding errors will be resolved considerably faster. Rather than companies running to upgrade to faster machines to run their programs, let them run to educate their programmers — their most valuable resource. This rearrangement of priorities will translate into dollars saved and more knowledgeable programmers ready to meet the new technical challenges ahead.

FIGURE 4

```
DATA DIVISION.
WORKING-STORAGE SECTION.
01  TABLE-SUB                          PIC S9(4) COMP.
01  TABLE-OF-VALUES.
    05  TABLE-ENTRY OCCURS 3 TIMES PIC S9(3) COMP-3.
PROCEDURE DIVISION.
    . . .
CLEAR-TABLE.
    PERFORM CLEAR-ENTRIES THRU CLEAR-ENTRIES-END
        VARYING TABLE-SUB FROM +1 BY +1
        UNTIL TABLE-SUB IS GREATER THAN +3.
    . . .
CLEAR-ENTRIES.
    MOVE ZERO TO TABLE-ENTRY (TABLE-SUB).
CLEAR-ENTRIES-END. EXIT.

-----------------------------------------------------------------

DATA DIVISION
WORKING-STORAGE SECTION.
01  TABLE-OF-VALUES.
    05  TABLE-ENTRY OCCURS 3 TIMES PIC S9(3) COMP-3.
PROCEDURE DIVISION.
    . . .
CLEAR-TABLE.
    MOVE ZERO TO TABLE-ENTRY (1).
    MOVE ZERO TO TABLE-ENTRY (2).
    MOVE ZERO TO TABLE-ENTRY (3).
```

ABOUT THE AUTHOR

Harvey Bookman is president of Bookman Consulting, Inc., a software development company specializing in programmer proficiency testing. He is responsible for the development of the expert system testing service, TECKCHEK™. Bookman Consulting, Inc., 67 Wall Street, Suite 2411, New York, NY 10005, (212) 819-1955.

DB2's Query Management Facility

One User's Experience

By Mary Jo Cater

Query Management Facility (QMF) is IBM's offering as an end-user report writing tool for DB2 data. It is marketed as a user-friendly product and, indeed, was proven to be so at the Electrochemical, Detergent and Specialty Products Group (ED&S) of Occidental Chemical Corporation.

As a member of the newly created four-person information center within the information services department when I joined the company in May, 1985, my mission was to provide micro and mainframe user tools and support to compliment the existing production systems.

One of the major problems at that time was a backlog of requests for ad hoc reports; the average turn-around time was slightly more than three weeks. It was obvious that the tools that were available to the user community (SAS, Inquire IV and AnswerDB) did not provide the ease of use that was required.

Most of our potential users were from the sales, marketing and customer service areas, people with minimal technical skills. This fact, combined with our limited training and support staff, dictated that we find a truly user-friendly product.

Most of the data required by users already resided on the mainframe and micro computers did not exist on a widespread basis; so our choices were limited to the mainframe environment. Most of the data was either in IMS databases or in flat files that further restricted possible product selections. This was a constraint unique to our staff size and the timeframe within which we needed to provide some relief. Writing our own bridges from IMS to a new product was not a viable alternative.

We reviewed most of the 4GL report-writing and database products on a preliminary level. An extensive comparison was made between IDMS/R and Cullinet's report writing products with IBM's DB2, DXT and QMF. For our final test, we asked both vendors to load our sample data on their machines. We then requested that they generate 12 specific reports while we watched.

The results of these comparisons provided a clear decision for us: IBM's DB2, DXT and QMF. The direction of IBM toward support of DB2 as a production database heavily influenced our decision as did the availability of DXT for simple IMS data extraction. The key reason for our selection was the immediate benefits that QMF could provide.

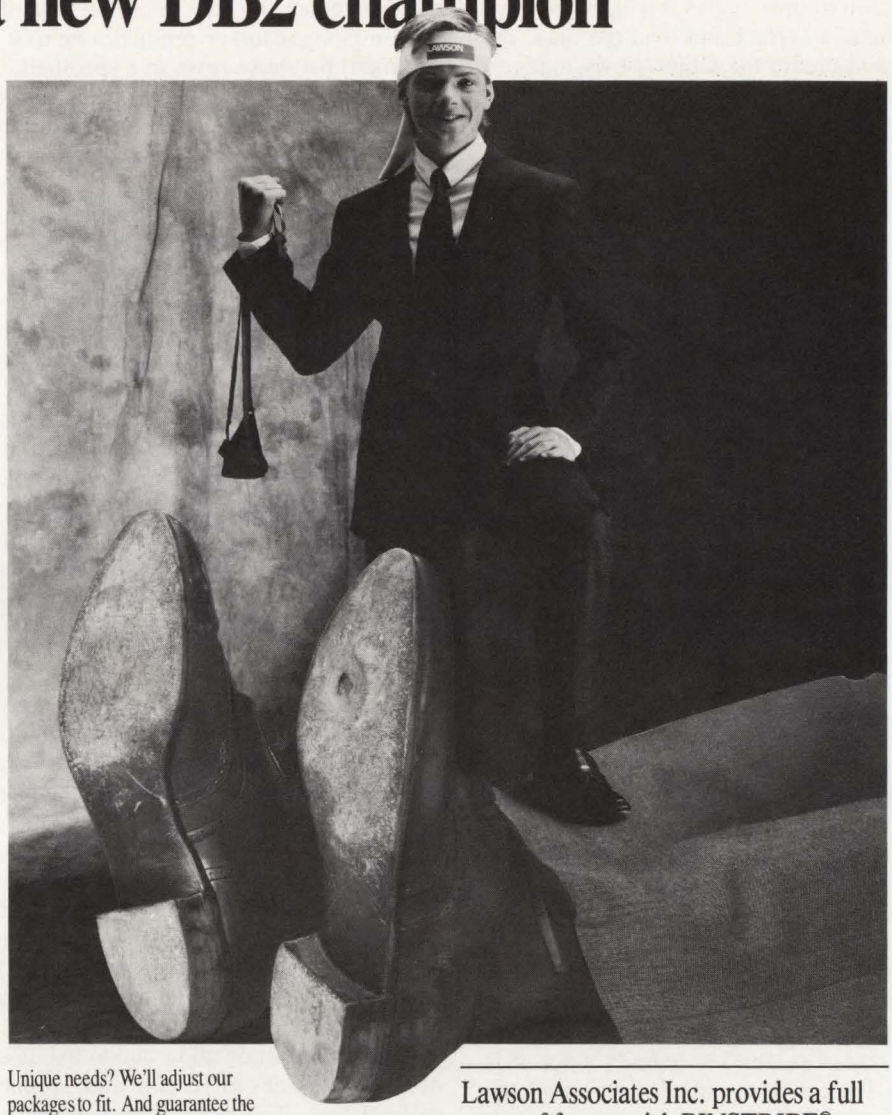
We selected several projects and users to be involved in a three-month pilot of the IBM products. We installed DB2, DXT and QMF under TSO in early January, 1986. During the first months of the pilot, the scope was expanded and the time extended to six months. At the conclusion of the pilot, a formal recommendation was made to purchase DB2, QMF and DXT. Due to enthusiastic user response, this recommendation was approved.

Over the past several years, the amount of data that has been made available to QMF users has increased dramatically. We currently have more than seventy tables/views and well over one hundred active users. New tables and users are constantly being added as quickly as we can accommodate them.

This activity is impressive only when you consider our staff size and the fact that there is no longer any backlog of requests for ad hoc reports. The Information Center (that has expanded to seven members) has

IBM mainframe application software has a new DB2 champion

We're Lawson, the IBM mainframe business application software company with the confidence to challenge the industry's Goliaths. And hit them with new ideas. Like low cost, low maintenance—yet full-featured—business application software.



Lawson: Leader in software technology.
Our new technology is rapidly showing the "big guys" a few things about PINSTRIPE® low cost, low maintenance software. Forget "other" IBM mainframe software. With its high prices, old code, time-consuming installation and expensive upkeep.

New technology made our lower costs possible. It also made us first with application software for the DB2 relational DBMS. Now you get a choice: DB2 or VSAM. With newly developed code that's easy to install. Simple and inexpensive to maintain.

Solid application technology.
Comprehensive, well-thought-out application features reflect the experience of our package designers. Seasoned accountants, CPAs and human resource professionals who know what features and functions your users need.

And features they request. In the last four years, 84 percent of user-group suggested enhancements have been added to Lawson PINSTRIPE® packages.

Service and support — cutting the "big guys" off at the knees.
We're big enough to get the job done—but small enough to be sensitive to our clients' needs. With the best service and support available on the IBM mainframe market today. Including your personal software consultant, an expert who knows your needs *and* your application.

A full range of optional Support-Plus services is available, too. Like implementation consulting, data conversion, technical and application consulting. And a post-sale audit to make sure you're utilizing every feature we've built into PINSTRIPE® packages.

Unique needs? We'll adjust our packages to fit. And guarantee the work. If you'd rather do the job yourself, you'll find Lawson software easy to modify. And we'll still support the rest of the package. Other application software companies won't.

Lawson, IBM mainframe software's new champion.
We've put it all together. New technology with fresh code, including both VSAM and DB2 expertise. Full-featured application technology. Backed up by support that sets a new industry standard.

All in low cost, low maintenance application software from Lawson. No wonder we can take on the "big guys." And win.

Lawson Associates Inc. provides a full range of feature-rich PINSTRIPE® business application software packages for the IBM mainframe hardware environment. Available in the U.S., Canada and overseas.

Accounting
- General Ledger • Accounts Payable
- Accounts Receivable • Fixed Assets
- Project Accounting

Human Resources
- Payroll • Personnel • Benefits

Distribution/Procurement Management
- Order Entry • Inventory Control
- Purchase Order

LAWSON

...WHATEVER IT TAKES!

For more information, call Sue Weinacht
(612) 379-0258
LAWSON ASSOCIATES INC.
1300 Godward Street, Minneapolis, MN 55413

CIRCLE #32 on Reader Service Card ▲

only two people with QMF/DB2 responsibilities. I extract and load the data, design and create the tables/views/indexes, monitor performance and train users. The other person spends almost 100 percent of his time writing complex ad hoc reports. The fact that we have been able to expand our user base without increasing support staff speaks highly for QMF.

Occidental has recently begun application development in DB2 on a corporate-wide basis and we expect to see continued growth in that direction. A key factor in DB2's selection for many of these initial projects has been the report writing capabilities of QMF.

QMF offers two languages (QBE and SQL), an extensive array of functionality in formatting reports and procs (procedures) that can run and print formatted reports in one step.

Query by Example (QBE) is similar to an older IBM product with the same name that ran under VM. Users familiar with the old QBE product naturally like the QBE language. During our pilot, some of our users were attracted to the simplicity of QBE but quickly (within days) became convinced that it is too cumbersome to use. QMF actually takes the QBE query and converts it to SQL prior to execution, so there is some minor additional overhead involved.

Structured Query Language (SQL) is actually a subset of the SQL language that is quickly becoming an industry standard. The syntax is quickly learned since it is fairly logical. (My advice is to read the query aloud; if it makes sense, it will probably run.)

Excellent Help Function

Part of the ease of use can be attributed to the excellent help function and error messages that are part of the QMF package. When the error message reads, "SQL error at or before line three position 24," the user has a pretty good idea why the query did not run. The help function will further explain why the error message occurred and offer some suggestions as to how to resolve the problem.

A typical query in QMF might read:

```
SELECT REGION,CUSTOMER, CUST_NAME,
  PRODUCT,SUM(NET_SALES)
FROM TSALES
WHERE DIVISION = '42' AND DISTRIBUTOR = 'N'
GROUP BY REGION,CUSTOMER,CUST_
  NAME,PRODUCT
ORDER BY 1,5 DESC
```

This query selects the named columns, adds together the net sales (so that for each region-customer-product, one row is returned) for those rows in a specified division and a specified type of distributor. The results are ordered first by region (and then within region) and are ranked by sales.

One key user-friendly feature is that spaces have no meaning. A query may be strung out on one line or spaced anywhere on the query screen. I suggest using a new line for each keyword as shown in the example. This makes the query easier for other people to read.

I have also created "common libraries" of queries and forms that are available to all users. This provides samples for users to modify and cuts down on learning time. Some of the queries in this library have been built with variables that prompt the user for specific restrictions at run time.

Form Function

Once a query has been run, the form function can be used to format the results. A default form is created by QMF that spaces the columns nicely across the page and uses the column names for headings. The user can change any of these defaults including omitting any column from the report, changing the number of decimal places reported, wrapping a column of data (like customer name and address), changing the column headings and changing the field lengths.

The form function allows up to five lines of title, five lines of footing and 14 lines of final text. The report can have up to six levels of breaks (a break separates the data for readability and subtotals) and provides the capability to add break text at the heading and footing. The form allows the user to obtain averages, percentages, totals, minimum and maximum values.

Both the query and form can be saved for later use. Simple commands allow the user to display a list of saved items, display one of those items, erase old items and so on.

Often the user is just looking for some information and may not even print the results of the query. If a printed report is desired, a PF key does the job. For two copies, you depress the function key twice. The print is directed to the printer defined in the user's QMF profile or the default printer. We have most of our remote sites defined to GDDM (that QMF uses for printing), so that we can print any report in our key locations.

Procedures

To further simplify the query/form process, QMF provides for the creation of procs. A proc is merely a list of commands that can be executed in one step, either on-line or in batch. A very basic proc might look like this:

```
RUN QREGION1 (FORM = FREGION1
PRINT REPORT
PRINT REPORT
RUN QREGION2 (FORM = FREGION2
PRINT REPORT
```

This proc runs the query QREGION1 with the form FREGION1 and prints two copies of the report. Then it runs the query QREGION2 with the form FREGION2 and prints one copy of that report.

Procs are especially useful in the batch environment. Batch can be run through a QMF panel, from TSO or scheduled as part of normal production. One nice feature of using a QMF proc for batch reporting is that the queries and forms can be changed to reflect user needs often without rewriting the procedure or changing the JCL.

During our pilot, users were exposed to very simple table joins (two tables with a customer number key, one with sales data and the other with customer name and address). The concept was poorly received, resulting in confusion and incorrectly joined tables. To avoid this problem, we make extensive use of views. Our users are not even authorized to see many of our base tables; they use views that join several tables and often restrict what data the user will see. If someone has a need to join other tables, I build them a view on an ad hoc basis.

QMF also provides the syntax to UNION tables with similar fields. We have two tables that are good candidates for union: Open Orders and Billed Orders. Like joins, users have difficulty with this concept and the creation of a view has solved this problem.

Security

Security is handled for the most part by DB2. Access into QMF can be either open or closed. I suggest using closed to maintain control over those who would like to "play around" with the product. Once a user is authorized into QMF, the DB2 authorization for each table or view controls access to specific data. DB2 provides for virtually any level of security that is re-

See DB2 *page 77*

EMC Introduces Main Memory for 4381 Models 21-24

MORE MBs FOR YOUR $$$

EMC's 4381 Memory Upgrades: Thousands of Users Have Saved Thousands of $$

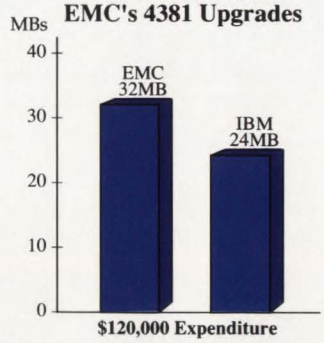
EMC's 4381 upgrades are priced 25% below those of IBM. Save $20,000 with each 16 megabyte addition to main storage and, as thousands of EMC customers have found, you can meet capacity requirements quickly and easily while improving reliability.

With EMC lower cost means higher reliability. By designing, manufacturing, testing and supporting our 4381 memory we are able to enforce strict quality control standards during all phases of the product development process. Each EMC 4381 memory board uses pre-tested logic components. In addition, each board undergoes 100 hours of stringent test and burn-in procedures and is qualified in one of EMC's in-house 4381 CPUs before certification for shipment.

Increased reliability and lower costs are only two of the benefits of EMC's 4381 upgrades. What's more, each EMC upgrade is backed by EMC's Critical Site Service Plan. This plan includes an On-Site Spares Kit and a worldwide network of Service and Support Offices to guarantee prompt assistance from our Customer Service Engineers.

Price advantages combined with proven reliability and support make main memory upgrades from EMC the right choice for maximizing the performance of your 4381 computer. Find out why EMC is the leading independent supplier of 4381 main memory upgrades by calling today or by writing: EMC Corporation, Hopkinton, MA 01748-9130.

Save 25% with EMC's 4381 Upgrades



For more information call today: **1-800-222-EMC2** (In MA, 508-435-1000).

EMC²

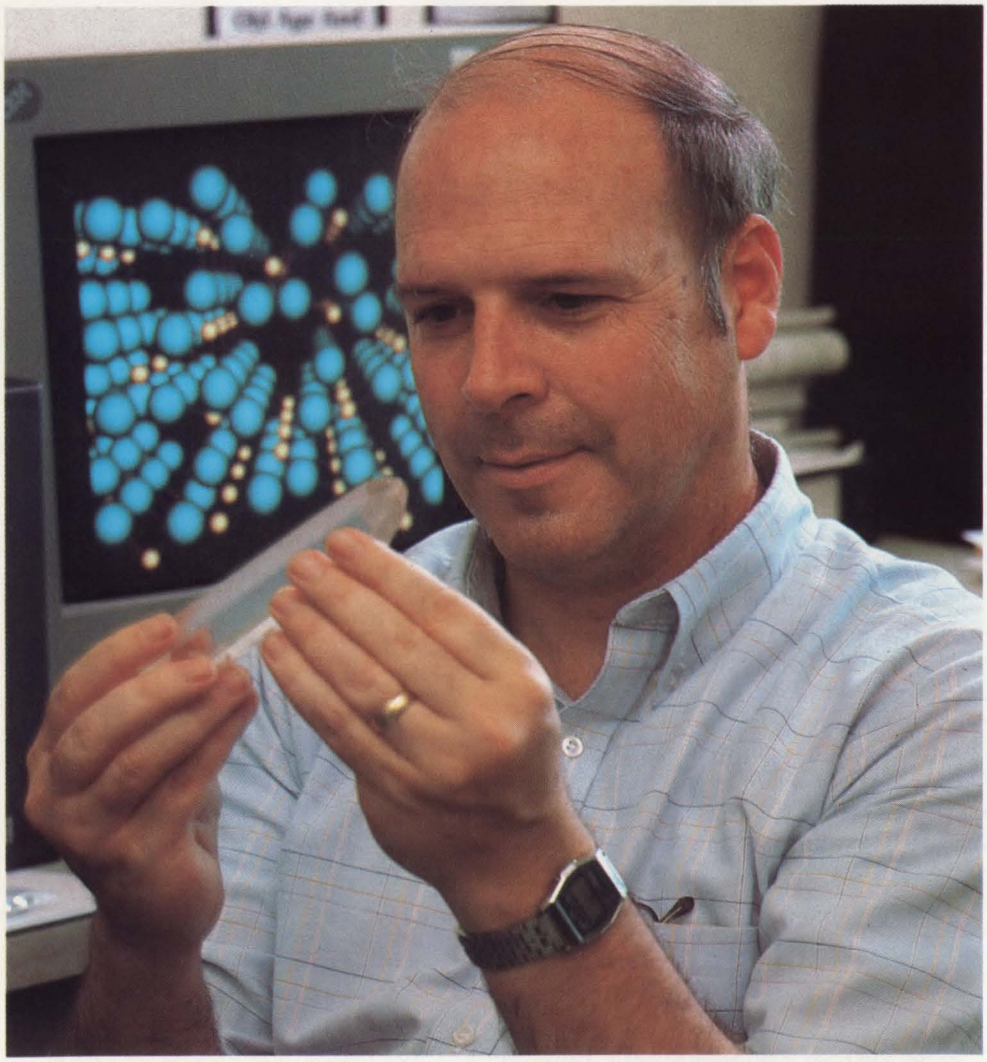
The System Enhancement Company.

IBM is a registered trademark of International Business Machines, Inc.

CIRCLE #95 on Reader Service Card ▲

ES/9370 And VM Generate Computer Simulations

By Stephen H. Quanrud



Dr. Teter of Corning Glass Works with a glass crystal. The screen in the background is a computer model of 288 atoms of silica, or glass.

Tapping at a keyboard, the researcher inputs commands telling the computer to translate huge numbers of instructions into graphic forms. Moments later, images of blue and yellow appear on the researcher's computer terminal.

As the screen fills, a pattern emerges: marble-sized spheres connected by rods, complete with shading. It looks vaguely reminiscent of high school chemistry . . . perhaps the pattern represents a molecule? Actually, it is less than that. It is one *slice* of one molecule.

This is a demonstration at the research and development complex at Corning Glass Works where a scientist is simulating a unique view of chemicals at work. Though it represents a tiny piece of matter, the model on the computer terminal carries big potential for the glassmaking firm. Generated by an IBM Enterprise System/9370, the graphics represent a precise view of how 288 atoms of molten glass might appear when cooled.

"When atoms are hot and moving around, they're not that important to us," says Michael Teter, Ph.D., Corning's engineering fellow. "We need to know what happens when they cool and settle down. Computer simulation gives us a still photograph of that phenomena very quickly without having to go through weeks or months of work."

To most engineers, computer simulations are becoming commonplace. Supercomputers routinely simulate phenomena ranging from the chemical reaction among atoms to the interaction of cold and warm fronts in weather. However, what is unique with the Corning computer model is the hardware used: this one was generated not by a supercomputer, but by one of the smallest of IBM mainframes, the ES/9370, in partnership with the IBM 3090 supercomputer at Cornell University's theory center.

"Using an elaborate, exact supercomputer program, we figure out what the interactions are between the atoms. Then we make up forces that approximate that for a simplified model and that's the one we do on the 9370," explains Dr. Teter, who personally wrote the modeling software used for Corning's ES/9370.

According to Dr. Teter, perhaps the biggest benefit of the 9370/3090 combination is at the heart of simulation itself: quick visualization of physical problems under investigation.

"You could look at 100 pages of numbers," he points out, referring to the com-

puter instructions required to generate the graphics on the screen. "But drawings like this help us visualize what the numbers mean by letting the computer figure out exactly what a simulated effect looks like."

For Corning Glass Works, the potential for improved product quality or even new products offered through computer modeling is tremendous. With 60,000 products, 47,000 corporate clients, millions of customers and $2 billion in annual sales, Corning (like other companies including IBM) is continually looking for ways to quickly and effectively improve its products and the processes that make them.

Corning gains cutting edge in glass modeling.

Computer modeling, Dr. Teter reasons, is perhaps the best tool because it allows researchers to simulate effects on the substances that comprise their products. "With models, we can do our chemistry by throwing atoms in a box," he explains, referring to computers as the box and numerical instructions as the atoms. "So we throw the atoms in this box, grind them down mathematically and let them cool. When you're done, you have a crystal. Doing this by computer cuts the time involved in the experimental process by at least one order of magnitude."

In addition to Corning Glass, IBM also stands to benefit from Dr. Teter's computer modeling. In what is termed a "cooperative testing environment," the ES/9370 Model 90 and associated VM software is on loan to Corning's research and development complex. As part of a joint study venture, IBM has agreed to let Corning operate the system for 18 months. Corning, in turn, will recommend possible improvements relating to hardware, software, service and marketing.

"As a company that's highly dependent on its research and development effort, Corning Glass is ideal for testing the 9370 in an engineering/scientific environment," points out Dick Sileo, an IBM advisory systems engineer for the Corning account. "Their researchers gain from the fact that with IBM's systems, they can obtain quick visualizations of their scientific solutions. IBM gains because Corning is on the cutting edge of this technology and they can provide valuable technical direction for our hardware and software."

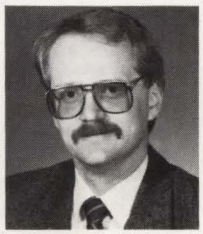
Although it has been an IBM customer for years, Corning, like many major users these days, is not "all IBM." Systems from IBM and other competitors are installed at many of the company's 45 locations worldwide. At Corning headquarters, for example, IBM equipment controls most commercial functions; DEC equipment controls most scientific and manufacturing processes.

According to Dr. Teter, the value of the IBM/Corning joint study relates to what can be learned about *productivity.* As the author of Corning's "Finite Element Fluid Flow" modeling software, Dr. Teter is considered a top theorist in computer modeling and glass chemistry. Computer modeling, he explains, holds great promise as a productivity tool for virtually all scientific disciplines.

"My vision for the future of supercomputing is that it will become the most cost-effective way to design and test new products, processes and materials. Eventually hardware and computing cycles will become effectively free . . . we will be limited by the number of truly capable people that we have. Consequently, my recommendation is to proceed aggressively with those factors which will increase the productivity of the people involved," Dr. Teter says.

The goal? Find radical new materials with which to "build" new products. "By letting the computer do the repetitive work that the theorists already know how to do, we can spend time on solutions. That's when the breakthroughs occur. That's what this is all about," he concludes.

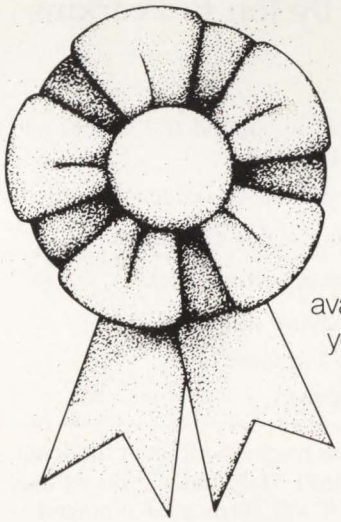
ABOUT THE AUTHOR



Stephen H. Quanrud is a communications specialist for IBM in Endicott, NY where he works as a writer and editor. After earning a Bachelor of Arts degree in Journalism from the University of Wisconsin, he worked for several newspapers before joining IBM in 1982.

ESP

EXECUTION SCHEDULING PROCESSOR™



The new standard in scheduling

ESP is the most versatile MVS scheduler available. The advanced design of ESP lets you manage your workload, whether you need a comprehensive system or want to phase in specific functions. Don't wait until problems get out of hand – ask us to demonstrate the flexibility of ESP today.

Cybermation Inc.

CYBERMATION INC.
16 ESNA PARK DRIVE
MARKHAM, ONTARIO
L3R 5X1
(416) 479-4611

Systems Excellence

CIRCLE #170 on Reader Service Card ▲

I·S·P·F Techniques

By Jon E. Pearkins

Interactive System Productivity Facility/Program Development Facility (ISPF) is an editor and on-line DASD utility as was explained in my previous article, "ISPF Spells Productivity," in the November/December, 1988 issue of MAINFRAME JOURNAL.

According to IBM, ISPF can be used to quickly and economically develop and maintain applications — batch and interactive. Programmers have convenient access to a set of tools used in all facets of on-line program development and testing. In short, ISPF offers an excellent full-screen editor and productive set of interactive panels for executing utility functions: delete, define, copy, rename and list information about datasets.

This article examines techniques that aid in making the use of ISPF more productive. These techniques include:

- Moving around fast in files and in ISPF itself
- The value of sequence numbers
- Unprintable characters
- Never getting stuck
- Tabbing simplified
- Performance.

LOCATE

If you happen to look at the letter "L" in the ISPF Help Index, you will see that the LOCATE Primary Command is one that can be used in various places in ISPF.

If you fill in an ISPF Library's name in the Browse or Edit Entry Panel but do not provide the name of the member you are interested in, you will see a Member Selection List listing the member names in alphabetical order along with statistics for each. For a large ISPF Library, scrolling forward (PF8) to the area of interest can be time-consuming. The LOCATE Command can be used to move directly to the area of interest. For example, if you know that the member you want begins with ICE, you can type LOCATE ICE to get to a list of all the members that start with ICE.

Once you are actually browsing or editing a file, the LOCATE Primary Command again eliminates a lot of scrolling. If you are working on several parts of a file, you can label the first line of each part and use LOCATE to get you directly to the appropriate part at any time. If you called a part WS, you would type .WS on the Command Line (= = = >) in Browse to mark the first line displayed on the screen as WS. LOCATE .WS would bring you back to this part of the file at

any time during the current Browse.

In Edit, you would use the Dot(.) Line Command to set up a label. Then, .WS would be typed over the sequence numbers of the first line of the part of the file you want to call WS. As with Browse, LOCATE .WS would later bring you back to the WS part of the file.

LOCATE can also be used with a number. LOCATE 400 would take you to the 400th line of the file in Browse. However in Edit it would take you to the line with a sequence number of 000400, usually the fourth line of a file RENUMbered to the default sequencing. This is useful when you are not sure where you are in the file since you can scroll a given number of lines by typing it in the SCROLL = = = > field near the top of the screen.

Tabbing

Tabbing is probably the most confusing part of ISPF. There are three kinds of tabs in ISPF: hardware, software and logical. Hardware tabbing is never used by programmers and software tabbing is used by ISPF internally to figure out where you might like the cursor placed after you hit Enter. Logical tabbing is normally what you would want to use.

For logical tabbing, you must identify what columns are tab columns and what character you are going to use to save you from using the space bar. You need a keen eye to move to a column using the space bar. For example in Assembler, IBM's standard columns are 10, 16, 35 and 72. In COBOL you might want 12, 16 and 20 to provide a standard block indentation for readability. You normally would not specify column eight because if you use ISPF's COBOL sequence numbering, the Editor will make it appear that column seven is the start of the line.

Use the TABS and COLS Line Commands to display the tab positions. On the line created by the TABS Command, place an asterisk *one column before* each tab position. For example with Assembler, you would put asterisks in columns nine, 15, 34 and 71. The asterisks are one character "off" where you think they should be placed because they mark the *end* of each field.

To start using the logical tab, you must define a logical tab character. If you find the back slash convenient, enter the Primary Command: TABS \

Now you can start using it: \LA\ R15,4\WARNING RETURN CODE

Tabs are part of the Editor's Profile and will be applied to all ISPF Libraries with the same extension.

Sequence Numbers

ISPF sequence numbers are viewed by many programmers as a bother and are frequently disabled. Why? Because the usefulness of numbering as a method of change control is not widely understood. Standard sequence numbers go in the first eight columns of datasets with variable-length record format and the last eight columns for fixed-length record format. In the most common case, 80-byte, fixed-length record format sequence numbers go in columns 73 to 80. Assuming you have STATS ON, the last two digits (columns 79 and 80) indicate the Modification Level of the member when the line was last modified. This is vital information when trying to answer the first question in problem determination, "What changed?"

Before you add sequence numbers to a large file, it is a good idea to check to see if they will be overwriting any useful information. If the file contains JCL, Assembler, SORT or any other IBM utility statements, column 72 should also be checked because any characters there will be interpreted as Continuation Marks. A fast way to check for non-blanks in columns 72 to 80 is with the Picture String of the FIND Primary Command: FIND P"¬" 72 80 (see Figure 1).

To add sequence numbers to a file, use the Primary Command NUMBER ON. Initially the lines will be numbered by hundreds: the first line will be 100, the second 200 and so on. After modification, the file can be resequenced to the original hundred sequence by the RENUM Command. On the other hand, if ISPF is numbering a file that contains data that will be read by a program that expects all 80 columns to be useful data (that is, a SYSIN dataset in JCL), you can blank out the sequence numbers and get ISPF to stop updating them by the UNNUM Command.

If you want to add sequence numbers to all members of a PDS or ISPF Library at once, 3.5 (Reset ISPF Statistics) will do that for you. Using the R option and answering NO to all but RESET SEQ NUMBERS, you can resequence all members in one operation.

Sequence numbers can also be used when you need ascending numbers in your file for other purposes. For example, in an installation without any of the non-IBM packages you would use to make a copy of a tape, you can write an MVS batch in-line PROC that executes IEBGENER with the file number as the parameter and the volume serial numbers of the tapes coded on the header line of the PROC as in Figure 2.

FIGURE 1

Picture String Tips

P"<"	— lower-case letter
P">"	— upper-case letter
P"."	— any character that cannot be displayed
P"-"	— non-numeric character
P"@"	— letters, either lower- or upper-case
P"#"	— single digit, 0-9
P"¬"	— anything but a blank character
P"$"	— special character
P"="	— match any character (for complex patterns)

FIGURE 2

```
//TAPE1FIL   PROC  FILENUM=,INSER=xxxxxx,OUTSER=xxxxxx
//TAPE1FIL   EXEC  PGM=IEBGENER,COND=(0,LT)
//SYSUT1       DD  LABEL=(&FILE,BLP),VOL=(,RETAIN,SER=&INSER),
//                 UNIT=TAPE,DCB=(RECFM=U,BLKSIZE=32760),
//                 DISP=(OLD,PASS)
//SYSUT2       DD  LABEL=(&FILE,NL),VOL=(,RETAIN,SER=&OUTSER),
//                 UNIT=TAPE,DCB=(RECFM=U,BLKSIZE=32760),
//                 DISP=(NEW,PASS)
//SYSPRINT     DD  SYSOUT=*
//SYSIN        DD  DUMMY
//             PEND
```

If you have 120 unlabeled files (or 40 labeled files) on the tape, as I did recently, it is easy to generate 120 lines that say:

```
//COPY1FIL  EXEC  TAPE1FIL,FILE=n
```

But to get the "n" in FILE= to sequence from one to 120 would seem to require a lot of (error-prone) keying. Here is how you can use sequence numbers to do it for you automatically:

- Create a new member
- Enter the NUMBER COBOL Primary Command
- Insert a blank line: I Line Command, then actually type a blank on the line somewhere and hit Enter
- Use the R119 Line Command to create a file with 120 blank lines
- Enter the NUMBER NOCOBOL Primary Command
- Then, BOUNDS 5 *
- On the first line, enter the))99 Line Command (be sure to leave a space after the 99 or position it at the end of the sequence number) and on the last line, the)) Line Command
- Enter the BOUNDS Primary Command with no parameters to reset the column bounds to their original values
- Insert a new line at the top of the file:

  ```
  //COPY1FIL  EXEC  TAPE1FIL,FILE=
  ```

- Enter the COLS Line Command over the sequence number of this new line
- From the column numbers marked, you can determine that the numbers generated should start in column 31. Hence, the numbers need to be shifted right 30 columns
- On the second line, the first line with numbers, enter the))30 Line Command; on the last line of the file, enter))
- Enter the M (Move) Line Command on the first line of the file and O9999 on the second line. This will reMove the first line and Overlay it on to the second line and on all 9,999 lines that follow
- Add a job card and the in-line PROC definition (shown above) before the first line of the file.

Unprintable Characters

As mentioned above, FIND P"." will locate any character in your file that cannot be displayed on your terminal. But there is no need to live in mortal fear of unprintable characters in every file you use. Whenever you edit a file with ISPF, you will receive a warning message if the file contains any unprintable characters. FIND P"." helps because it will give you the hexadecimal equivalent of each character you find. What if you have a whole group of unprintable characters and want to see them all? FINDing one character at a time can be tedious.

HEX ON is a Primary Command in both Browse and Edit that will display each line in both character and hexadecimal form. In Edit, a change to either the hexadecimal digits or character displayed will result in a change to the file — extremely useful for lines with mixed unprintable and printable characters.

Whether or not you have HEX ON, you can locate any unprintable character by its hexadecimal representation. FIND X"16" would locate the backspace control character in either Browse or Edit.

One final word about unprintable characters: ISPF only considers characters with hexadecimal codes less than X"40" (blank) worth bringing to your attention. Although the ISPF Editor will not give you a warning for an X"FF" in a file, it will be displayed as a period by Browse and can be located with the FIND P"." Line Command in both Browse and Edit.

Keeping Old Versions

It is often advisable to keep an old copy of any file you are going to change in case the changes do not work. The Editor allows you to abandon your edit session with the CANCEL Primary Command. It is clear that more comprehensive methods are needed.

One installation uses the convention of adding an @ to the name of the old copy of the member, if working with an ISPF Library and adding it to the second-to-last level of the dataset name otherwise. Where the member or level is already eight characters in length, the last character is replaced by the @. In VM, the @ is added to the *filename*.

The easiest way to create the backup member is with the Editor just before you change the file. Use the Line Command C9999 on the first line of the file and the CREATE name@ Primary Command. However this approach may eliminate valuable information maintained by ISPF Statistics.

For each member, ISPF Statistics tell you:

- Creation date
- Date and time of last modification
- The user ID doing the last modification
- The number of times the member has been modified
- The number of lines in the member when it was created
- The number of lines in the member now
- How many current lines have been changed or added.

This information is stored in the directory of the ISPF Library, but only if you have STATS ON (the default) when using the Editor to create or modify the member. Typically, members without statistics have been created by IBM or user programs rather than ISPF.

Statistics do take space and creating them for an MVS PDS (which you can do for all members at once using 3.5 — Reset Statistics), may require an increase in the number of directory blocks allocated for the PDS. With Statistics, you will need one directory block for each six members.

To preserve Statistics for both the new and backup versions of a member you are going to change, rename the member adding the @ sign and then use 3.3 to make a copy of the member without the @. The copy of the member without the @ can then be edited.

Global Commands

Up to now, we have concentrated on the ISPF Editor and Browse. The commands and techniques that follow can be used anywhere in ISPF (see Figure 3).

Option 6 is not the only place where you can issue a TSO Command. If you precede any TSO Command with "TSO," you can execute that command from the COMMAND ===> line of any ISPF screen. Following its completion, you are back where you started in the same ISPF panel. TSO STATUS can be used to save you from moving into 3.8 every time you want to find out the status of a batch job you are running. In VM ISPF, you would use "CMS" before any valid VM/CMS Command.

PRINT-HI and PRINTLHI will send copies of the ISPF panel you currently have displayed to the ISPF List File for later printing. The HI indicates that any highlighted portions of the screen will be double printed, making them *bold* on most printers. PRINT-HI prints the screen as you see it while PRINTLHI is useful in Split Screen Mode where it prints the complete *logical screen* of the *current side* of the Split Screen.

PF7 (UP) and PF8 (DOWN) are the normal ways of scrolling in ISPF. In the SCROLL ===> field in the upper right-hand corner, you can specify how much you want to move backward or forward in the file or list. PAGE will move you a full screenful of lines; DATA, one line less than a screenful; HALF, a half screenful. Specifying a number moves you that number of lines. CSR (cursor) will make the line where the cursor is the top (for PF8) or bottom (for PF7). If the line is already at the top or bottom, then the movement will be a full screenful of lines.

The advantage of using PAGE, HALF, DATA and CSR rather than a number is that they adjust automatically when the screen size changes as it does if you use Split Screen or a 3270 terminal that allows more than 24 lines on the screen.

You can also type MAX in the SCROLL ===> field to get to the top or bottom of the file. A much faster way of doing this is to type the letter M on the COMMAND ===> line and hit PF7 or PF8. This works because ISPF takes the PF key meaning and concatenates it with anything found on the Command Line after inserting a blank. DOWN M is a legitimate command, "M" being a legitimate abbreviation of MAX.

Moving Around Fast

The characters = . and ; help you move around in ISPF more quickly. Although PF4 takes you from any ISPF screen to the ISPF Main Menu, the = indicates that you want to use the Main Menu as your starting point. The number or letter following the equal sign is the item on the Main Menu you want to choose. For example, =X would indicate you want to immediately exit ISPF (or terminate Split Screen Mode), while =3 would take you to the Utilities Menu. A period, and another number or letter following, indicates that you also want to specify a menu choice from the menu displayed as a result of your (first) choice from the Main Menu.

Assuming ; is the special character you choose for a Command Delimiter on the 0.1 ISPF Terminal Characteristics Panel, you can do the equivalent of several entries on the COMMAND ===> line at the same time. This is especially useful in a dial-up or other situation in which the data communications speeds are slow (that is less than 9600 baud). For example, =2;;SEL FASTCOPY;CHANGE "CLASS=Q" "CLASS=P";SUBMIT; CANCEL;=3.8;L would submit a batch job after temporarily (CANCEL) altering the job class and then check on its status. Because you cannot use PF keys in the middle of the line, you will find yourself using some unfamiliar commands: the Primary Commands you normally hit PF keys to do. If you are using Split Screens, you will be surprised what you can do on one line if you include the SWAP Primary Command (PF9).

FIGURE 3

Global Command Tips

TSO — issue a TSO Command from the COMMAND Line of any screen
CMS — issue a CMS Command from the COMMAND Line of any screen
PRINT-HI — print the physical screen as you see it with bolding
PRINTLHI — print one side of a Split Screen with bolding
PF7 — scroll up the amount shown in the Command Line or Scroll Field
PF8 — scroll down the amount shown in the Command Line or Scroll Field
=m.n — go immediately to menu choice "y" on Main Menu item "x"
=X — immediately exit ISPF; terminate one side of a Split Screen

When You Are Stuck

The most common way to get stuck in the ISPF Editor is to try to insert characters into an existing line with NONULLS in effect (the default) or to insert characters in a newly-created line even with NULLS in effect (see Figure 4). ISPF fills all spaces on the line with blanks even after the end (last non-blank character) of the line. Trying to insert characters on a 3270 terminal gives you an error that varies depending upon the model of 3270 you have. Often, a stick man followed by a > is displayed in the Status Line at the bottom of the screen. Protocol convertors typically beep at you. A similar situation also exists when you forget you are in Browse and try to overtype characters in the file. The error displayed is different but the result is the same: you cannot type anything more until you clear the condition. The solution, too, is the same. Hit the RESET key on the 3270 terminal.

What happens if you want to redisplay the entire screen? ISPF is smart: it will only write out those characters that need to be changed from its last write and your last entry on the screen. Data transmission errors may have given you a different screen than ISPF thinks you have. Hitting the PA2 key on a 3270 terminal will get ISPF to redisplay the entire screen. PA2 is also a good way to solve any of the following problems:

- You regret the changes you have typed since the last time you hit ENTER or one of the PF keys
- You accidentally hit the ERASE INPUT or CLEAR key on the 3270.

PA1 can also be hit when you are in Option 6 (Enter TSO Command or CLIST) to halt the command or CLIST, but it is generally inadvisable elsewhere in ISPF.

In a desperate situation in which you are really stuck and getting no response at all, you can force the system to log you off, stopping all processing you have requested. Hit the SYSTEM REQUEST key on the 3270. This key is often labeled SYS REQ and its use usually requires holding down the Alternate (ALT) key while hitting it. Then type LOGOFF TYPE(UNCOND). Note that this is a VTAM command and assume that you are running VTAM to control your terminal (which almost all installations do).

FIGURE 4

Help When You Are Stuck

NULLS Primary Command — allows you to insert characters in a line
RESET key — clear "Input Inhibited" to allow keyboard input again
PA2 key — redisplay the entire screen
PA1 key — to stop TSO Commands that are processing
SYSTEM REQUEST key, then LOGOFF TYPE(UNCOND) — to sign off

Performance

For performance reasons, a large number of installations do not allow their MVS applications programmers to use ISPF under TSO. Instead, they are restricted to VM/CMS or a CICS-based editor and job submission/retrieval package. However other shops have found that, used with a

See ISPF Techniques *page 91*

Soon "Lights Out" will be the standard way of running large MVS datacenters. The service improvement and cost reduction available from an aggressive Automated System Operation implementation effort is too large to ignore.

OPS/MVS from MVS Software is the only product that delivers the technology *today* required to achieve the promise of Automated System Operations (ASO).

OPS/MVS earned The Gartner Group's rating as "the top gun" Automated System Operations product because it has the functions and facilities that are required to meet the demands of unattended operations:

REXX language... Rule-based Automated Operations Facility... Interative automation testing support... Automation audit trail... OMEGAMON® interface... Message text independence... MCS console enhancement... Programmable Batch Operations Interface... Full-screen, menu-driven operations interface... VM Guest Support... JES3 support... Multi-system communication... Console Consolidation Facility... Single point of control... Outboard PC programmed in REXX... Remote operation... Pager support... IMS/DC and CICS Operation Facilities... Expert System Interface... MVS/ESA support.

Our customers, with configurations varying in size from a single 4381 to multiple 3090-600s at multiple sites, agree: OPS/MVS is not only the most powerful product available; OPS/MVS is also the easiest product to use. And OPS/MVS is the only product designed for *unattended* operation of a major datacenter.

OPS/MVS prices start at $9,500 for a 4381 running MVS/JES2. For more information on OPS/MVS, including pricing for your site's CPU configuration, contact us at 12555 West Jefferson Blvd., Suite 221, Los Angeles, California 90066 or call **213-578-1147**.

MVS SOFTWARE, INC.

The Automated System Operations Specialists™

CIRCLE #31 on Reader Service Card ▲

ESA Implements Expanded Addressing

By Michael Haupt

Enterprise Systems Architecture or ESA, IBM's new large system architecture, dramatically expands virtual addressing capabilities under MVS/XA. A previous article, "ESA Is On The Way," in the January, 1989 issue of MAINFRAME JOURNAL, described how this capability equips large installations to cope with processing requirements of the next decade. This article will examine how ESA implements this expanded addressing.

The new ESA/370 architecture is implemented in the Enterprise System/3090 "E" and "S" Models and 4381's "E" Models. These processors contain additional hardware, new registers and the enriched instruction set required to run ESA.

The ESA architecture implements expanded addressing through new facilities collectively known as the Advanced Address Space Facility (AASF). The AASF facilities provide access to greatly increased quantities of virtual storage. They simplify and improve the reliability of complex inter-address space communication. This article probes several of these AASF facilities namely the data space, Address Space Control (ASC) modes, access registers and the linkage stack facility.

Data Space

One innovative concept in ESA is that of the data space. Programs can easily refer to data outside of their address space. Data spaces provide application programs immense data addressability allowing them to break out of the 2GB virtual storage limit imposed by the 370/XA architecture. Under ESA, a program can address 16TB (terabytes) that is 8,000 times greater than XA's limit.

Data Space Structure

A data space is similar to an address space with 2GB of virtual storage. The data space contains nothing but data. Executable code can be stored in a data space but must be moved into an address space before being executed.

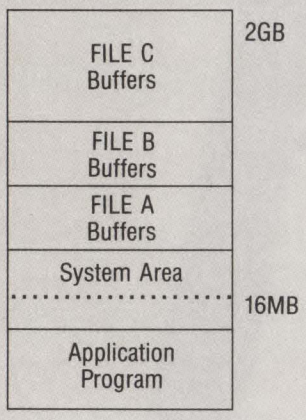
Figure 1 shows a simplified diagram of an application in an address space. The address space contains 2GB of virtual storage. Part of this storage, on both sides of the 16MB line, is used by the System Area for the nucleus, LPA, SQA and CSA. The application program's executable code occupies storage in the address space. The remaining virtual storage is available for data buffers for all files used by the program.

In the simplified diagram, 2GB provides plenty of storage for both the program and an adequate quantity of buffers for three files. However, if this address space contains a subsystem, such as CICS, the available storage could be shared by buffers for a hundred or more files.

Figure 2 illustrates the same application under MVS/ESA. Two files now reside in

FIGURE 1

Virtual Storage Layout
MVS/XA Address Space



their own data spaces. The application program has full access to both data spaces as well as buffers for the other file in its own address space.

As you can see in Figure 2, the entire 2GB in the data space is devoted exclusively to data. No space is sacrificed to the system area or program code.

Data spaces can be created and used directly by Assembler programs. High-level languages use data spaces through Data Windowing Services. A future article will explore more fully the ways data spaces satisfy a variety of needs.

Data Space Benefits

Each data space contains data from only one file. This data isolation helps prevent it from being corrupted by problems in the application's address space.

Data spaces also improve an application's performance. When a program needs data that is not in a buffer, the access method performs an I/O. As seen in Figure 1, an address space can devote a limited amount of virtual storage for each file's buffers. The result is a large amount of access method I/O.

A data space, on the other hand, is nothing but data buffers. This drastically decreases access method I/O. A file with less than 2GB of data can be completely contained in the data space, eliminating access method I/O during processing.

Because a data space consists of virtual storage, access method I/O may be replaced by paging. While paging is considerably faster, it impacts the entire system. ESA's performance, especially when using data spaces, will be even more dependent on adequate paging resources than XA.

Address Space Control (ASC) Modes

ESA reduces the difficulty a program has in communicating outside its address space by introducing a number of ASC modes.

Evolution of Inter-Address Space Communication

ASC modes are an extension to current Cross Memory Services (XMS). To understand them, look briefly at the development of XMS.

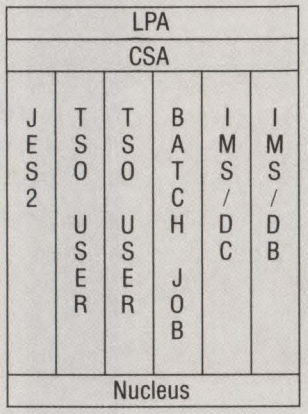
Early MVS

One of the main features of the 370 architecture and MVS/370 is the integrity provided by the address space structure as shown in Figure 3.

A separate address space is provided to each user whether batch job, TSO session

FIGURE 3

MVS/370
Virtual Storage Layout



or major subsystem such as JES, IMS/DC and IMS/DB. Some storage, such as the nucleus, Link Pack Area (LPA) and Common System Area (CSA) is used by all address spaces and placed in a Common Area.

Address spaces are in the Private Area, protecting each user's program and data from the others. Each address space has its own unique segment table controlling storage. A Control Register (CR 1) points to the active segment table. MVS controls which address space's storage is used by changing the pointer in the control register.

This architecture has a number of benefits.

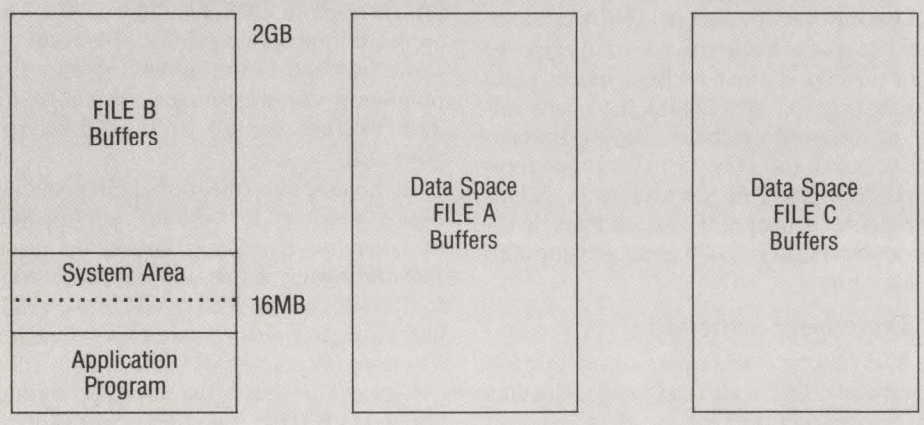
- A single copy of common storage serves any number of address spaces.
- Each address space can use the maximum range of virtual addresses in the Private Area. Available virtual storage does not vary with the number of users (address spaces).
- No address space can encroach on any other address space. Each address space can only access storage pointed to by its segment table.

Communication between address spaces is possible but only indirectly. For example, the following steps take place for IMS/DB to fill a request by IMS/DC:

1. IMS/DC places a data request in the CSA
2. IMS/DC creates a Scheduler Request Block (SRB) requesting the IMS/DB address space to process the data at the CSA address
3. MVS processes the SRB and passes the address to IMS/DB
4. IMS/DB processes the request, placing the result in the CSA

FIGURE 2

Virtual Storage Layout
MVS/ESA Address Space with Two Data Spaces



5. IMS/DB creates a SRB informing IMS/DC of the result of its request
6. MVS processes the SRB and passes the address to IMS/DC
7. IMS/DC retrieves the result from CSA.

As you can see, inter-address space communication in early versions of MVS was complicated. Increased system activity revealed a number of other limitations of the single address space architecture.

- The amount of available virtual storage is limited.
- Storage protection is provided by protect keys associated with an address space.
- Shared code and data reside in common storage. In exchange for performance gains, common storage reduces virtual storage available to address spaces and sacrifices storage protection within the common area.

Cross Memory Services

To avoid problems like these, MVS/SP introduced Cross Memory Services (XMS).

XMS simplifies communication between two address spaces. Bill Carico provides an in-depth description of XMS in a series of articles in MAINFRAME JOURNAL beginning in July/August, 1988.

Three separate services are provided by XMS.

- Data Movement: data can be moved directly between two address spaces without the SRB/CSA overhead.
- Data Sharing: a program in common storage can reference data in two address spaces.
- Program Sharing: a program executing in one address space branches to a program in another address space.

MVS/SP and XA use a dual address space structure to provide these Cross Memory Services.

Each address space uses two control registers that can point to separate segment tables. As with base MVS, CR 1 identifies the Primary segment table. Another control register, CR 7, can point to an alternate or Secondary segment table. A bit in the PSW (bit 16) indicates whether the Primary or Secondary Control Register is used for data addressing.

This dual address space structure provides a number of advantages over the earlier MVS structure.

- Performance of inter-address space communication substantially improved.
- Some Virtual Storage Constraint Relief (VSCR) is provided by an address space's ability to access data in another address space. It is also provided by the movement of code and data from the common area into private address spaces.
- Improved integrity as code and data move from common areas into isolated address spaces.

XMS's dual address space structure still has a number of limitations.

- Virtual storage remains constrained because only two address spaces (Primary and Secondary) can be used simultaneously and common storage consumes a portion of each address space.
- The typical address space contains both code and data, compromising integrity.
- While conceptually easy to comprehend, the instructions to implement XMS, as Carico's articles demonstrate, are complicated and difficult to use.

Address Space Control (ASC) Modes

ESA/370 provides a multiple address space structure that is an extension of XMS. Two PSW bits govern the ASC mode which designates the control register to be used when executing the current instruction. Three control registers are used to point to different segment tables. Table 1 summarizes the different ASC mode possibilities.

TABLE 1

Address Space Control Modes

PSW 16	Bit 17	ASC Mode	Control Register
0	0	Primary	CR 1
0	1	Secondary	CR 7
1	0	Access Register	none
1	1	Home	CR 13

Maintaining compatibility with XMS, bit 16 switches between Primary and Secondary ASC modes. The use of bit 17 is new with ESA. When on, bit 17 indicates either the use of the Home address space's control register or the use of another new facility, Access Registers (ARs).

When in the Primary, Secondary and AR ASC modes, instructions are fetched from the Primary address space. In the Home ASC mode, instructions are fetched from the Home address space.

The ASC mode can be changed in several ways including the new SAC instruction (Set Address Space Control). Programs in the problem state can use SAC to change between Primary and AR modes. However, a program must be privileged to change to the Home mode.

ESA's multiple address space structure and ease of ASC mode switching is an improvement over previous techniques. The real power in ESA comes from its new AR mode.

Access Register Translation (ART)

A program's ability to reference multiple address and data spaces is implemented by new hardware in ESA/370 computers that provides Access Register Translation (ART). ART is analogous to the DAT function that provides Dynamic Address Translation for virtual storage.

ART redirects virtual addresses prior to DAT processing. An address in a program running in one address space can refer to an address in another address or data space.

Access Registers

ESA/370 computers provide an additional set of 16 registers called Access Registers (AR). These registers are used to identify an address or data space for data fetching.

Before ESA/370, with the exception of the XMS instructions, instructions and data always reside in the same address space. Data addresses are computed using the Base-Displacement Addressing technique. The Base is an address (either 24 or 31 bit) held in one of the 16 General Purpose Registers (GR). Figure 4 shows an instruction at address 1,000 moves 10 bytes from address 4,000 to 6,000.

ARs are paired one-to-one with the GRs. An AR associates the base register with an address or data space. Using ARs, a program can simultaneously access 15 address or data spaces.

ARs are used only for addressing. They cannot be used for arithmetic or indexing operations. Also, ARs are used only for data fetches, NOT instruction fetches. A program cannot branch across address spaces using an AR.

When in AR ASC mode, as shown in Figure 5, the AR identifies the address space to which the base register applies.

Here is the same MVC instruction running in Address Space "A." The contents of the GRs are unchanged. However, the contents of AR 5 specify Address Space "B." The source of the data to be moved is address 4,000 in Address Space "B." Sim-

ilarly, the destination is redirected, by AR 6, to location 6,000 in Data Space "C." The AR specifies the target address or data space by means of an Access List Entry Token or ALET.

Access List

The Access List is a table that defines the address and data spaces that a task can access using ARs. The AR contains a token that serves as an index to an entry in the Access List.

The Access List can hold up to 253 entries. Since each entry refers to a 2GB address or data space, the Access List provides the program with the ability to reference 16TB of virtual storage.

ARs use the Access List for every data fetch when in AR ASC mode. Special ALETs, shown in Table 2, are used for frequently used address spaces. This bypasses AR translation overhead for most data references in AR mode without forcing the program into the tedious chore of constant mode switching.

Tokens

Address spaces are identified in the Access List by a Space Token or STOKEN. A STOKEN is conceptually similar to the ASID (Address Space Identifier). There are some notable differences between the two. The STOKEN identifies both address and data spaces; ASIDs only refer to address spaces. Also, the STOKEN is 64 bits long, permitting more unique identifiers than the 16-bit ASID. In addition, the STOKEN remains unique throughout the life of the IPL. Unlike an ASID, a STOKEN will not be reused, eliminating many operational difficulties.

IBM's stated direction is to use STOKENs to identify address spaces to MVS services instead of ASIDs or ASCB addresses. The STOKEN is part of a larger token for each task called a TTOKEN or Task Token.

TTOKENs are 128-bit identifiers that describe a task. Like STOKENs, they are unique for the life of the IPL and are not reused as are TCB addresses. These new tokens make it possible to verify that the intended address space and task have not terminated.

Access List Types

There are two types of Access Lists: the Dispatchable Unit Access List (DU-AL) and the Primary Address Space Name Access List (PASN-AL). A program can acquire an ALET from either of its lists.

MVS has a unique DU-AL for each work unit (TCB or SRB). Problem state programs can add and delete entries on its DU-AL. Every MVS address space also has a unique PASN-AL. Updating the PASN-AL is limited to programs in supervisor-state or with Key 0.

This access list structure is a security feature that provides for the separation of data from problem programs while permitting major subsystems access to that data.

FIGURE 4

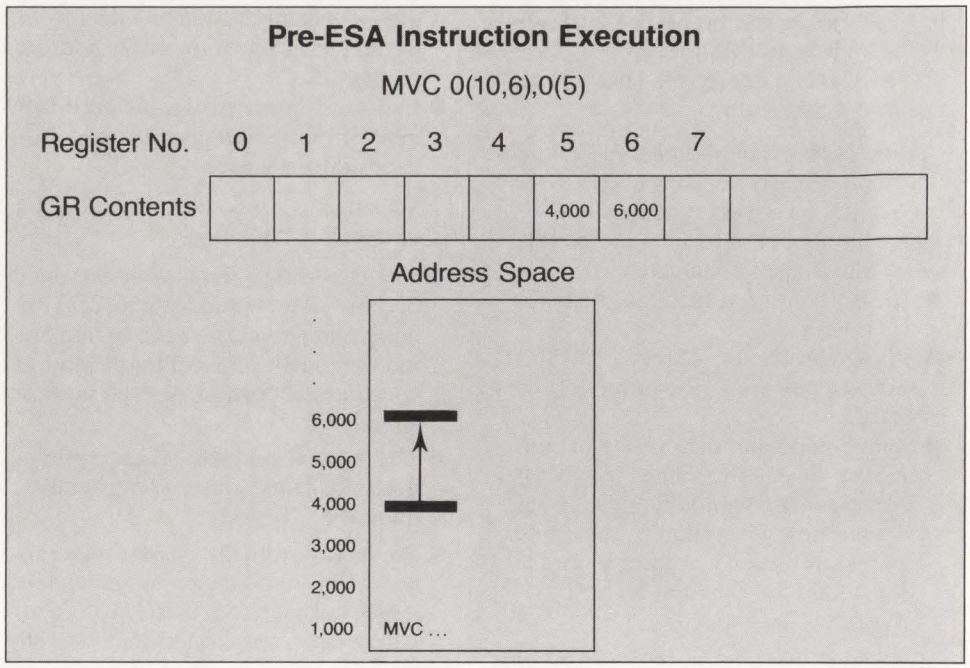


FIGURE 5

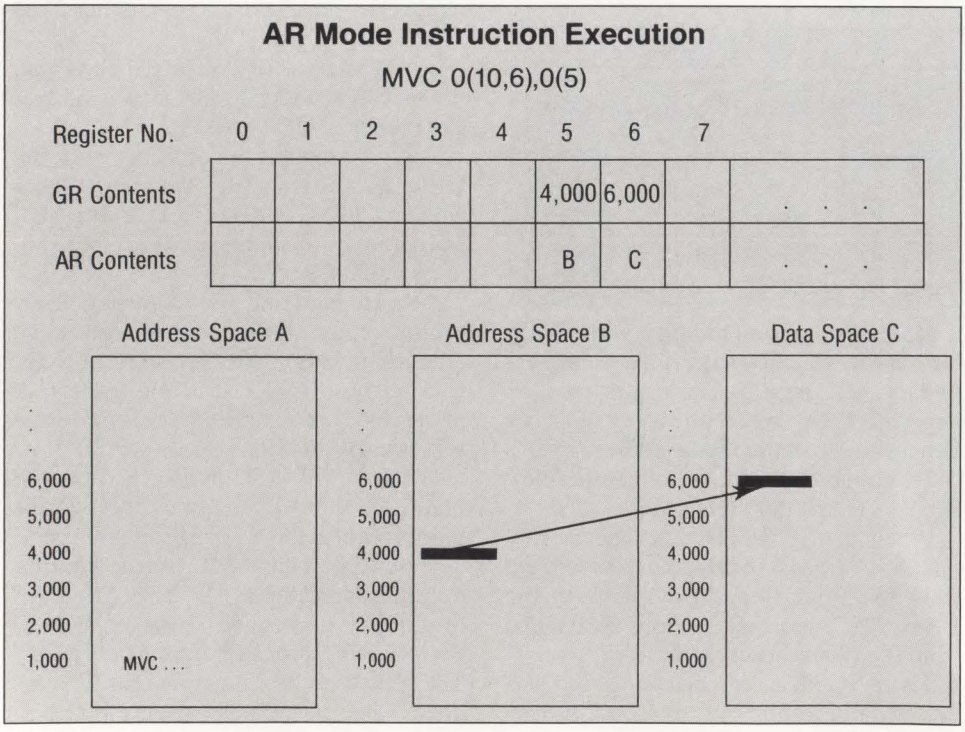
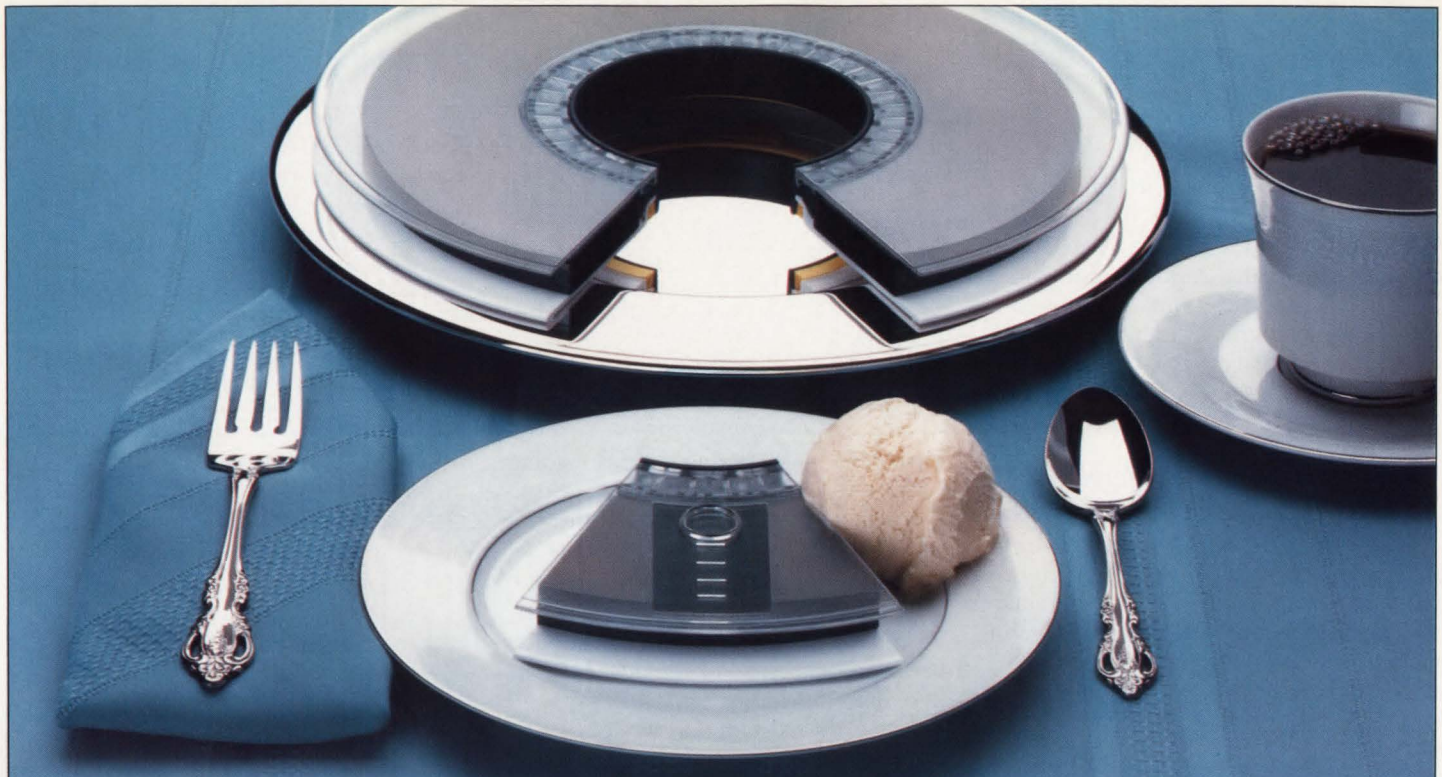


TABLE 2

Special ALET Values

ALET Value	Address Space	Control Register
0	Primary	CR 1
1	Secondary	CR 7
2	Home	CR 13



When choosing a network access package...why settle for just a piece of the pie?

If all you want is access to your network, a number of software vendors can help you. But, if you want more than just a piece of a product, you need NCI from Westinghouse.

Network Control Interface gives you the power of total network control. It is the only network access and control product available for MVS, MVS/XA, VSE and VM operating systems. In fact, regardless of the size and complexity of your VTAM network, NCI provides all your users with a single system image, with identical entry screens and procedures.

Of course we give you single point entry for quick application access at the push of a single key. That's the easy part. But we also provide application load balancing, logon queueing, unlimited exit processing and a dialog manager...powerful features to improve your productivity...without sacrificing system performance.

NCI can read the contents of your security data base and dynamically build user menus, listing only applications the user is authorized to access...reducing user confusion and simplifying security administration.

With NCI, you can customize menus to meet your specific user needs, or use some of the over 100 sample panels provided in our Starter System...the options are unlimited.

To truly appreciate the power of NCI, we invite you to call or write us for a free, 30-day trial.

More than just simple network access...NCI from Westinghouse.

Westinghouse
SOFTWARE **SOLUTIONS**

Westinghouse

Management Systems Software
P.O. Box 2728
Pittsburgh, PA 15230
(800) 348-3523
(412) 256-2900 in PA

CIRCLE #128 on Reader Service Card ▲

Extended Authorization Index (EAX)

MVS controls the ability to build and use access list entries through the Extended Authorization Index (EAX). The EAX is an extension to the Authorization Index (AX) used by XMS. Authorizations for different tasks running within the same address space can vary. EAX controls apply only to address spaces. EAX authorization is not required for data space access.

Instructions and MVS Services for AR

AR mode processing is designed to be as transparent as possible. However, it is a major extension to the 370/XA architecture. As such, there are effects in terms of machine instructions and MVS macros services.

AR Instruction Support

As described earlier, ARs are used to redirect addresses when retrieving data. Because this redirection is implemented in the data fetching hardware, the entire instruction set, including vector instructions, gains the addressing benefits.

An AR is used only when it corresponds to a base register. If the GR is used as an index register, the AR is ignored. A few new instructions, listed in Table 3, provide AR control mechanisms such as loading, storing and copying. The new AR instructions run in problem state. They can be executed whether or not the program is in AR mode.

TABLE 3

New AR Instructions

Name	Function	Description
LAM	Load Access Multiple	Load multiple ARs from storage
STAM	Store Access Multiple	Store multiple ARs into storage
SAR	Set Access Register	Set an AR from a GP register
EAR	Extract Access Register	Set a GP register from an AR
CPYA	Copy Access	Copy on AR into another AR
LAE	Load Address Extended	Load address and set AR according to mode
TAR	Test Access Register	Determine whether ART will take place

New MVS Services

MVS also has a number of new services that support the use of the advanced address space functions. Table 4 shows some of the more important new services.

Other MVS Services

Existing MVS services, unless explicitly permitted, cannot be invoked when the program is in AR mode. This restriction applies to all services whether invoked by an SVC or a branch entry. The program must issue the SAC instruction to switch to Primary mode before using the service.

MVS is gradually changing the services to support AR mode callers. Services that support AR mode invocation are suffixed with an "X." Below is a list of MVS services that already support AR mode:

ATTACHX	SNAPX
ESTAEX	XCTLX
LINKX	SYNCHX
SDUMPX	

TABLE 4

New AR Related Macros

Name	Function	Description
ALESERV	ALE Service	Controls of Access List Entries such as adding, deleting, searching and extracting
TESTART	Test ART	Determines whether an ALET/EAX combination is currently valid without using it for reference
TCBTOKEN	TCB Token	Obtains the TTOKEN for a task; Verifies that the subject task has not terminated
LOCASCB	Locate ASCB	Converts an STOKEN to an ASCB address
ATSET	Authority Table Set	Grants authority to users of a specific EAX
AXRES	Authorization Reserve	Reserves a value to be used as an EAX
STORAGE	Storage	A front-end to GETMAIN/FREEMAIN to obtain and free storage within AR qualified address spaces

Linkage Stack Facility

ESA offers yet another feature to reduce complexity of program linkages thus simplifying multi-address space environments. The processor, instead of software, saves and restores linkage registers and status in a hardware-controlled stack. Besides being faster, this new facility improves the reliability of complex linkages.

Linkage Stacks

The Linkage Stack (LS) is a secure area provided by MVS for each dispatchable unit (TCB or SRB). It is used to save a program's execution and state information across a linkage.

LS Storage

LS storage is allocated in 4K sections each containing up to 24 entries. LS sections can be chained together if more than 24 entries are needed.

This chaining permits LS storage to be acquired dynamically. It avoids two problems common to stack structures: the requirement for preallocated, contiguous storage; and a fixed, limited number of entries on the stack.

Linkage Stack Entries (LSE)

Linkage stacks have three types of entries: Header Entry, Trailers Entry and State Entry.

There is one header and trailer for each 4K LS section enabling the chaining.

The State type LSEs are 168 bytes long and contain status information at the time of a program linkage. This information includes: GRs 0-15; ARs 0-15; PASN & SASN; and Current PSW.

Stack Handling Instructions

Table 5 summarizes instructions that use the linkage stack.

TABLE 5

Linkage Stack Instructions

Name	Function	Description
PC	Program Call	Transfer control to another program after optionally stacking execution information
BAKR	Branch and Stack	Stack execution information and branch
PR	Program Return	Restore execution information and return
EREG	Extract Registers	Loads specified AR or GP registers from the last stack entry
ESTA	Extract Stacked State	Obtain state information (PSW, PASN, SASN) from the last stack entry
MSTA	Modify Stacked State	Saves the contents of an even/odd register pair in the last stack entry

Program Call (PC) Stacking

The PC instruction has a new stack option. When requested, the PC instruction automatically creates a state type LSE saving the GRs, ARs and other status information. The instruction may also replace the PSW-key, PSW-key-mask and the EAX. When the stack option is not specified, the PC is compatible with existing software.

Branch and Stack (BAKR)

The new BAKR instruction is a variation on the other linkage instructions: BALR and BASR. BAKR allows the linkage stack to be used on branch linkages within a single address space. It stacks the caller's registers and status.

Program Return (PR)

PR is a new instruction that provides the return mechanism from a stacking call. It restores registers and status from the stack, removes the LSE and returns to the caller.

Registers (both GR and AR) 15, 0 and 1 are not restored from the stack. Since these registers are primarily used for parameter lists, return codes and the like, PR preserves their values.

Other Linkage Stack Instructions

The final three instructions, EREG, ESTA and MSTA provide additional functions of retrieving or modifying selected information in the LS. These instructions work with the current LSE, leaving the stack pointer unchanged.

Associated Recovery Routines (ARR)

The PC instruction's enhanced linkage can identify an Associated Recovery Routine (ARR). It eliminates the overhead of establishing and cancelling recovery environments for subsystem programs entered using the PC instruction.

ARRs are like automatic ESTAE recovery routines. The ARR is in effect while the PC routine is executing. It is not necessary to create or delete the recovery routine with each call and return.

ARRs can be stacked and percolated along with ESTAE routines.

To establish an ARR, use the new ETDEF (Entry Table Descriptor Definition) macro. ETDEF generates the PC Entry Table Descriptor specifying PC options such as: ASC mode, EAX, SASN as well as the ARR address. Once defined, the descriptor is in effect for subsequent PCs.

Summary

Some of ESA's Advanced Address Space Facilities were examined in this article. These facilities provide capabilities to address vast quantities of data required for forthcoming applications. AASF also simplifies and improves the reliability of multiple address space operation, a key element in the future environment.

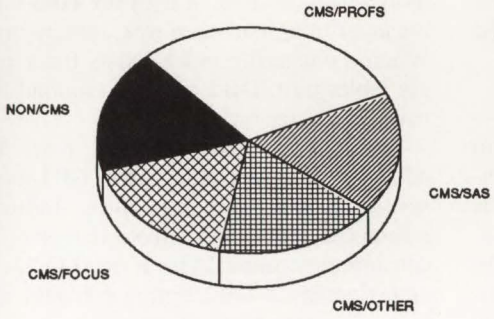
A future article will explore data spaces and hiperspaces and how MVS/ESA uses them to improve performance while increasing reliability and availability.

ABOUT THE AUTHOR

Michael Haupt, CDP, is a senior associate with Austin Consulting in Rosemont, IL. During his 15 years in data processing, he has worked extensively with large scale IBM systems and has been active in professional associations and user groups. Austin Consulting, 9801 W. Higgins Rd., Rosemont, IL 60018, (312) 696-0500.

Please Remember To Fill Out The Reader Service Card Located Between Pages 58 And 59

VM Usage



Now that you know that CMS is using most of your VM system...

Don't You Need To Know How CMS Is Being Used?

With VM◪CMAP, VM◪CMAP/FOCUS, VM◪CMAP/PROFS and VM◪CMAP/SAS, you can get information on user and command activity, response time analysis and other vital information required for effective performance analysis of your VM system.

VM◪CMS Unlimited, Inc.
A subsidiary of
First Alliance Software & Technologies, Inc.
First Alliance Plaza
11770 Bernardo Plaza Court
San Diego, California 92128

For Info Call: 800-527-3237
In California: 619-487-8030

CIRCLE #64 on Reader Service Card ▲

DB2 REORGS?

Don't Waste Your Time!

By Bill Cunningham

Is IBM's REORG utility the answer to degraded DB2 performance caused by disorganized data? Only if you are prepared to spend two, four or even eight hours running the REORG and if you can find a window when the data is not needed so the job can be run.

The DB2 REORG Utility

Disorganized data is a major problem contributing to degraded DB2 performance and higher resource utilization. It is also one of the easiest problems to solve. IBM designed the DB2 REORG utility to correct the problems caused by disorganized data. However disorganized data has these same adverse effects on the REORG utility itself. For example: the DB2 REORG utility requires one and one-half hours to reorganize one million rows of 15-percent disorganized data. It takes just 12½ minutes when this data is already organized.

The DB2 REORG utility has six phases, each processed serially:

- UTILINIT — utility set up and initialization
- UNLOAD — writes all data in the tablespace to a sequential dataset
- RELOAD — loads the data back into the tablespace and also extracts index keys
- SORT — sorts index keys extracted by RELOAD
- BUILD — builds all indexes if any exist
- UTILTERM — utility clean-up and termination.

The UNLOAD phase uses most of the reorganization time. It uses the DB2 data manager to unload each row, one by one. When a clustering index exists for a single tablespace, DB2 uses it to unload the rows in cluster order.

A problem occurs when the rows are taken out of cluster order. Then DB2 reads several pages more than once. Indirect references make this situation worse by causing even more I/Os. Extra I/Os generate increased DB2 buffer manager activity that affects all users.

If no clustering index exists or if the tablespace contains multiple tables, DB2 unloads the rows using a sequential scan. This scan is faster than using an index but the rows are not reloaded in any particular order (no clustering). This process also increases buffer manager activity.

The UNLOAD phase writes the rows to a sequential file that is, in turn, read during the RELOAD phase. Reloading the data consists of reading data from the UNLOAD file and rebuilding the data pages of the tablespace. This process is relatively faster than UNLOAD but still uses the DB2 data manager.

Another task of the RELOAD phase is to create the work file used to build the indexes. It extracts the keys for all defined indexes and writes a record for each to the work file. A single table, containing 500,000 rows with a clustering index and two secondary indexes, creates 1,500,000 work file records. Each record in the work file contains the index number, the key value and the new RID of the associated row.

The SORT phase sorts the records by index number and key value. This sort arranges, in order, all of the records for the first key then the second key and so on. When using a clustering index to unload the rows, some redundancy occurs since the data is already in clustering key sequence.

The BUILD phase uses the output of the SORT phase to build all of the indexes associated with the tablespace. It builds each index, one at a time, using the key value and the new RID assigned during RELOAD. Typically, the BUILD phase is shorter than RELOAD. However a tablespace with many indexes will spend considerable time in the BUILD phase.

So, what benefit can be derived from the time and resources required for reorganizations?

DB2 performance problems can be resolved, although the solution is temporary if the tablespace is subject to constant updating. Also, the data is available READ-ONLY during UNLOAD and completely inaccessible during the remaining phases.

Periodic data reorganization is inevitable; therefore, shortening the time it requires becomes essential.

Improving the DB2 REORG Utility

Optimal performance by maintaining organized data requires DB2 users to balance degraded performance against time needed for reorganization. Another problem, deciding when to reorganize data, develops and this is not an easy problem to solve. Ultimately the solution is twofold: to have sufficient statistical data to predict the need for REORG and to significantly lessen the time it takes to do reorganizations by using a fast REORG utility.

This fast REORG utility uses the standard tools made available by IBM through the MVS/XA operating system. These tools are: VSAM control interval processing, track length buffers, efficient blocksizes and multitasking. Intelligent processing and these tools go a long way to reduce the time required to complete the reorganization of a DB2 tablespace.

In using these tools, the area with the most opportunity for improvement is I/O. DB2 has an excellent data manager that supports a buffer pool of unprecedented size and uses state-of-the-art management techniques. However it suffers from the drawback of being a general-purpose I/O subsystem. It must serve multiple users, each with a different purpose. It also executes in a separate address space from the application where the data is needed. This general-purpose cripples a highly-specific function like REORG.

A single-purpose, efficient I/O design provides substantial improvements over the data manager. Since the intent is to improve a single function, the reorganization of tablespaces, specialized I/O is essential. The foundation for this improved I/O subsystem is VSAM asynchronous control interval processing with sequential read-ahead.

When a tablespace contains a single table, DB2 uses the clustering index, if defined, to unload the rows. It also follows indirection when encountered. Since a REORG is being run, the data is either out of cluster order, has substantial indirection or both.

These conditions cause DB2 to access pages in a random order. They also cause it to read pages more than once. Avoiding use of the clustering index and ignoring indirection allows unloading each dataset of a tablespace sequentially. Sequential I/O coupled with appropriate buffering allows VSAM to show its true colors as a high performance access method.

Proper buffering is an essential component of high speed VSAM I/O. Checking the device type on which the dataset resides allows determination of the number of control intervals per track. Using this many data buffers means that the number of EXCPs required to read the data will equal the number of tracks containing data. Assigning twice this number allows VSAM to do sequential read-ahead. Sequential read-ahead reduces the number of waits issued for I/O completion. Cylinder size buffering will reduce the number of EXCPs but is detrimental to the goal of reducing elapsed time. VSAM spends more time in buffer management and, therefore, significantly less memory is available for other uses.

If the tablespace is partitioned, the use of multitasking and parallel processing provides an even greater improvement.

FIGURE 1

Unloading Multiple Partitions Concurrently

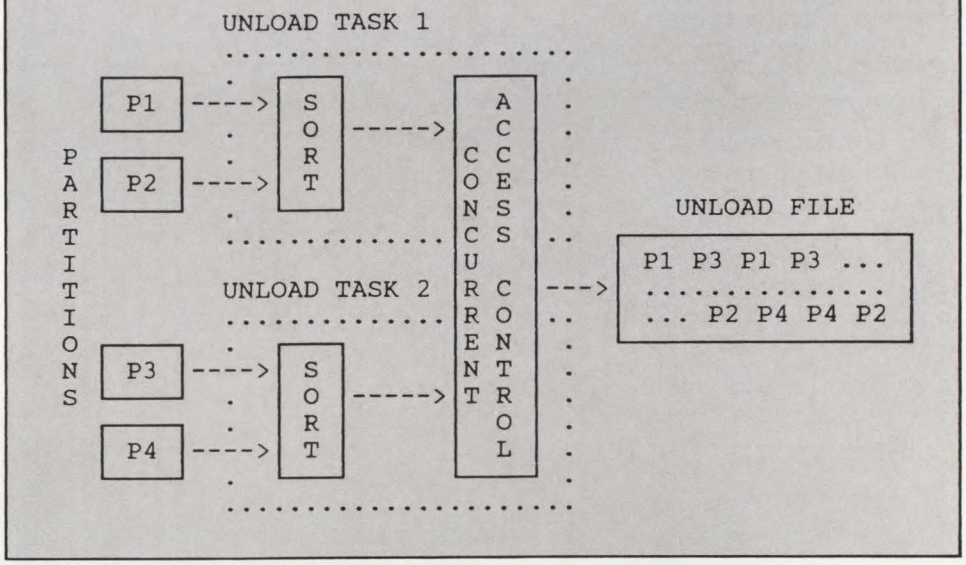
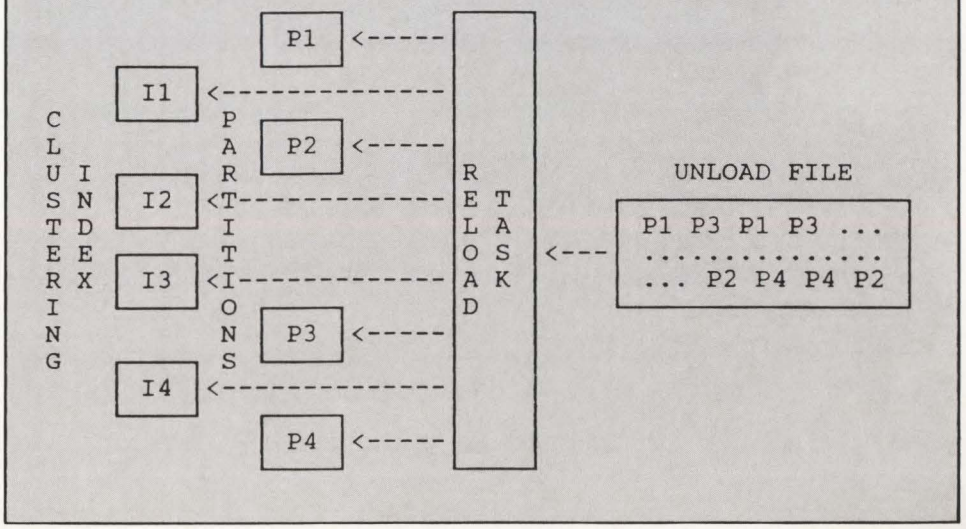


FIGURE 2

Reloading Multiple Partitions Concurrently



Succeed

PLATINUM *technology, inc.*
The DB2 Company™

PLATINUM *technology, inc.* is the only full service software company specializing exclusively in DB2. All our effort and energy is focused on delivering the broadest array of quality solutions for the DB2 user. As a result, IBM® has designated PLATINUM a Business Partner through the Authorized Application Specialist program specifically for DB2.

PLATINUM PRODUCTS

PLATINUM offers a complete line of software tools, education, and published products to ensure your success with DB2.

Software Products

The PLATINUM product portfolio consists of a complete family of administration, development, and end user DB2 software products. All are compatible with DB2 V1.3 & V2.1. The software products include:

- **RC/Query™** - A comprehensive DB2 catalog query tool.
- **RC/Update™** - The industry's best DB2 object management and data editing tool.
- **RC/Migrator™** - A complete object and data migration tool.
- **RC/Secure™** - An extensive DB2 security management tool.
- **PLATINUM Database Analyzer™** - The DB2 database and DASD analysis, audit, and management tool.
- **PLATINUM Report Facility™** - The DB2 query and reporting system for developers and end users operating in TSO and CICS environments.

DB2 Education Courses

A complete series of hands-on DB2 training courses. PLATINUM courses cover all aspects of DB2, QMF, and CSP. All courses are available either at your facility or at our Corporate Education Center.

- Introduction to DB2
- DB2/SQL Application Programming
- DB2 Application Planning and Database Design
- DB2 Database and System Administration
- Using DB2 and QMF
- CSP Application Programming

Published Products

The most recognized and authoritative DB2 standards, methods, and guidelines for DB2 implementation.

- **PLATINUM DB2 Guide/Online™** - The industry's leading standards manual for design, development, and administration of DB2 systems.
- **The PLATINUM Reference™ for DB2** - The quick, pocket-sized reference for DB2 information.

Support

All products and services carry our PLATINUM Quality Assurance Guarantee. Support is available 24 hours a day, 7 days a week via our toll-free hot line.

WORLDWIDE AVAILABILITY

PLATINUM's products and services are available around the world through our Affiliate Network. PLATINUM's full service capabilities include local support, education, and superior DB2 professional services.

with DB2!

PLATINUM *technology, inc.*
555 WatersEdge Drive
Lombard, IL 60148-9930
(312) 620-5000 FAX (312) 953-1923

For further information, in-house demonstration, or our exclusive no-obligation product evaluation call:

1-800-442-6861

Business Partner

CIRCLE #7 on Reader Service Card ▲

One task can unload several partitions at once. Using asynchronous I/O and maintaining reads for each partition reduces the number of waits issued for I/O completion. By dividing the partitions among multiple tasks, work is performed even if a task must wait for an I/O operation to finish (see Figure 1).

Unloading the rows sequentially requires sorting to place them in cluster key order. By sorting the rows during the UNLOAD phase, each task sorts only the portion assigned to it for unloading. This approach provides two benefits: it reduces the number of rows into each sort and it allows several sorts to be active concurrently.

If the tablespace is partitioned, the output of each sort is written to a single file. If the tablespace is not partitioned, a big improvement is still gained through asynchronous I/O. Also, sequentially unloading the data and then sorting it is faster than following the cluster index. Combining the data from each partition allows reloading of multiple partitions in parallel.

Placing most of the burden on the UNLOAD phase provides another benefit. Assigning the new RIDs during unload allows creation of work file(s) containing the secondary index data. Now it is possible to build the secondary indexes at the same time the data is reloaded. DB2 creates the work file during the RELOAD phase. This approach must wait until after all the data is reloaded to build the secondary indexes.

Assigning the new RIDs during unload also allows determination of the exact number of pages required for each partition of the reload. When running with the UNLOAD PAUSE option, the partitions can be reallocated accurately and the reorganization started.

The RELOAD phase also benefits from parallel processing and multitasking. The DB2 REORG utility uses three separate phases to rebuild the tablespace. These phases, RELOAD, SORT and BUILD, run serially. Combining these three phases into one will significantly reduce the time it takes to reorganize a tablespace.

The mixing of records in the UNLOAD file sets up reloading of partitions in parallel. Assigning the new RIDs during unload permits building of the clustering indexes at the same time the data is reloaded. Using asynchronous I/O, the tablespace RELOAD process becomes an I/O manager driven by the UNLOAD process (see Figure 2).

Getting Organized with DB2: Storing and Accessing Data

In order to understand how DB2 stores and accesses data, you first need a short explanation of some important DB2 terms.

Database is the logical representation of one or more DB2 objects. It names a collection of tables and indexes along with the tablespaces and index spaces in which they reside.

Tablespace names a physical area of DASD storage containing one or more tables. It consists of one to 64 VSAM datasets containing up to 64 gigabytes of data. A tablespace contains one or more tables and is the unit of reorganization or recovery within DB2.

Table is a collection of rows all having the same columns. DB2 stores all data in tables.

Row is an ordered set of columns containing a single record of data. One or more rows comprise the data in a table.

Column is a single field of data stored in a row. One or more columns comprise the data in a row.

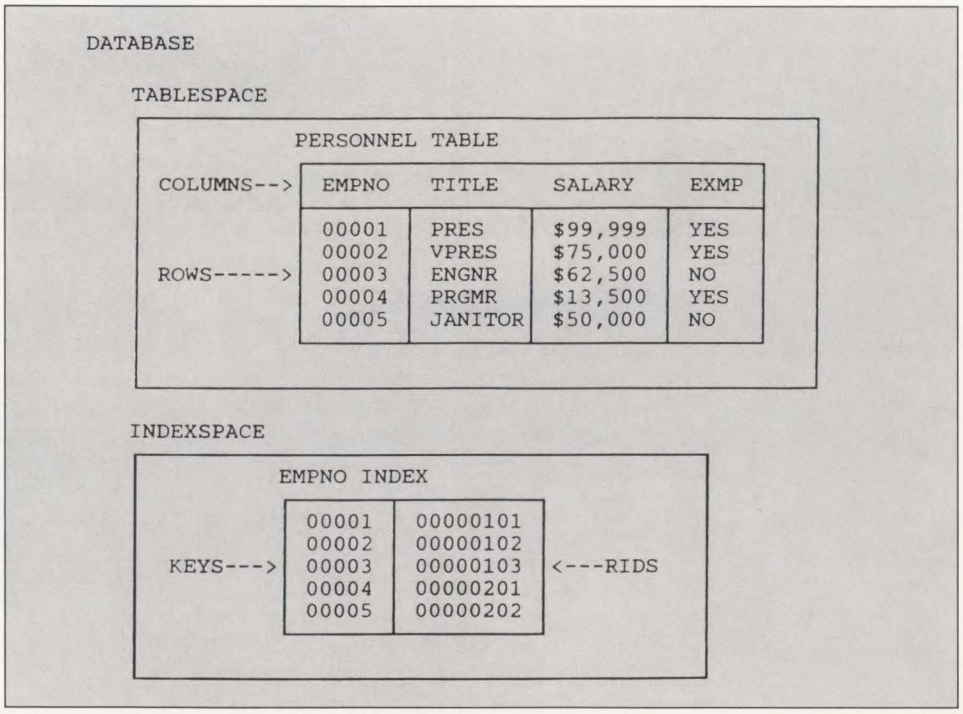
There are two ways DB2 can locate a row in a tablespace: by using an index or scanning the entire tablespace. When no index is available, DB2 scans the entire tablespace sequentially until it locates the desired data. If the tablespace contains more than one table, the scan may inspect data from any or all of the tables.

It is possible to use a key to locate data when one or more indexes exist for the table. Each row in a tablespace has an address known as a Record Identification (RID). The RID contains the page number and slot within that page of where the row resides. If the DB2 optimizer chooses an index as its access path, DB2 uses the RID of the desired row to address the page and row directly (see Figure 3).

Four types of update activity that are the major contributors to performance degradation within DB2 are: dropping tables, deleting rows, inserting new rows in a table with an index and increasing the length of data in VARCHAR columns. To illustrate these activities, we will use the PERSONNEL table from Figure 3.

Suppose that, just after loading the table, the programmer dropped it by mistake. To correct the problem, the programmer redefined the table and again loaded the PERSONNEL data. Dropping a table removes the existence of the table and its associated data from the DB2 catalog. It does not free the space occupied by the data. Space occupied by dropped tables remains unusable within the tablespace until reclaimed by reorganization. Sequential scans of the tablespace include the

FIGURE 3



rows from dropped tables. This perusal of useless data causes increased I/O activity and CPU utilization.

Deciding that selling flowers on the street corner is a less stressful occupation, the programmer decides to take early retirement. Prior to leaving, he deletes his information from the PERSONNEL table. DB2 marks the area occupied by the deleted row as free space. Free space is available for storing new information added in the future. Maintaining free space is essential to a DBMS like DB2, but can cause performance problems.

Seeing the problems created by inexperienced programmers, the company hires five capable DB2 programmers. They quickly add themselves to the PERSONNEL table in order to get paid. DB2 inserts the information for the first new programmer in the free space made available when the original programmer deleted his information. The remaining four new programmers are inserted at the end of the table. Now, however, the rows are no longer in the same sequence as the employee number index. Using the index to access the data requires DB2 to read some pages more than once causing increased I/O activity and CPU utilization.

After two months of steady business, one of the new programmers clearly exhibits superior leadership abilities. The company, pleased with his performance, promotes the programmer to manager of DB2 services. Now that the programmer's title is longer, the information does not fit into its original location. In this case, DB2 stores the entire row at a new position large enough to hold the data. Reflecting this change in every possible index is extremely time-consuming. Instead, DB2 marks the old location as a pointer and stores the RID of the new location at the old one. Now the index does not point to the actual data. Rather, it points to the page and slot that contains the RID of the actual data. This indirect reference requires an extra I/O to retrieve the row. Also, the free space returned to the tablespace is less than the original row length. It is less by the amount needed to hold the new RID and indirect reference indicator.

This illustration will give you a basic understanding of how update activity causes tablespaces to become disorganzied (see Figures 4 and 5). Apply the principles illustrated above to a large order entry system and you can see the effect they have on I/O and CPU resources.

Bill Cunningham

FIGURE 4

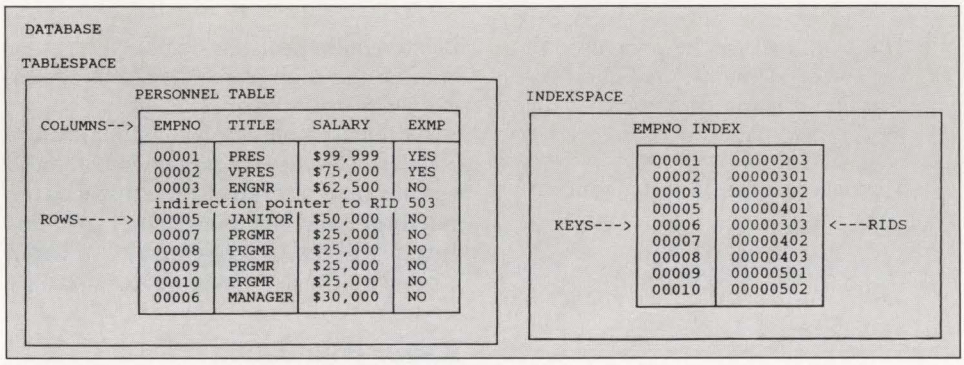


FIGURE 5

Problems Caused By Update Activities

- Dropping tables — causes large areas of unusable data space that must be inspected during scan operations
  — increases I/Os
  — increases CPU utilization
- Deleting rows — causes fragmented free space that may or may not hold future data
  — increases I/Os
  — can increase CPU utilization trying to locate free space
- Inserting new rows with an index — causes the data to be out of sequence with the index
  — increases I/Os
- Increasing size of VARCHAR values — causes data indirection
  — Increases I/Os

Building the secondary indexes using the data in the work file(s), created during the UNLOAD phase, are in clustering sequence. Sorting by secondary index value places the data in proper order. DB2 does this sorting as a separate phase and writes the data to an intermediate dataset. This, in turn, involves more I/O. DB2 then reads this to the intermediate file during the BUILD phase, that involves more I/O.

Using a separate task-per-work file allows simultaneous building of the indexes. Each task sorts the data and builds the indexes without intermediate work files. This approach to tablespace organization provides tremendous reductions in both elapsed and CPU times.

A benchmark used to test a prototype of this design is diagrammed below:

- IBM 3084Q with 96MB of real memory
- MVS/XA 2.2 and DB2 1.3 under VM/SF (two dedicated processors and 48MB of real memory)
- 1,000,000 rows of data, 15 percent out of cluster order
- Average row length: 225 bytes
- Clustering index.

DB2 takes one hour and 47 minutes elapsed time and 10.5 minutes CPU time to reorganize the tablespace. The method outlined in this article takes just under 12 minutes elapsed time and 2.25 minutes CPU time. This is an 89 percent reduction in the time it takes to reorganize the tablespace plus a 79 percent reduction in CPU costs.

Conclusion

The era of DB2 is unfolding. DB2 is rapidly becoming an integral part of production computer applications. It offers application development that is less time consuming and does not require the technical knowledge of earlier database management systems.

DB2 provides the power and the ease of use rquired by today's information systems. This power and ease of operation puts the benefits of DBMS into the hands of many more users, thereby increasing the need for quick solutions to performance and resouce utilization problems.

ABOUT THE AUTHOR

Bill Cunningham is a specialist in DB2 and I/O subsystems and he is a senior developer in the DB2 Products Group of BMC Software, Inc. P.O. Box 2002, Sugar Land, TX 77487-2002, (800) 841-2031.

Capacity Planning:

The Applications Development Viewpoint

By Michael Snyder

Capacity Planning (CP) deals with user demand on a computing complex. In addition, it encompasses technical minutiae such as disk I/O per second, virtual storage occupancy and CPU wait/busy ratios. These ratios are used, for example, to predict the month in which a new mainframe must be brought on-line. Since a month's interest on the $10 million outlay for a mainframe is $80,000, CP can be much more than an intellectual exercise.

Capacity Planning in the Abstract

For the purpose of this article, the discipline of CP is the process of forecasting the workload and data storage requirements of an Electronic Data Processing (EDP) complex for the purpose of meeting service level objectives in the face of a changing level of demand on the resource. A quick review of the terms used in this definition follows.

- It is a forecasting process because the demand level (workload and data storage) changes over time. Absent this, you are a lucky person because you do not need CP.
- The workload can be composed of a variety of elements, including business transaction processing, time-sharing and batch work.
- The data storage of prime concern is disk.
- Service-level objectives can include on-line response time and application availability to users.

If maintenance of consistent on-line response times is an objective, then CP must address itself to workload peaks rather than averages. This has the happy effect of focusing the planner's objective and discussions on a specifically identifiable grail.

When on-line application availability is an objective, CP must deal with one or both of the following issues: scheduled outages of the application and the "other kind" of outages of the application. In the former case, consider a CICS/VS region that has to be shut down periodically for batch processing of its databases. Demands for longer CICS uptime can cause this so-called "batch window" to shrink to the point that more hardware must be planned to finish the batch work on time. Batch requirements thus enter CP's scope of effort due to the force they exert against the CICS operational schedule.

As for the "other kind" of application outages, CP may be employed if these outages pose a survival-level threat to the enterprise. Having little experience in this area, I will take the opportunity to avoid a subject that just makes people unhappy anyway.

Kaiser Foundation Health Plan, Inc.

The company is the northern California component of a nationwide membership-based prepaid health care program. In this region, Kaiser owns and operates 26 medical centers and clinics that are connected to an IBM-based central data center. The medical facilities being of differing sizes, each contributes a different share to the computer workload. Using MVS/XA, 10 "production" CICS regions are run on a 3090-300E and 3090-400. Other workload is primarily composed of CICS testing, TSO and batch on a 3081KX and a 3090-300E with a high percentage of the disks shared among the four machines. Most of the operational data is managed

by IMS/DB and DB2 on the two production CICS machines.

Three applications dominate the CICS workload:

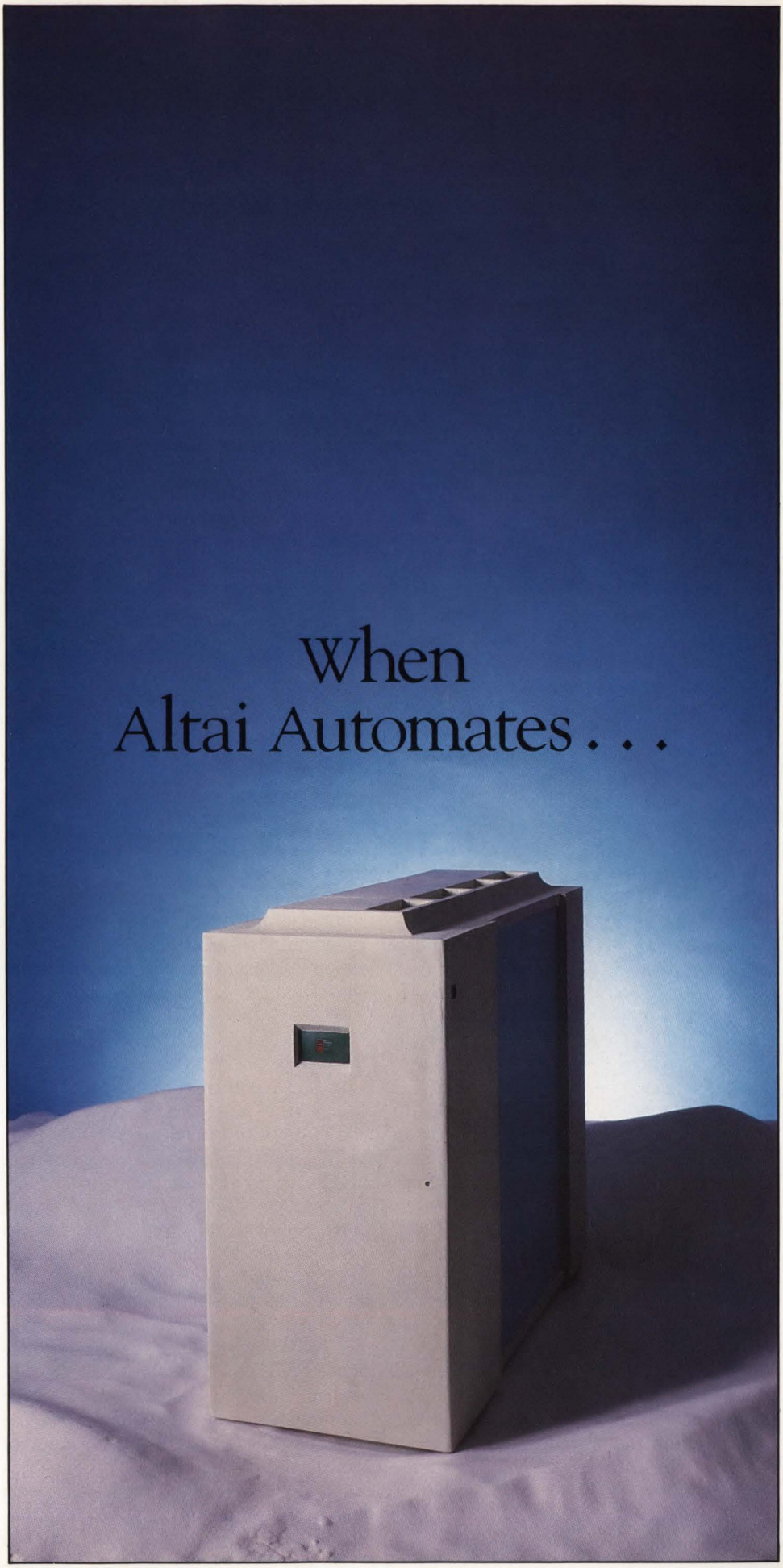
- Patient Appointment, Registration and Reporting System (PARRS)
- Medical Records Management System (MRMS)
- Admission, Discharge and Transfer System (ADT).

Each of these three applications is only partly installed in the sense that not all of the medical facilities are using them yet. As hardware is installed and users are trained, the workload increases in a stepwise fashion. On a peak day, around two million CICS transactions are sustained across the two production machines with short-term peaks of 70 transactions per second. There are other large applications on the way to implementation.

PARRS is used by about 1,000 apointment clerks dealing with patients on the telephone. It is primarily a day-shift application and the biggest of the three applications in terms of computer consumption. It has a predictable and significant Monday morning peak. There is annual seasonality as well. During Monday mornings in the winter, the action gets fierce with 30 appointments booked every minute during the peak two-hour period. PARRS will book around 11 million appointments per year when it is installed in every ''appointed'' patient service department of all 26 medical facilities.

MRMS deals with patient medical records in paper jackets. It is used to keep record room inventories and to track signing-in and -out records among record rooms and medical personnel who use them. It has three separately installable components: inpatient record jackets, outpatient record jackets and X-ray photo jackets. MRMS is a day-shift application and, due to outpatient records representing most of the action, experiences annual seasonality consonant with PARRS. Its weekday peak is less distinct than that of PARRS.

ADT is used to record data concerned with the admission of an inpatient, the various care units to which he might be transferred within the hospital and discharge from the hospital. It requires a high scheduled uptime with 24-hour operation as the ideal. There is little annual seasonality; however, different medical facilities have different weekday peaks. Running at all 14 of the medical facilities that admit patients, ADT serves 200,000 hos-



pitalizations per year, including one-day surgeries.

Capacity Planning Incarnate

I represent the application development group in CP. Four other planners are drawn from the MVS and CICS systems programming groups and the data administration group. For each of us, CP represents a small fraction of our work time (for which we are grateful).

There are several reasons why we are grateful. The forecasting process is difficult especially in light of seasonal and random variations in patient activity. Then, due to the fact that the forecast data and application implementation schedules are by definition rather "soft," the CP process is subject to second-guessing by whoever wishes to challenge our work. Nevertheless, it is fascinating.

For application developers, capacity planning is primarily a twofold concept: CICS transaction workload and disk occupancy. We do not need to plan for CPU seconds or virtual storage or I/O rates. That is not our part of the job. Our job is to express capacity consumption in terms of Natural Business Units (NBUs) that can be related mathematically to the nuts and bolts of computer system capacity.

An NBU is an application-specific unit of consumption that relates to the real-world activity that caused the consumption. If the quantity of a given NBU doubles, the computer resource consumption should double. If the NBU quantity goes down by 10 percent, the computer resource consumption should do the same and so forth. There are several types of NBU by which computer workload and disk occupancy are characterized.

A Workload NBU (WNBU) is used to express the CICS workload caused by an application. My CP colleagues can relate these to actual CICS resource consumption via the CICS Monitoring Facility (CMF) and Resource Measurement Facility (RMF) statistics.

A Disk NBU (DNBU) is used to express disk occupancy. The number of DNBUs is determined partly by the workload (and resultant data growth) and partly by our data retention policy (that is, how long a historic record remains on file). The embodiment of a DNBU is, in our database scheme, a record stored either in DB2 or the IMS/DB database management system. It is preferable to declare a small number of "major" DNBU types for each Application Function (AF). The application developers can use rules of thumb and the entity-relationship diagram to determine what other records will exist due to the existence of the major records. Knowing the bytes consumed by the various major and tributary records, they can relate each different major DNBU instance to actual disk bytes consumed. We can account for non-database disk files in the same general fashion.

CP Example

To make sense of this, the Medical Records Management System will be used as an example. There are four reasons why it represents a complex case of capacity planning.

The first reason is that MRMS is composed of three subsystems: inpatient medical records, outpatient medical records and X-ray records. They do not all need to be installed as a unit. Therefore we must do capacity planning for the resource consumption of each of the three subsystems separately. Each subsystem is called an AF. An AF is a complete application or a major subsystem of one if the subsystems are separately installable.

Capacity planning is a twofold concept: CICS transaction workload and disk occupancy.

Second, the workload of the three MRMS AFs is characterized by three different types of WNBU: inpatient admission (the inpatient record WNBU), outpatient visit (the outpatient record WNBU) and X-ray visit (the X-ray WNBU). It is unimportant that MRMS does not admit patients or register outpatient visits. What is important is that these WNBUs should correlate reasonably well to the MRMS workload and that the company accumulates, forecasts and internally publishes statistics on these metrics. Each of these WNBU metrics corresponds to an MRMS AF. To add to the complexity, CICS transactions in each of the three MRMS AFs present differing amounts of resource consumption in the EDP complex.

Third, MRMS has six basic types of DNBU: the existing inventory of inpatient records, outpatient records, X-ray folders plus incremental inpatient records, outpatient records and X-ray folders created at the medical facilities by the ongoing workload. The MRMS data store is complex, using both IMS databases and non-IMS VSAM and QSAM disk files.

Fourth, MRMS workload and disk occupancy will change over time because it is not yet installed at all medical facilities. We need to express the NBUs done by existing MRMS AFs, those to be done by future installations of the AFs and incremental ones due to business growth.

Application Functions

In MRMS each AF has it own WNBU (admission, outpatient visit and X-ray visit). Each WNBU uses a different amount of the CICS resource and it is necessary to relate these WNBUs to the system software monitoring statistics. The CICS statistics from CMF are tallied for each of the various four-character CICS transaction IDs (Transids) rather than for each application program. Therefore in order to do capacity planning, there should be a standard that each AF should tend to have a set of Transids that apply to that AF only. (Transids consisting of a menu transaction from which the user may select different AFs are an obvious exception to this standard.) If all three MRMS AFs ran under the same set of Transids, there would be no way to relate the CMF statistics to each of the three AFs that comprise the application.

The foregoing sounds complicated until you consider the underlying simplifications that are presented in the sequel to this article in the next issue. We have, as a fast-growing shop, about as simplified a version of capacity planning as we can get away with and still have *reasonable confidence in the results*.

- Start by counting the number of DNBUs of the existing records inventory for each record room (inpatient, outpatient and X-ray) for each MRMS AF currently installed at each medical facility individually.
- Then, count the number of WNBUs pertaining to each MRMS AF currently installed at each medical facility individually: admissions, outpatient visits and X-ray visits.
- Based on the CICS transaction IDs used by each AF, support groups can relate CICS consumption (processor service units, memory occupancy and I/Os) to the MRMS workload caused by each business event: admission, outpatient visit and X-ray visit.

Continuing with the MRMS example, the exercise goes like this.

- Based on the database design, relate the existing records inventory and workload-generated DNBUs to disk bytes.
- The result is how much capacity is consumed by *existing* MRMS installations.
- The next step is to consider the number of DNBUs from existing records inventory and workload-generated DNBUs and WNBUs for each MRMS AF for each planned future installation. The various record rooms and the corporate statistics office supply data by which these estimates can be made. Each AF at each facility may have a different installation date, so that is the way the forecast has to be done.
- Finally, based on business forecasts, estimate the extra WNBUs and DNBUs of the existing MRMS installations due to business growth.

Now three areas are accounted for: the existing MRMS installations, future installations and business growth. As the quantity of NBUs changes, computer resource consumption should change proportionately. Simple.

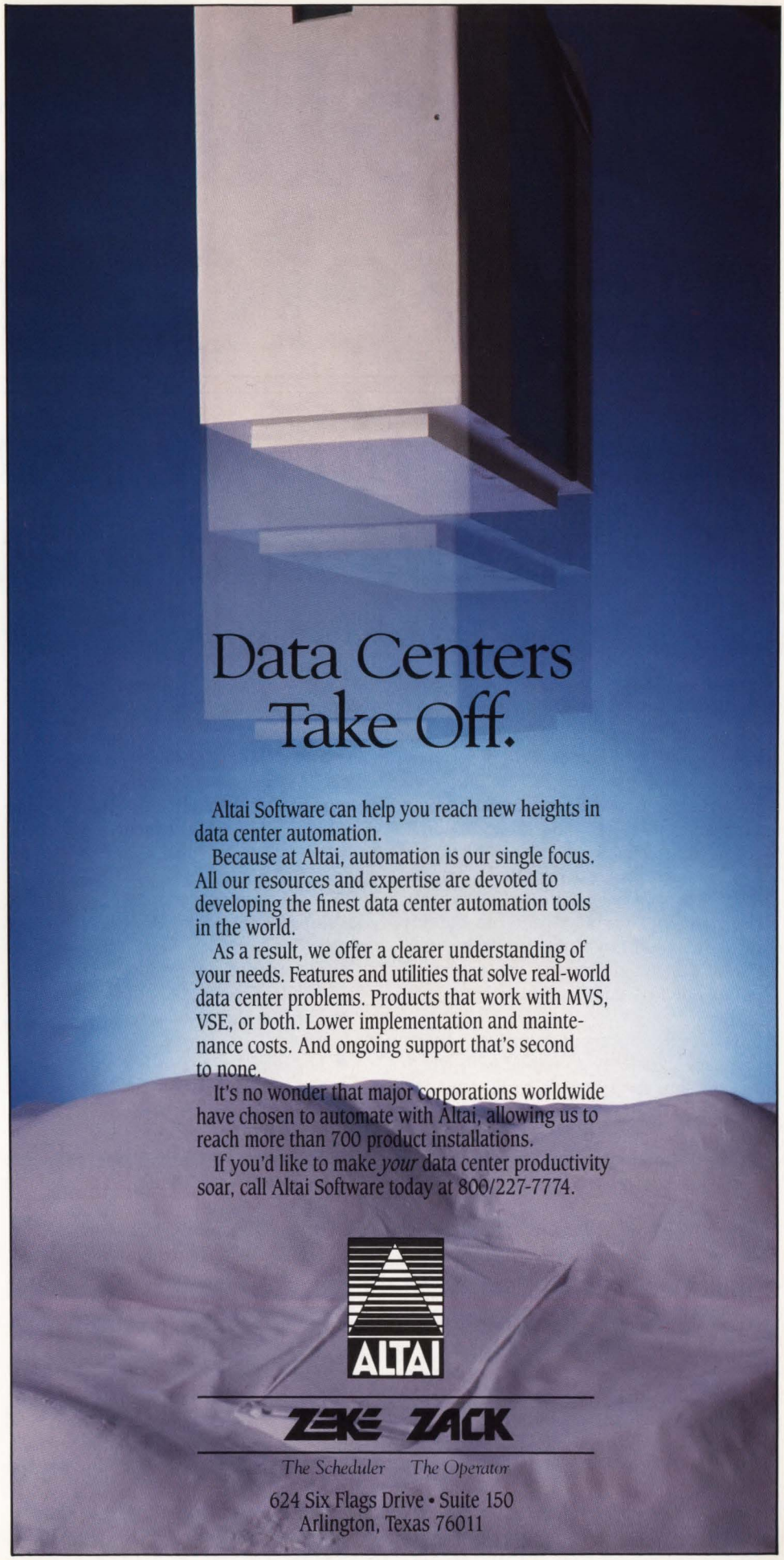
There is one more concept that needs to be considered in capacity planning: the granularity of the implementation schedule. This is a complicating factor best exemplified by PARRS.

When we install PARRS, it is installed in a given medical facility *department* (that is, an allergy clinic), not in the entire facility all at once. (This is largely governed by the speed with which terminals are installed and users are trained.) Each department sustains a different number of NBUs in the course of its work. Hence PARRS could be described as a granular installation. For PARRS, our NBU accounting must be extended down to the medical department level. (For MRMS, the granules for the inpatient record, outpatient record and X-ray AFs are the facility as a whole since each facility has only one inpatient, outpatient and X-ray record room.)

There can, of course, be a different installation granule for each AF in an application. Envision an application containing two AFs, one of which is installed a department at a time and one which is installed for the entire facility at once.

To bring all the foregoing together, observe that when NBUs for an application are counted, we account individually for each reporting period, each AF, each

See Capacity Planning *page 95*



CIRCLE #114 on Reader Service Card ▲

Applications Development

A guide through the maze of tools and methods.

By Robert Lorentzen and Paul Tinnirello

In the revolutionary world of applications software development, technology moves forward like a speeding bullet. An assortment of tools, languages and methodologies whizzes by with some products on their way to massive popularity and others en route to extinction.

For the manager of any sophisticated development project, the choice of development tools acts as a bullet as well. The right tool for a project can quickly hit its intended target and produce a successful system; the wrong one can stray, just as quickly, and become a lethal projectile.

System development managers wonder whether to abide by a method with which they are familiar (the standard system development life cycle, for example) or to test a new technique (prototyping, perhaps, or a modified development cycle).

They worry that the tools they choose for a project — the fourth-generation languages, application generators, relational database management systems and so on — will pass into obsolescence before the project ends. They wonder, too, whether executive management will hold them liable if this worst of ends occurs.

Such concerns suggest that organizations need to adopt procedures for weighing each new development tool's promise against each project's realities and for predicting obsolescence, even accepting it when necessary.

The system development manager should not attempt to shoulder the entire burden himself. Rather, his organization should work out a plan that allows a variety of people to guide the acquisition and use of new applications development tools. Staff members from the DP department should contribute their technical knowledge while executives and end users lend their business savvy to the cause.

A solid corporate plan for applications development includes five major components:

- The identification and classification of business needs and problems
- The formation of a technology task group
- The decision to use traditional or experimental technology
- The selection of specific development techniques and tools
- The implementation of development techniques and tools and the control of their use.

Participants should take care not to burden the plan with bureaucratic overhead. If overcomplication and overmanagement seep in, the plan will hinder the development process rather than help it.

Identifying Business Needs

Careful scrutiny helps an organization determine which of its applications will benefit from new development technology and which will not. As such, this first stage of the corporate plan is a crucial one.

For help in identifying needs, employees who participate in development tool acquisition should turn to their organization's overall business plan. Most organizations rely on one-year or five-year plans to guide their business strategy and most plans list the needs and problems the corporation wants to address. If a software development team attacks these specific needs and problems, it increases its chance of developing successful systems.

Participants should also learn to distinguish between ongoing business needs and dynamic business problems. Typically, an organization can rely on traditional tools and methodologies to meet its stable needs like accounting and payroll. In the same vein, more experimental methods may provide a firm with its only hope of responding to rapid or unexpected change in the business climate.

Clearly, then, if system development managers want to find their way through the maze of applications development tools, they cannot remain ignorant of the business functions their applications are meant to serve. If the data processing department hopes to make wise decisions, it cannot keep itself apart from other business groups.

The corporate plan for tools acquisition should help everyone come together because it eases the strains that often mark relations between DP and other functions. Everyone involved with a given application, whether from the business side or the technical side, helps determine how to proceed.

Forming a Technology Task Force

An organization forms a technology task force to keep steady watch over advances in technology. Managers can assign people to participate either full-time or part-time, but the task force should operate consistently.

Task force members perform two major functions relative to applications software development. First, they investigate and evaluate the development tools that the organization uses day-to-day, noting those that could stand improvement. Second, they keep watch on new technology, looking for tools that might improve targeted weaknesses. When they spot a new tool that seems well suited to the organization's business problems, they alert the development manager and managers of relevant end-user departments.

Because task force members monitor existing operations and upcoming technology, they can act decisively when they uncover a new product that suits the organization's needs. Their preparations allow the organization to implement any new technology in the least possible time.

This ability to act quickly helps an organization survive in today's hectic business environment where companies require shorter and shorter development times. Drawn out development schedules are becoming completely unacceptable because no company can afford to fall behind its competition.

The task force also helps the applications development effort by relieving the development manager of the responsibility for keeping up with every technical advance that comes along.

Deciding Between Traditional and Experimental Technology

In making this decision, the development manager often confronts a dilemma. He knows the traditional system development life cycle will work because it has worked in the past. He realizes, too, that his programmers cannot meet the short development schedules required of them unless they use newer, quicker, untried techniques.

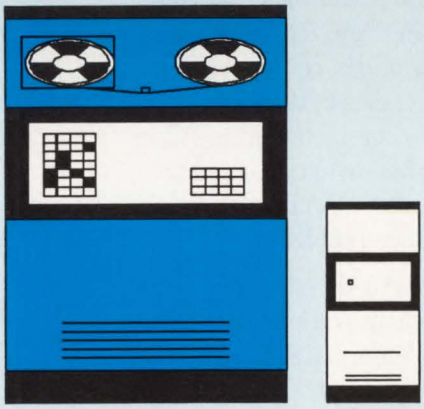
If an organization has no corporate plan for development tool acquisition, the manager finds himself struggling with one problem after another.

- He considers using rapid prototyping methodologies and tools that promise drastic cuts in development times but he finds that these aids sometimes raise end-users' hopes impossibly high.
- He then looks into relational database management systems but the debate on that front only adds to his confusion.
- Next, he tries to find out whether procedural languages, fourth-generation languages or artificial intelligence languages serve best for development but all he discovers is another lively debate.

To compound these problems, the manager must also judge whether the new technology is worth the cost and effort its adoption will entail and whether premature obsolescence will put his final choice to shame. Finally, the manager gives up.

Unable to choose among alternatives, he decides just to wait and see if a sure

Bigger Isn't Always Better!!!



THEIR'S

OUR FAST A480

Introducing the FAST A480 Cartridge Subsystem for IBM and DEC.

Compared to the IBM 3480, the FAST A480 requires half the floor space and approximately 68% less electricity. And, the FAST A480 is fully format and media compatible with the IBM 3480 tape Subsystem, assuring complete data interchange and compliance with ANSI standards. No software additions or modifications are required to your IBM or DEC systems software.

- **Lower Cost**
- **Significant Time Savings**
- **Less Floor Space**
- **Data Transfer Rate Doubled Over Previous Generation Tape Systems**
- **Advanced Technology Assures Data Integrity**
- **Cartridge Tape Technology For:**
  - Added Performance
  - Library Storage Convenience
  - Greater Capacity
  - Higher Reliability
  - Lower Cost Media
- **Optional Cartridge Loader**
  - Allows Automatic Loading of 10 Cartridges
  - Reduces Operator Intervention
  - Increases Thruput

These factors combined wth high density integrated circuits, brushless D.C. Motors and improved packaging all contribute to a compact design with fewer parts and reduced cost of ownership.

For More Information Call ....

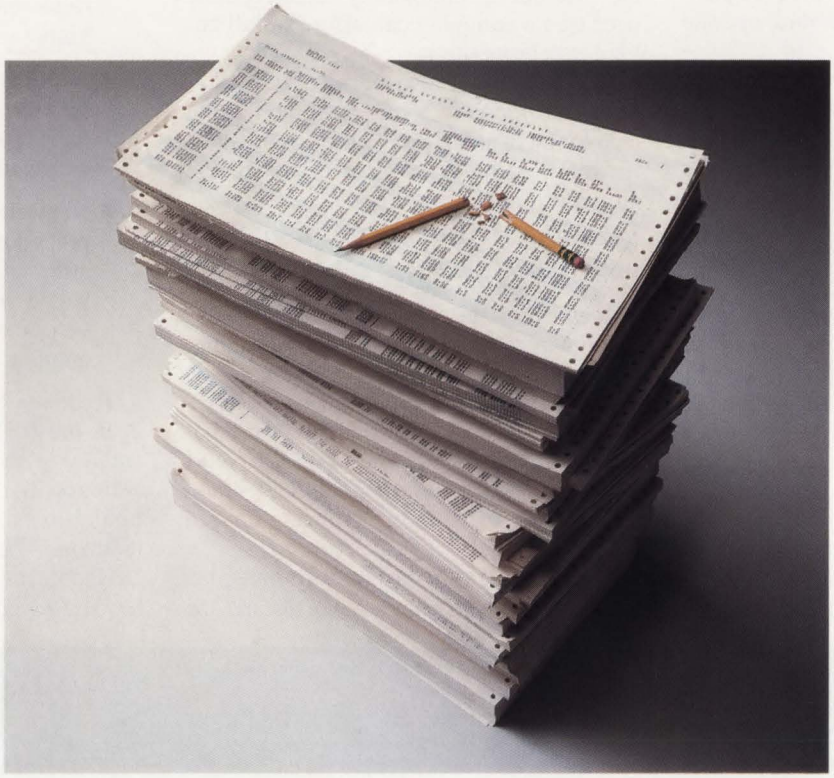
1-800-527-3237 or 1-619-487-8030

fast

First Alliance Software & Technologies, Inc.
First Alliance Plaza, 11770 Bernardo Plaza Court, San Diego, CA 92128

CIRCLE #107 on Reader Service Card ▲

MOST DASD PERFORMANCE SOFTWARE TELLS YOU WHAT'S GOING WRONG.



WE TELL YOU HOW TO MAKE IT RIGHT.

Until now, DASD tuning was a painstaking task-requiring hours of statistical analysis just to figure out what was wrong. Correcting problems often involved guesswork, trial and error and just plain luck. DASD ADVISOR from Boole & Babbage tells you exactly what's going wrong and how to make it right.

DASD ADVISOR is an EXPERT system-based DASD tuning tool that eliminates the need to wade through piles of performance statistics. It analyzes the performance of your entire DASD subsystem, from individual data sets, through hundreds of devices, controllers and channels. It identifies data bottlenecks, then makes specific tuning recommendations. All in concise English. So you have what you need to improve DASD performance. And time to solve other system performance problems.

For a free demo diskette that shows you how DASD ADVISOR can help *you* make it right, call Marty Johnson. In California: 800-624-5566. Outside California: 800-822-6653.

Boole & Babbage, Inc., 510 Oakmead Parkway, Sunnyvale, California 94086.

Boole & Babbage

The Performance People

CIRCLE #165 on Reader Service Card ▲

thing shows up in the near future. More often than not, nothing does.

If, on the other hand, an organization identifies its business needs according to a corporate plan, the manager's task becomes relatively easy — especially if he can rely on his firm's end users and technology task force members for help. The development manager and the other participants should begin by deciding whether the application will serve a long-term, stable business need or a short-term, dynamic business need. The classification will, in turn, help the participants to determine whether the application will benefit most from traditional or experimental technology.

Short-term needs justify the use of low-cost experimental technology — especially personal computer-based development aids — mainly because the risk of obsolescence is lower and the consequences of failure are less dire than they would be in more far-reaching applications.

In long-term applications, the risks and consequences of choosing experimental technology increase and many organizations choose to stay with a tried-and-true approach.

Development managers become especially apprehensive when they hear reports that seem to discredit experimental tools.

However fear of failure or obsolescence is no reason for a firm to shun new development technologies altogether. The larger the project, the more good a speedy, efficient tool can do. Most organizations will benefit from a stepping-stone approach to long-term projects.

Under this approach, a company acquires new tools and techniques one by one as a series of short-term solutions to a specific long-term business problem.

When one tool's usefulness expires, the organization re-examines the business problem and replaces the obsolete tool with a new one that is better suited to current business circumstances.

Companies that are currently switching from COBOL code generators to preformatted screen painters are using the stepping-stone technique to good advantage.

Selecting Specific Tools and Techniques

Once an organization decides to meet one of its business needs with a fourth-generation language, a prototyping methodology or some other type of advanced development tool, it needs to choose a specific product from a specific software vendor.

A formal selection process — one that works toward a well-defined set of objectives — helps tremendously at this stage.

A company should feel free to use any common selection method (a weighted, unweighted, static or dynamic evaluation; a pilot project; or a benchmark) because no one method ensures significantly better results. However participants in the selection process need to recognize five important points.

1. Choosing from a few excellent alternatives is more desirable than choosing among a plethora of poor selections.

The software development marketplace remains so competitive that anxious vendors are releasing some low-quality, untested products.

Participants in the selection process should identify such products early on and eliminate them from consideration.

2. No vendor deserves blind trust, no matter how good that vendor's reputation is.

User organizations must see that each of their prospective vendors defines its products clearly and explicitly.

Many vendors either over-sell their products' capabilities or underestimate their clients' needs although honorable vendors recognize when their product is inappropriate and objectively recommend alternatives.

3. The longevity of a vendor is as important as the viability of that vendor's software tools.

User organizations should stay alert to the need for continuity of support throughout their applications' life cycles. If a vendor is not around to service and maintain its products, the user organization loses out.

4. Complete objectivity does not exist; everyone holds biases and preferences.

End users, managers and consultants have all been influenced by exposure to certain vendors and products and all participants view the selection process from different angles.

If the selection team recognizes its members' biases, the group as a whole can work around them.

5. Many software selection processes fail. This is not because an organization follows a bad method but because the company fails to define its business problem properly or fails to assign trained personnel and sufficient time to the process.

Maintaining the software selection process as an integral part of the corporate plan provides continuity and a foundation for success.

Implementing Development Tools and Controlling Their Use

Because poor implementation can undermine the best of plans, this stage is as critical as the selection process. As with the selection process, success depends on incorporating implementation within the overall corporate plan.

Once an organization decides to acquire a tool, corporate management should inform end-user departments of the tool's intended use and invite end users to participate in the upcoming implementation.

Another key to success lies in understanding the implementation's purpose.

That purpose is to use acquired software tools for solving business problems and meeting business needs — not to bring in new technology so it can proliferate beyond control.

An organization may, for example, address a short-term problem — an unusual reporting requirement, perhaps — with a tool that the firm fully expects to become obsolete within a few years, such as a simple microcomputer database program.

The tool is appropriate to the specific business need but its unrestricted use poses some risk. If users put the tool to other tasks, they may develop an unhealthy dependence.

When the tool becomes obsolete and the organization wants to drop it, these users will stand opposed.

On the other hand, a tool that an organization installs as a solution for a short-term problem may sometimes merit long-term use.

This is especially true if the tool grows or evolves, and the business need continues longer than originally expected.

Some companies that originally purchased Lotus Development Corp.'s 1-2-3 simply for its spreadsheet capabilities have discovered, for example, that the product's macro language supports useful applications development as well.

Keeping Technology in Perspective

Keeping technological advancements in perspective with development goals poses a real challenge to managers of applications development.

When a new product or new technology gets announced, the manager worries that his current development projects may be technically obsolete. When his talented DP employees leave to work at a

firm that implements state-of-the-art technology, the manager's worst fears seem realized. Questions of technical obsolescence become a burden and a nightmare.

These reactions are stronger than the situation merits. True, no manager wants to develop a corporate system that is obsolete before its implementation, but many managers simply do not keep their fear of technological advancements in perspective. Some managers go so far as to develop an obsolescence syndrome, a fear that their skills are no longer current. This fear, in turn, leads a manager to self doubt, which helps no one — not the manager, not his firm.

Systems development managers who find themselves doubting their abilities should take a lesson from their counterparts in software maintenance.

Managers of software maintenance functions often shoulder accusations of not being technically current. But these seasoned professionals shrug off their detractors' comments. Maintenance managers realize that most of today's hot new software will lose its popularity as time passes. They know, too, that their talents will never lie idle; every development tool will require their support sometime down the road.

Applications development managers must recognize one double-edged fact: not all technology is obsolete, yet any technology faces a chance of obsolescence. If a development manager anticipates each application's life expectancy and implements new or old technology according to the relative risks, he can do little more and should try to relax.

Sometimes quantum leaps in technology such as the emergence of microcomputer spreadsheet software warrant a reevaluation of system development goals.

Managers should keep themselves open to the possibility of redirecting a project in those cases. It is not wise, however, to impede system development projects each time development technology advances.

In addition to keeping a personal perspective on technology, the development manager should also help end users and corporate managers balance the promises and risks of technological development.

End users are no longer naive about computer technology. They can see the productivity that is possible with spreadsheet packages and menu-driven microcomputer databases and they may have used fourth-generation languages in a previous job.

Inevitably, they hold opinions about software tools and vendors. Likewise, most upper managers have worked with computer-based systems — some successes and some failures. They therefore harbor strong biases of their own.

These knowledgeable people outside of the DP community are privy to announcements of fourth-generation languages and other tools, that raise their expectations — sometimes impossibly high. Users and managers then pressure the system development manager to act.

When questions of control arise, these users and managers no longer give in to pressures from the DP organization. Rather, they take an active role in and responsibility for the systems. And that is as it should be.

The development manager should caution corporate executives about premature use of new technology but should remain flexible in his convictions. When a new technology purports to provide greater power and control for end users, the system development manager may have to compromise so as not to alienate these important constituents.

Quite clearly, problems like end-user autonomy and premature obsolescence make system development a complex task, despite the common belief to the contrary that the task gets easier as technology advances. Compared with the relatively simple world of 10 or 15 years ago, today's development environment is terribly complex.

Development managers face advanced business requirements and must choose among a variety of development methodologies and a vast set of tools.

The vendors that develop new products deserve commendation for their contributions. However the technical and business communities cannot continue to praise blindly each new development. User organizations and their vendors need to address the practical problems that a company encounters as it applies new tools to the development of business systems.

The impact of new technology is felt throughout the entire business and responsibility for coping with it must spread throughout the company as well with corporate managers, end users and applications development managers all playing a part. Recognizing this is the first step toward strengthening the use of new products and developments. Finding new techniques, such as the corporate plan for new technology, is the next step.

Copyright 1986 by CW Publishing, Inc., Framingham, MA 01701.
Reprinted from COMPUTERWORLD.

A Corporate Plan for the Acquisition of Development Tools and Techniques

Identify and Classify Business Needs

An organization's overall business plan generally lists specific needs. Careful scrutiny of these needs helps determine which ones will benefit from new development tools and techniques and which ones will not.

Form a Technology Task Force

A group of people dedicated to watching for new developments can help an organization act quickly and decisively when a tool that matches a business need comes to market.

Decide Between Traditional and Experimental Technology

For short-term projects, experimental tools post little risk if they are inexpensive and can be implemented quickly. For long-term projects, experimental tools carry the risk of obsolescence, but that risk diminishes if an organization implements them as a series of short-term solutions.

Select Specific Tools

Any well-directed selection process works as well as any other, as long as managers work toward solving business needs and keep the process in line with the overall corporate plan.

Implement Tools and Control Their Use

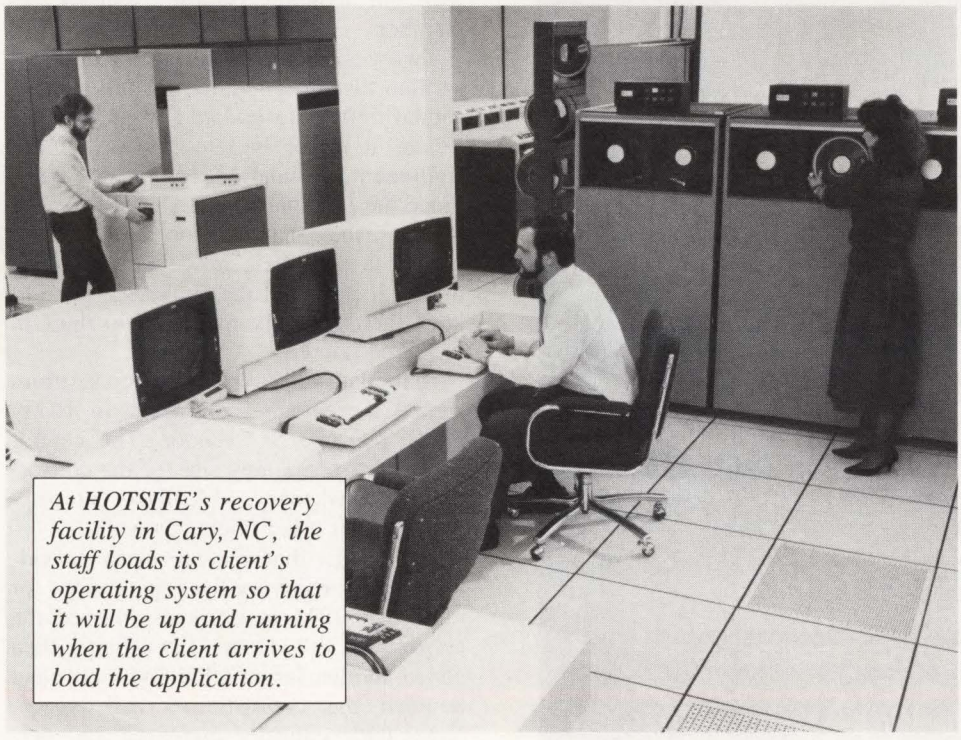
The purpose of implementation is to use acquired software tools for solving business problems and meeting business needs — not to allow new technology to proliferate beyond control.

ABOUT THE AUTHORS

Robert Lorentzen is vice president of Data Services and Paul Tinnirello is assistant vice president of Data Services Programming for A. M. Best Co., an insurance industry information services firm in Oldwick, NJ.

Banking Consortium Solves DISASTER RECOVERY Dilemma

By Beth Ellyn Rosenthal



At HOTSITE's recovery facility in Cary, NC, the staff loads its client's operating system so that it will be up and running when the client arrives to load the application.

While terrorist bomb threats were unlikely, Alan Tarbox worried about floods and fires. As president of the data processing subsidiary of a $1.8 billion bank in Nashua, NH, Tarbox is responsible for the smooth handling of data regardless of the circumstances. Like the rough and tough riders of the Pony Express, the members of Indian Head Data Services had to proof checks and handle exceptions even if someone did a good job of pulling the plug.

Tarbox realized Indian Head Banks, Inc. needed a backup processing site for disaster recovery. "Every financial institution has a responsibility to provide ongoing service to the customer even when the computers are destroyed by fire and flood. We cannot afford to be out of operation for five days," Tarbox explains.

What is at stake is the customer's consummate belief in the integrity and viability of the bank. "Banking is based on the confidence your customers have in you. Much of that confidence stems from your continued service," Tarbox points out.

On the financial side, banks cannot afford to take the day off. They immediately lose interest on the available funds they were not able to process. Also, they can fall out of compliance with federal banking rules, including the new Expedited Funds Availability Act. Finally, stockholders are worried about the security of their investment. Disaster recovery is a key component in that protection.

However, Indian Head, a member of the Fleet/Norstar Financial Group of Providence, RI could not afford the cap-

ital costs of duplicating its own equipment elswhere to create a mirror image processing center to be used only in time of trial.

"The costs are significant," says Tarbox, estimating a backup site would cost the bank about 75 percent of its data processing equipment costs. "First, we had to rent space that is available 24-hours-a-day. Then we needed a new computer system that we could switch on if something went wrong. Add to that the maintenance costs and we just couldn't do this on our own," he explains.

So Indian Head, along with eight other banks in northern New England and Maine, formed a consortium to share the costs of establishing and running a backup site. The consortium, formed in 1985, was christened the Northern New England Backup Services Corporation.

The consortium rented space in the Federal Reserve Bank of Boston's regional check processing center in Lewiston, ME. It purchased an IBM 4341, a system common to all members. Each bank then solicited employees to staff the site on a voluntary basis.

The data processing chieftains breathed a collective sigh of relief when the lights on the backup computers started to glow. Each member had the reassurance the bank could continue processing in the midst of a disaster at a cost the financial institution could afford.

However, the calm turned out to be short-lived.

The Boston boom spilled over into New England where banks began to grow at an exponential rate. Like the piglets in George Orwell's "Animal Farm," some banks grew faster than others. They acquired ravenous appetities and began swallowing the little fish. On the other hand, some members, including Indian Head, were acquired by area giants.

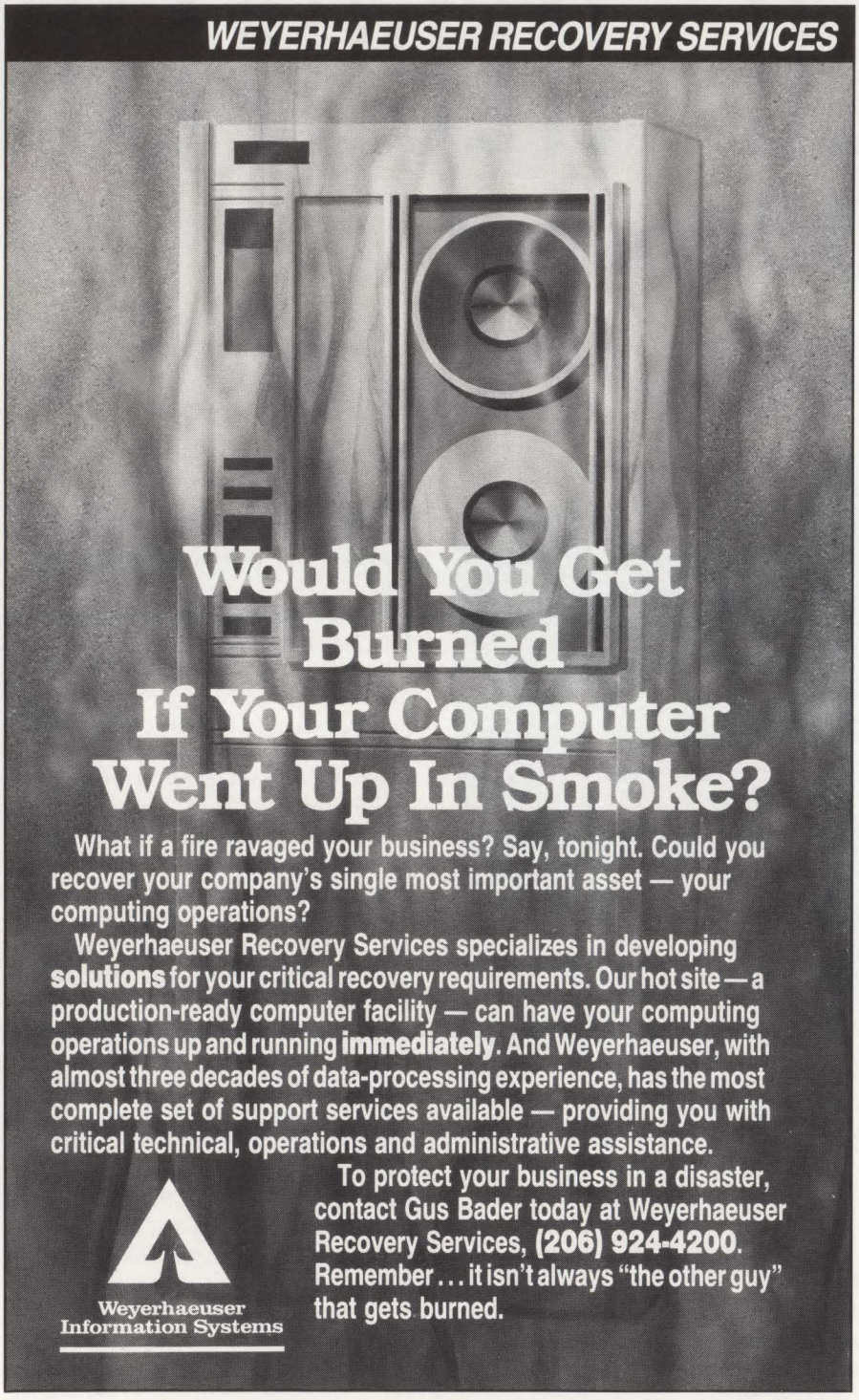
Suddenly each bank had different needs and conflicting requirements. The success of the region was tearing asunder the carefully crafted consortium. "The needs of our members changed so dramatically we found some members could not live with our existing system," Tarbox remembers.

Two years after its inception, the consortium faced a dilemma. Facing surging capital outlays to update and upgrade its center, it either had to purchase more equipment, disband or seek outside help. Since the nine banks had worked so well together, they searched for a vendor who could service their now heterogeneous needs. If possible, the bankers wanted to keep their monthly membership dues in the same range.

In the late fall of 1987, the consortium signed a five-year contract with HOTSITE, a Cary, NC vendor. The vendor agreed to open a new site for the consortium in Tewksbury, MA if the company could solicit other clients as well.

HOTSITE, that specializes in providing disaster recovery backup services for mid-range IBM mainframes, moved the consortium's 4341 to Tewksbury, then added another larger CPU and more than doubled the consortium's disk space. "Our CPU capacity increased 250 percent. HOTSITE greatly expanded our data processing facilities," Tarbox adds.

These amplified capabilities have allowed the member banks to add new services that the understaffed, overworked consortium center could not handle. For example, while all the member banks had item capture and batch processing well



CIRCLE #28 on Reader Service Card ▲

THE IBM MID-SIZE MAINFRAME DISASTER RECOVERY SOLUTION

On-Site 3890 Capability

Cary, North Carolina

Tewksbury, Massachusetts

Niles, Ohio

Relocatable Shell Facilities

HOTSITE® offers:

- Multiple, fully equipped recovery centers
- Relocatable shell facilities
- PC based planning services
- Data communications network backup

For more information about HOTSITE® return the coupon or call our Marketing Services Department.

HOTSITE®
A DIVISION OF COMPUSOURCE

HOTSITE® Marketing Services, Cary, NC (919) 460-1234

YES, please send me more information about your disaster recovery services.

Name _____

Title _____

Company _____ *Phone* _____

Address _____

City _____ *State* _____ *Zip* _____

Mail to HOTSITE®, Marketing Services, 110 MacKenan Drive, Cary, NC 27511

CIRCLE #40 on Reader Service Card ▲

covered, none were backing up the online transactions of their branches. Now HOTSITE is also doing that for them.

Tarbox was also gratified that HOTSITE supplied trained personnel to man the center round the clock. "They have 15 people there at all times monitoring our accounts. Before, Federal Reserve employees were there all night, but it was not their job to service us. We provided our own staffing," Tarbox points out.

The volunteer system was not the optimal staffing approach, points out John Butch, division manager for HOTSITE. "The volunteers made their full-time job their first priority. Finally it became a burden to manage the facility," he comments. A permanent staff gave HOTSITE the manpower and the technical support to perform more extensive disaster testing that the consortium's employees had neither the time nor the training to do.

Finally, HOTSITE also owns a relocatable shell facility that any member of the consortium can use after a natural or manmade disaster. The modular shell snaps together like Lego toys to form a 3,500 square-foot computer room with the requisite raised floor. "In less than two weeks the consortium had a new computer center," Butch points out.

After HOTSITE assembles the shell in the bank's parking lot, communications lines are tied in. The bank buys its new equipment that HOTSITE installs in the portable center. When the permanent facility is rebuilt, HOTSITE moves the computers to their real home.

Butch says relocatable shells are critical when disaster really strikes. Even the swiftest construction foreman requires a minimum of three months to rebuild a bank's computer center. That means for at least 90 days, the data processing staff must relocate to the backup site that can easily be a two-to five-hour drive until the dust settles back home.

"It doesn't take long for an employee to be away from home before becoming disgruntled. Relocatable shells allow the bank to minimize the inconvenience," notes the HOTSITE executive.

So how much does this fortified service cost? "We were able to put together backup services for the consortium with enhanced equipment, on-site technical assistance and full time management for no more dollars out the door," Butch points out.

Tarbox adds that each member continues to pay his current assessment, a price that will remain fixed until 1990. Thereafter, the consortium dissolves, allowing each financial institution to negotiate its own rates with the vendor.

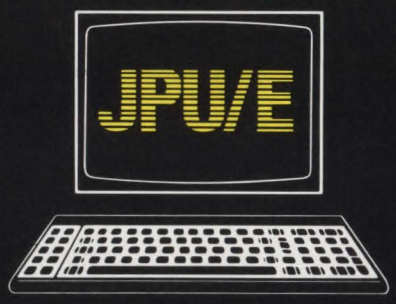
HOTSITE is happy it can help bankers like these sleep well at night. "Computers are pervasive in banking. Many banks open new accounts through a terminal. ATMs are run by computer. The commercial and installment loan departments need access to the data bank. I don't know any bank today that processes a check without a computer. HOTSITE is their best insurance policy," concludes Butch.

ABOUT THE AUTHOR

Beth Ellyn Rosenthal is the founder of the Writers Block, a Texas writers' syndicate. She is the syndicate's banking and technology specialist.

WHEN DATA SEEMS DAMAGED BEYOND REPAIR,

HERE'S HOW TO RECOVER.



THE JOURNAL PROCESSING ENVIRONMENT (JPU/E) from Softsystems is the most sophisticated and flexible data recovery software on the market. With JPU/E, you can:

- Recover VSAM files, BDAM files and DL/I databases.
- Recover both batch and online updates since your last backup.
- Automatically archive (backup) any or all CICS disk journals.
- Review and manage system status through online interface.
- Automatically update "hot" backup files at specified intervals.
- Perform comprehensive journal processing, including merging of journals and extensive printing of an array of statistics and detailed journal record contents.

To find out more about JPU/E, call Softsystems today. Before a damaged data file throws you for a loss.

SOFTSYSTEMS, INC.

300 Mallick Tower • One Summit Avenue • Fort Worth, Texas 76102
800-331-7846 • 817-877-5070 • In Canada: 800-367-8673

Available for VSE, VSE/SP, VS1, MVS and MVS/XA installations.

CIRCLE #10 on Reader Service Card ▲

For CICS/VSAM Data Recovery, Integrity Solutions is the Global Choice.

These leading IBM mainframe companies trust the integrity of their corporate data to Integrity Solutions. They selected Integrity Solutions' DATA RECOVERY SYSTEM (DRS) because it provides the most comprehensive forward and backward recovery for files accessed by CICS/VS and VSAM batch processing.

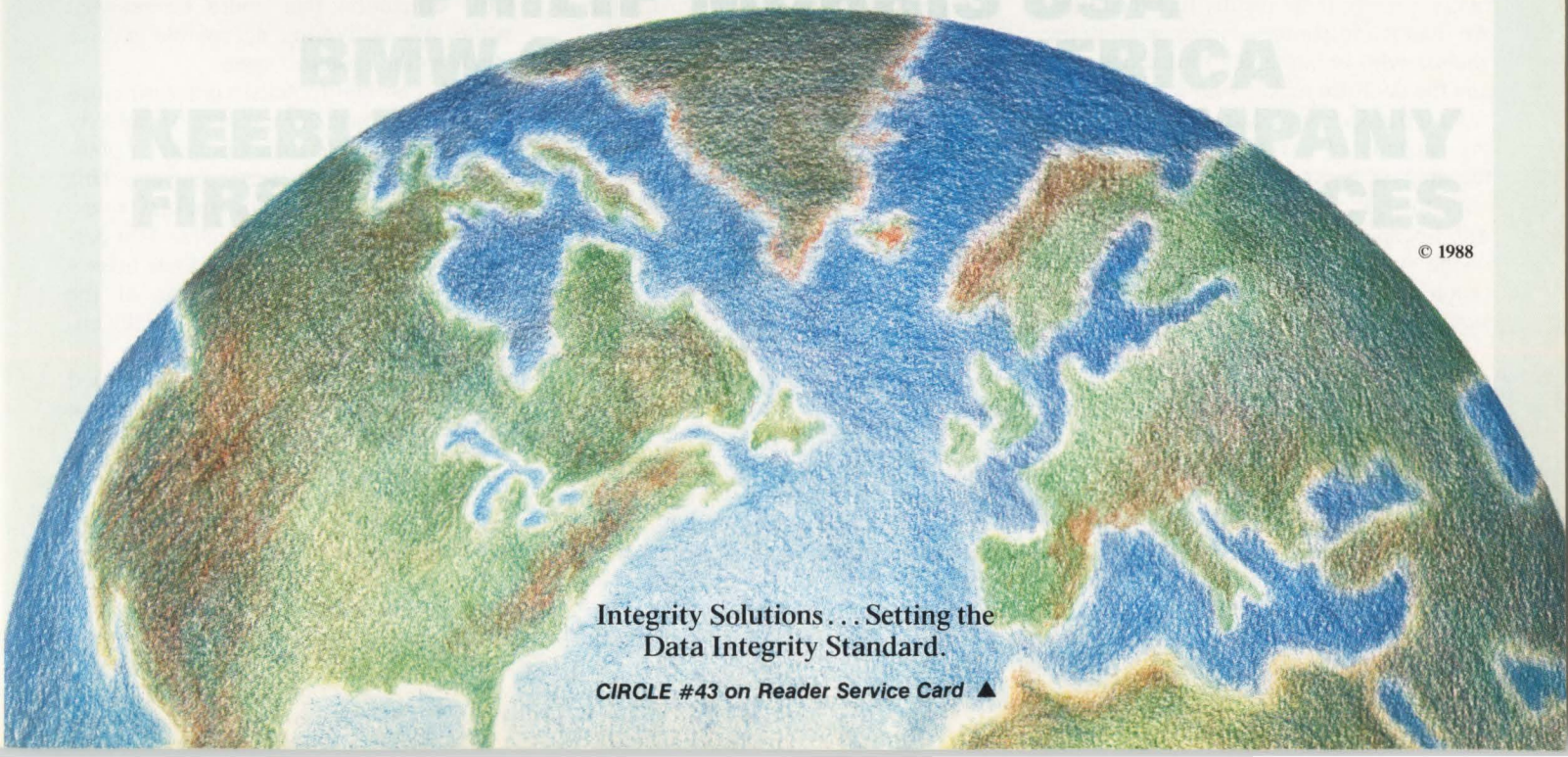
Integrity Solutions introduced DRS more than 8 years ago, and with over 1000 product installations worldwide, Integrity Solutions will continue setting the data integrity standard. By providing leading-edge products and the highest level of support.... to serve your data recovery needs now and in the future.

Guided by what users say they need next, Integrity Solutions is setting the data integrity standard for the emerging electronic vaulting and mirror processing marketplace as well.

The DATA RECOVERY SYSTEM from Integrity Solutions, because your data is worth it.

Integrity
Solutions, Inc.

1-800-289-9900
1-303-794-5505



© 1988

Integrity Solutions... Setting the Data Integrity Standard.

CIRCLE #43 on Reader Service Card ▲

Capacity Surfaces

The capacity surface offers the analyst a vehicle for communicating large quantities of complex modeling results to decision makers.

By H. Pat Artis

Perhaps the most difficult problem faced by capacity planners and performance analysts is not how to measure, predict or improve the performance of their systems but how to effectively convey their results to executives. An interesting paradox is presented to the analyst who is immersed in detailed data and the decision maker who wants a quick summarized view of daily performance or the capacity of a system. On one hand, the analyst is concerned that responding with too little information will result in misconceptions about the characteristics of the system. On the other hand the decision maker typically requests a *one page summary* that will allow him to understand his alternatives and reach valid conclusions.

In an effort to achieve a compromise, many analysts have attempted to provide abbreviated report formats, perhaps just a few pages, to address the decision maker's needs. These analysts are often horrified when the decision makers net out their presentations to a single number that the analyst feels is invalid. *One trait of human nature is that a person who wants to simplify a decision will do so with or without the help of others involved in the process*. For decision makers, this reaction is not an attempt to trivialize a problem but rather an attempt to fit the decision process within the time available in their schedules.

Historically, a variety of numerical and graphic presentations have been employed to convey performance and capacity data to decision makers. For example, indices have been used to rank service level and utilization measurements to relieve the recipients of the reports of the need not to remember what values are good and bad. While such approaches can convey a tremendous quantity of data, the recipients are left to interpret the relationship between the indices themselves. A variety of graphic presentation formats have also been investigated.

Perhaps the most commonly used graphic format is shown in Figure 1. This figure presents a history of CPU utilization and a forecast of expected future values of the processor's utilization. Using a rule of thumb guideline for target CPU utilization, the capacity of the system is defined as the point at which the average CPU utilization during the peak hour exceeds the selected target value. In the example shown in Figure 1, 85 percent was chosen as the target maximum value of CPU utilization that could be realized while still delivering acceptable service levels to the system's users.

It is important to realize presenting utilizations as a primary measure of capacity obfuscates the real issue of capacity planning, in other words service levels. This often leaves the decision makers questioning why they cannot realize 100 percent of the apparent value of their investment. Moreover, the recipients of the report were very likely to simplify the entire decision process by concluding that their system's capacity is the specified target utilization whatever that meant in terms of their workload or service levels.

When analytic queueing models were introduced in the late 1970s, they provided the capacity planner with powerful tools for relating workload arrival rates, utilizations and service levels. Depending on your perspective, the wealth of detailed data that these models provided the

analyst either revolutionized or further complicated the capacity planning process. From the analyst's perspective, the models provided them with a much more detailed understanding of the interactions and service levels of the system's workloads. From the decision maker's perspective, the quantity of data, however useful, that the capacity planner wanted to convey to them markedly increased. Not withstanding these issues, the use of analytic models focused both the analysts and decision makers on service level related measures of capacity for the first time.

Perhaps the most common graphic presentation format used to convey the results of the analytic models is shown in Figure 2. The graph shown in the figure relates the response time of the system's critical workload to the transaction volume with the limit line being defined by the service objective for the application . . . assuming a given background level of other workloads. One unfortunate characteristic of this format is that it invites the recipient to reach a potentially invalid conclusion. For the system portrayed in Figure 2, the decision maker is invited to conclude that a throughput of eight transactions per second can be achieved while still meeting the service objective of one second. This conclusion is likely to be invalid long term since the model assumed a constant value for the background workloads that are also likely to grow in the future.

From the analyst's perspective, such problems illustrate the pitfalls of the *net it out in one page* school of thought. Simply described, there are far too many important variables to allow meaningful results to be summarized onto a single page. Further complicating the entire management process is the fact that the one page daily performance summary often bears little similarity to the reports provided during the capacity planning process.

Recently, during a conversation with a corporate executive, I had used these arguments in an attempt to explain the limitations and pitfalls of the one page summary to the client. Unhappy with the explanation, the client rephrased the question in a manner that provided me with a new insight into the problem. What the client said was, ''What I really want to know is how close I am to the edge and how far I can go before I fall off?''

The question suddenly implied that the system's capacity was better represented by the concept of a surface than by absolute values of arrival rates or utilizations. Since a surface is defined by thousands of points, it provides a single page format that can incorporate the wealth of data identified by the capacity planner while still realizing the decision maker's objective of brevity.

To paraphrase an observation made by Alan Kay, widely regarded as one of the true visionaries of personal computing and an Apple Fellow, *''most corporate executives, although they've never really thought about it, would prefer to have a model of their system's performance with which they can visualize its response to a variety of situations''* rather than rely on records to see how changes in the past have affected the service levels delivered to their users. The capacity surface offers an approach to addressing this requirement. The remainder of this article describes how capacity surfaces are generated and will show how they can be used to address the daily performance reporting, capacity forecasts and analyze the benefits of alternative hardware upgrades.

FIGURE 1

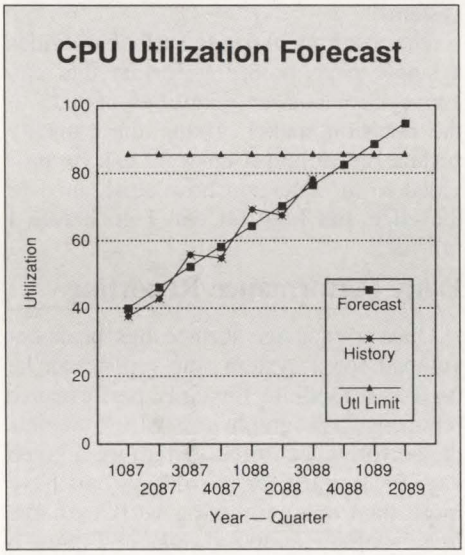


FIGURE 2

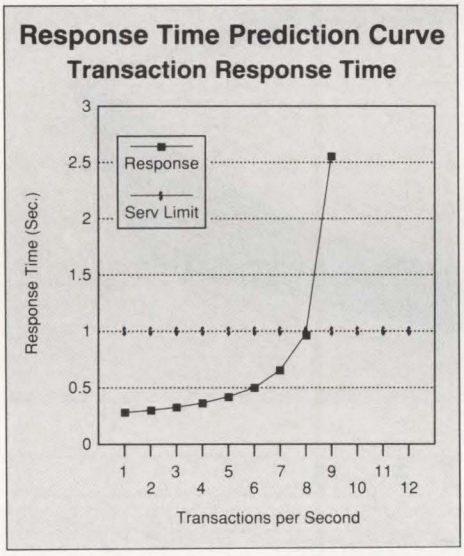
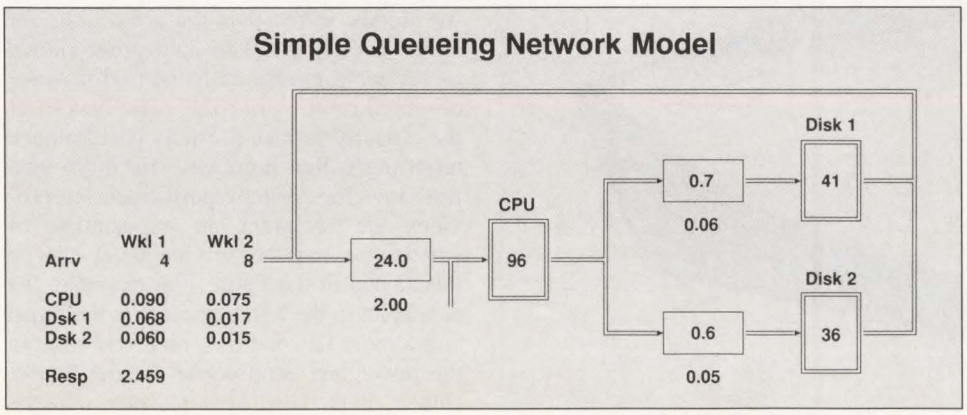


FIGURE 3



Building A Capacity Surface

To illustrate the development of a capacity surface, it is useful to introduce a simple queueing model. Figure 3 depicts a simple two workload model that describes the performance of a system comprised of a CPU and two disk drives. While this model is far simpler than the commercial tools that are used to analyze actual MVS environments, it will serve the purpose of demonstrating how capacity surfaces are developed. By varying the arrival rates of the two workloads, the performance and utilization levels of the system can be predicted for any environment. For example, if a background arrival rate of workload two was assumed to be three transactions per second, the curve previously shown in Figure 2 would be produced when the arrival rate of workload was varied from one to eight transactions per second.

If the model were used to calculate the response time of workload one for all possible combinations of the arrival rates of the two workloads, then the three dimensional plot shown in Figure 4 could be developed using the arrival rates and predicted response times. In effect, the ribbons shown in the figure represent the family of response time curves from which Figure 2 was selected. An alternative representation of the family of curves shown in the figure is the capacity surface shown in Figure 5. The capacity surface conveys the predicted response time of the system over all possible combinations of arrival rate values. The broad sloping surface that begins in the lower left hand side of the figure represents the range over which the system can deliver acceptable service levels. The ledge at the upper right hand side of the slope represents arrival rate combinations. Although they do not saturate the system, they result in response times in excess of the one second target for system. Finally, the void in the upper right hand side of the figure represents arrival rate combinations that saturate the system.

FIGURE 4

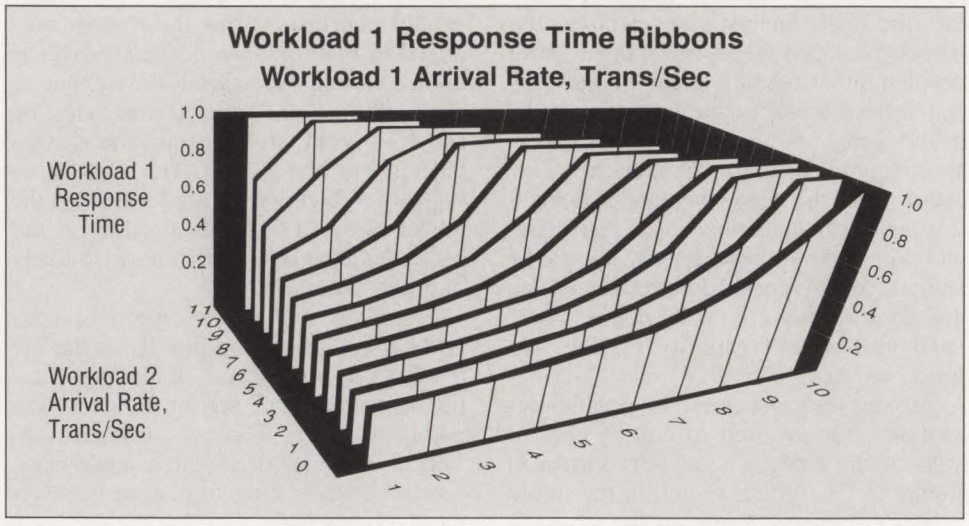
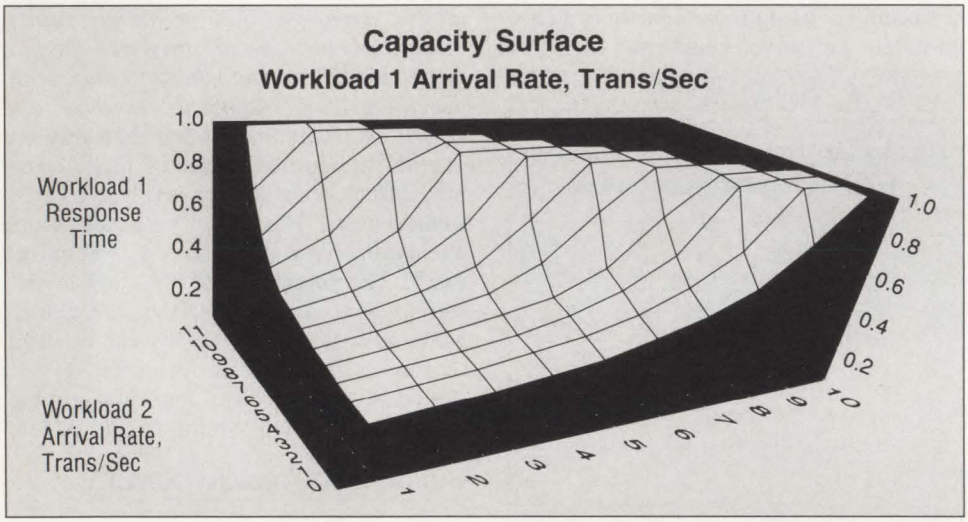


FIGURE 5



The resulting capacity surface provides a single page reporting format that can convey tremendous quantities of data to the decision maker. Using the capacity surface, meaningful answers can be provided to the question how close am I to the edge and how far can I go before I fall off?

Daily Performance Reporting

Once a capacity surface has been developed for a system, the report can be used as a medium for daily performance reporting. The graph in Figure 6 depicts the performance of the system for a given day. Since most real world systems have more than one competing workload, the Y-axis of the graph shown in Figure 6 expresses the percentage of CPU utilization represented by all of the competing workloads, rather than the arrival rate for any one of them. The data point shown on the surface represents the performance observed on the prior day. The benefit of the capacity surface for daily performance reporting is that it conveys far more than just how the system performed; it provides the recipient an expectation of whether or not the service level can be maintained in the future. For example, the data point in the figure appears in the broad and almost flat plane of response time in the lower left hand corner of the figure. Hence, it is reasonable to infer that the

FIGURE 6

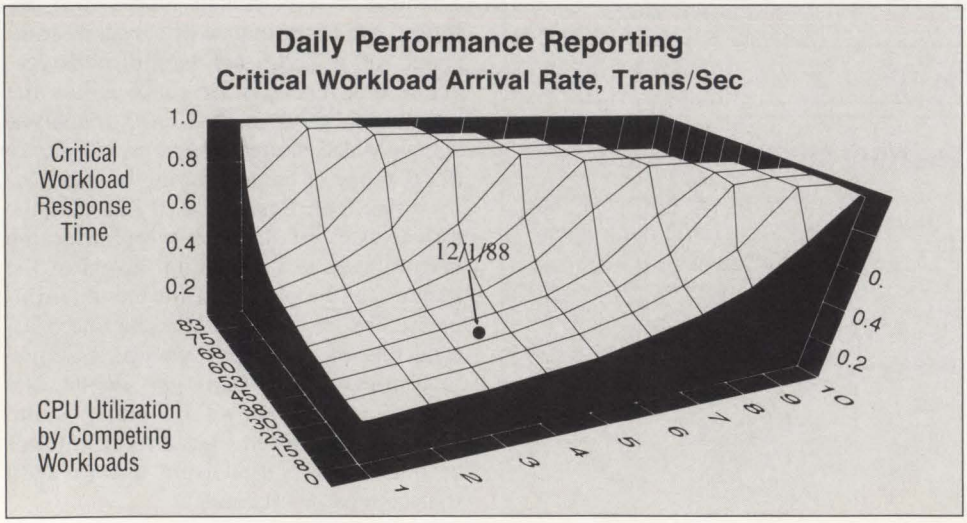
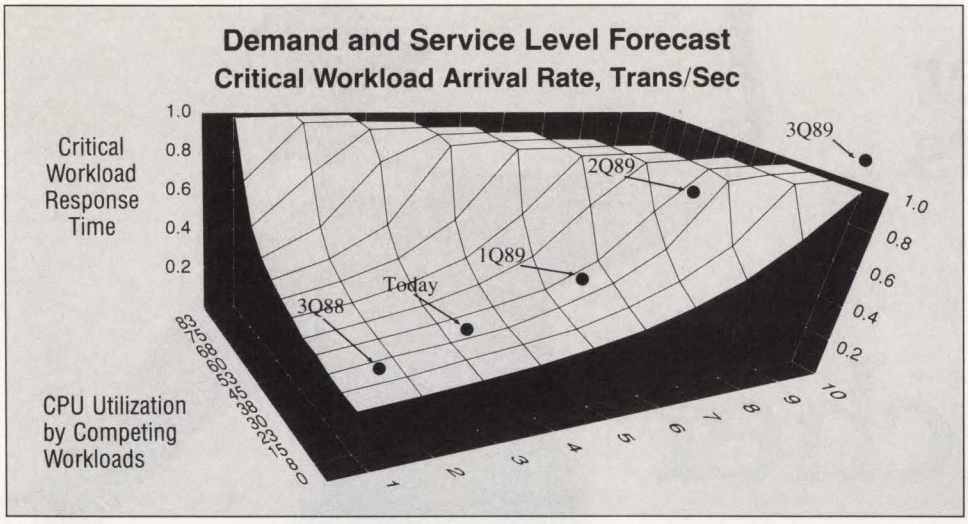


FIGURE 7



system's performance would be maintained even if substantial changes occurred in the users' demand on the system. However, if the data point appeared near the ledge on the steep part of the surface, then it would be reasonable to infer that future performance levels might be erratic or unacceptable.

Capacity Forecasts

Like performance prediction, workload forecasting involves a tremendous quantity of detailed data. While a single workload forecast may suit the needs of a simple environment, typical workload forecasts are the result of independently analyzing the behavior of perhaps a dozen primary applications. Once this data has been collected and analyzed, the capacity surface provides a medium for communicating it to the decision maker.

Figure 7 provides a workload and service level forecast using the capacity surface as a reporting medium. The data points connected by a solid line represent the historical behavior of the system and the projected load and service levels for the next three quarters are denoted by discrete points. As can be seen in the figure, response time will degrade over the next two quarters. Moreover, the demand forecast for the third quarter does not represent a point on the capacity surface or even a point on the ledge where the system could serve the expected workload at a degraded level of service. Hence, the third point conveys that the system will be completely saturated in the third quarter of 1989 and that the current configuration is incapable of supporting the anticipated load at any level of service. Once the exhaust point of the system has been identified, the next natural question is what kind of upgrade will be required to serve the anticipated load?

Evaluating Upgrade Alternatives

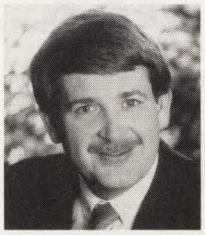
Another potential application of the capacity surface is the evaluation of upgrade alternatives for a system. For example, what would be the impact on the system of replacing the system's CPU with one that is one third faster? To answer this question, the transaction CPU times previously shown in Figure 3 would be reduced by one third and then the response time values used to develop the capacity surface would be recalculated. Figure 8 shows the capacity surface that would result if the CPU were upgraded. As can be seen in the figure, upgrading the CPU substantially expands the range of arrival rates over which the system can meet its response time objectives.

In the event that multiple upgrade alternatives were being considered, a capacity surface could be developed to facilitate comparisons of the alternative configurations. The principal benefit of the capacity surface in this process is that it focuses decision makers on what a potential configuration can do for their systems rather than abstract comparisons of MIPS or device characteristics that may or may not be exploitable by the system's workload.

Remarks

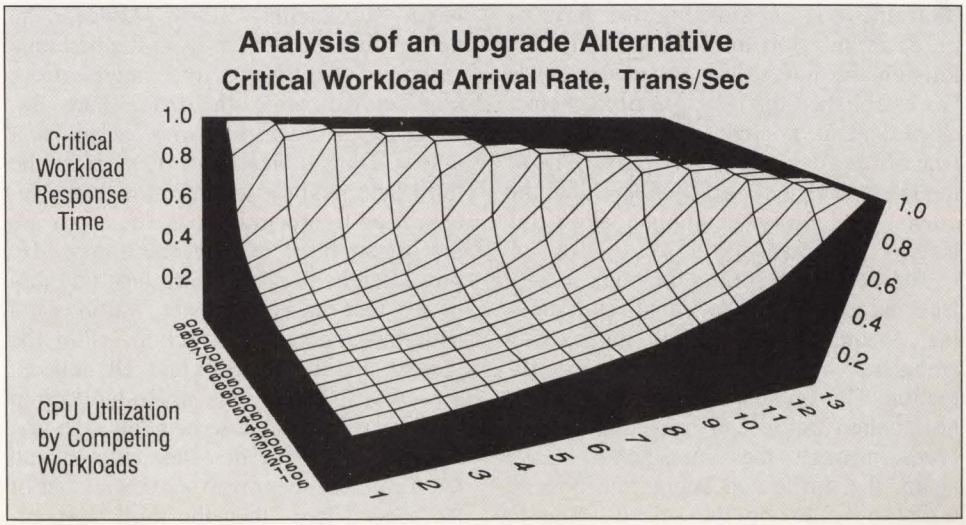
While it is impossible to envision that any single page reporting format will ever meet all of the requirements of the analyst, the capacity surface offers the analyst a vehicle for communicating large quantities of complex modeling results to decision makers. Moreover, it provides a common report format for addressing daily performance reporting, capacity forecasts and the evaluation of alternatives in a complex environment.

ABOUT THE AUTHOR



H. Pat Artis is a recognized authority in workload characterization, forecasting simulation modeling and capacity planning. He is the author of more than 75 papers and lectures internationally. Artis is president of a private consulting firm, Performance Associates, Inc., Palm Desert, CA 92260, (619) 346-0310.

FIGURE 8



VSE Under The Covers

The Dispatcher And Scheduling System

By Eric L. Vaughan

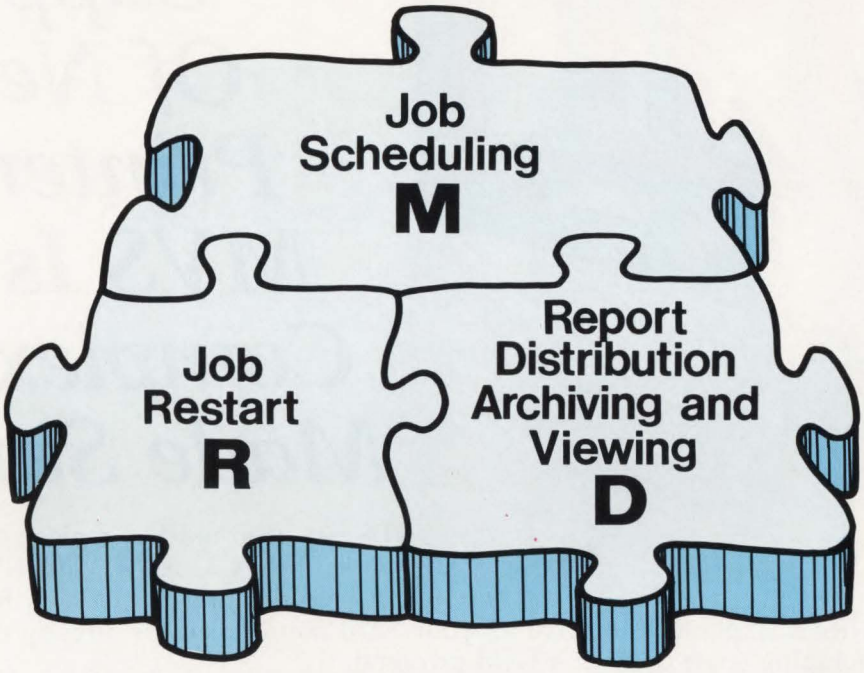
It is five p.m. on Friday in the middle of a busy four-way intersection. A traffic cop is directing the flow of traffic in an effort to move as many cars through the intersection as possible. He has established his own priority scheme, albeit random, in order to determine which lane of cars to begin moving first. One of the lanes tends to build up faster than the others so he always attempts to allow this lane to move first.

The traffic cop selects the high priority lane and motions for the cars to start moving. As the sixth car passes, he hears an ambulance coming from the opposite direction. The ambulance creeps around the now halted traffic and immediately proceeds through the intersection. Once again, the traffic cop begins the flow of traffic by allowing the priority lane to move. When this lane is empty, he turns to the lane next in priority in order to begin "dispatching" cars. However, he notices that the last car from the first lane remains slightly in the intersection, blocking the path of the second lane. Because of this he decides to select the next lane that has a clear path. Cars from the third lane begin to move through the intersection freely until the traffic cop hears the urgent blare of someone's horn. He immediately stops the third lane and discovers that the horn-blower, waiting on a path through the intersection, is in the second lane that is now clear. He notices, however, that cars have again built up in the high priority lane so he allows the first lane to move. The first lane empties and the traffic cop begins to move cars out of the second lane, then the third lane. Finally, he turns to the fourth lane and motions for the cars to begin moving. One of the cars stalls out and now the fourth lane is waiting so the car can be removed. This move optimizes the entire intersection because each of the other three lanes has built up a line of cars again. The traffic cop starts moving the high priority lane but notices the mayor's car at the head of lane two. Being a model citizen as well as a member of the same political party, he immediately halts the first intersection to allow only the mayor's car clear passage.

Traffic Cop?

From reading the preceding scenario, it is obvious the traffic cop may not be qualified to move traffic. However, the scenario illustrates the fundamentals of the

The Only Integrated Solution



To The Automated Operations Puzzle!

Meet The CONTROL Team

CONTROL-M

- *State-of-the-Art Job Scheduler*
- *NO system/JES hooks, SMF exits or SVCs*
- *2 hour Installation*
- *ISPF, ROSCOE or CICS On-Line Facility*
- *Forecasting/Simulation*
- *Automatic Date/Control Card changes*
- *Fastest schedule implementation*
- *Automatically Open/Close CICS files*
- *Can be integrated with CONTROL-D and CONTROL-R*

CONTROL-R

- *Automated JOB Restart System*
- *Eliminates manual intervention*
- *Automatic catalog/GDG adjustment*
- *Modifies JCL as required*
- *Eliminates lost time and the errors associated with reruns*
- *NO system/JES hooks, SMF exits or SVCs*
- *Integrated with CONTROL-M*

CONTROL-D

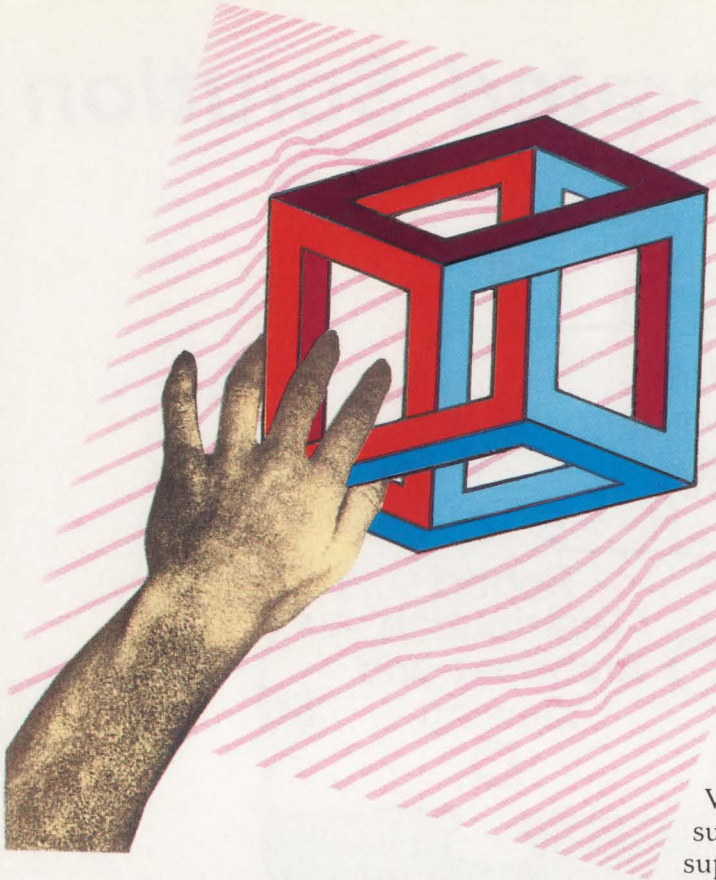
- *Automated Report Distribution, Viewing, and Archival System*
- *NO system/JES hooks, SMF exits or SVCs*
- *Easy On-Line report definition and viewing using ISPF, ROSCOE or CICS*
- *NO permanent database required*
- *True laser printer technology*
- *Printer workload balancing*
- *Can run independently or integrated with CONTROL-M*

Products Designed To Work Together

TONE® Software Corp.

1735 S. Brookhurst Street • Anaheim, California 92804 • (800) 833-8663 • (714) 991-9460 • TELEX: 4974583 • FAX: (714) 991-1831

CIRCLE #144 on Reader Service Card ▲



VPS Support Of Network Printers On MVS Is A Complex Task Made Simple.

VPS has been used to replace other products, such as: IBM's 328x/ADMPRINT/DSPrint, CICS supported printing, SASWTR®, RJE and many others, with a single task to drive all your 3270 family printers directly from the JES spool (including cross domain VTAM printers).

1400 Sites use VPS as their shop standard 3270 family printer driver.

- Printers supported include the full array of 3174/3274 attached printers, including IPDS support for 42x4's, HP and Xerox lasers, plotters and PC printers.
- VPS runs as a VTAM application. NO system modification. NO JES maintainance.
- Automatic forms control, full FCB support, dial-up PC printer, printer pooling and "hot" printers are all supported.
- Full screen "ISPF-like" command interface for CICS, TSO and ROSCOE permit end-user control of printers with totally menu driven command entry.

CALL or write for more information, or to arrange for a free TRIAL – ATTN: Marketing Dept.

LRS

Levi, Ray & Shoup, Inc.

Specializing in Computer Systems Software

2387 West Monroe • Springfield, Illinois 62704 • (217) 793-3800

Telex (510) 60-00675

® Trademark of SAS Institute, Inc., Cary, NC.

CIRCLE #121 on Reader Service Card ▲

VSE dispatcher and scheduling system. The dispatch mechanism resembles the traffic cop deciding which task in the system deserves its resources, yet it does so without any political prejudice in the decision-making process unlike the traffic cop in the example.

Although there exists a large number of external modules and programs within the system, the heart of the VSE operating system is known as the *supervisor.* The supervisor is a large Assembler language program shipped by IBM in source code format containing several macros (for example, SGSCHED, SGSTAR and so on). A few default ''sups'' (pronounced ''soups''!) are sent with the product install for VSE and in many cases you may not need to assemble one. For purposes of VSE problem debugging as well as for this article's focus, it may be enjoyable to reference a listing.

On an IBM System/370 architecture CPU, the system is known as an ''interrupt-driven'' system. There are five classes of interrupts including Input/Output (I/O), program check, ''Supervisor-Calls'' (SVCs) and external and machine check. Both the SVC and program check interrupts are caused by the execution and failure thereof, respectively, of an instruction. The VSE supervisor is comprised of routines for all five classes of interrupts, dubbed ''interrupt handlers,'' and the task scheduling system called the ''dispatcher.'' In our last peek ''under VSE's covers'' in the November/December 1988 MAINFRAME JOURNAL, the major control block structure of the VSE system was discussed in detail. In this article, the supervisor's interrupt and scheduling mechanism will be explored in depth.

Who's On First?

Fortunately, the supervisor's scheduling technique is not subject to the kind of confusion that the Abbot and Costello routine exemplified. Its role is to allow the highest priority task in the system that is ready to run to use the CPU. The interrupt process is an asynchronous one however. That is to say an interrupt is not scheduled, it simply ''happens'' as a result of an event. An example is the completion of an I/O request.

When a program is executing and an interrupt occurs, the CPU's microcode stores the current Program Status Word (PSW) and interrupt information in a fixed location in low storage. Depending on the class of interrupt, a new PSW is loaded from low storage for the appropriate routine to handle the interrupt (see Figure 1). The second word or address portion of the PSW will point directly into the interrupt handler routine in the supervisor. (A note of caution: several OEM vendors routinely modify the new PSWs as a way of ''hooking'' the system.) The new PSW that is loaded has all interrupt mask bits off in order to single thread the interrupt routine.

In all cases, final exit of the interrupt handlers is through the dispatcher or task selection routine.

The interrupt routines store registers nine through 13 in a field in low core called ERA so they may establish addressability. At this point a routine is executed called GENeral ENTry (GENENT) to save the interrupted PSW and registers that were in effect prior to the interrupt into the task's save area located off of the TCB. How did we know which TCB to address? Remember, in the Fixed Control Area the major task control blocks are anchored here for the task currently ''dispatched'' (see Figure 2). Thus we load the pointer to the TCB from low core address X'260' and address the save area at TCB +X'0C'. Then the appropriate interrupt routine handles its business based on the interrupt information. Following is a brief look at two of the most common interrupt routines.

In the event of an SVC interrupt, the supervisor first checks for a ''fast-path'' SVC (SVC107). Then if it *is* a fast-path SVC, the supervisor branches directly to its service routine. Otherwise, it takes the interrupt code at hex location X'88' (see Figure 1) and uses that code as a ''vector'' off of the SVC table (SVCTAB) to locate a service routine. SVCs are requests from a task in the system asking the supervisor to perform services such as start an I/O request (EXCP macro for SVC00) or get the time of day (GETIME macro for SVC34). If the SVC routine frees a resource or changes the calling task's dispatch status, the SVC handler exits to the task selection routine in the dispatcher. Otherwise, a special exit is

FIGURE 1

Storage Locations Used for Interrupt Processing

Interrupt Type	Old PSW Address	New PSW Address	Interrupt Information
External	X'18'	X'58'	X'86-87' (external code)
SVC	X'20'	X'60'	X'88-8B' (SVC code)
Program Ck.	X'28'	X'68'	X'8C-8F' (PgmCk code)
Machine Ck.	X'30'	X'70'	X'E8-EF' (MCK code)
I/O	X'38'	X'78'	X'BA-BB' (I/O address)

FIGURE 2

Fixed Control Area (PSA)

Low Core Address (HEX)	Description
240-241	Flags
242-243	Routine Indentifier (RID)
244-247	Current Time Pointer
248-24B	Address of PIB2TAB
24C-24F	Address of System PCB
250-253	Address of SCB Address Table
254-257	Address of Current SCB
258-25F	Partition Selection String (PSS)
260-263	Pointer to Current TCB
264-267	Pointer to Current TIB
268-26B	Pointer to Current PIB
26C-26F	Pointer to Current PCB
270-273	Debug Header Address
274-277	Debug Parameter Address
278-27B	Address of First Level Interrupt Handler
27C-27F	Address of Dispatcher
280-2BF	ERA Save Area
2C0-2C3	Address of TIBATAB-4
2C4-2C7	Address of PCB Address Table
2C8-2C9	Number of Page Frames Available
2CA-2CB	Minimum Number of Pages Not Fixed
2CC-2CD	Save Field for RID

taken to the label SAMETASK in the dispatcher in order to simply re-dispatch the calling task.

An I/O interrupt is processed based on the device address that has been stored at low core address X'BA' (see Figure 1). The device address is used to scan the PUB table and set up the proper addressing to the control blocks, that is the CHANQ, PIB, TIB, TCB and PCB. The Channel Status Word (CSW) that was also stored at low core address X'40-47' at the time of the interrupt is examined to evaluate the status of the interrupt. If final status is indicated for a specific I/O request, the CCB is posted complete, the I/O is dequeued from the CHANQ and the task that had requested the I/O is posted "ready-to-run." At this point, the interrupt handler branches to the channel scheduler whose responsibility is to scan the PUB table for the next device on the interrupting channel with an outstanding I/O request and to initiate a Start-I/O (SIO) for that device.

And now, one lucky TASK will be selected to . . .

In all cases, final exit of the interrupt handlers or any other supervisor service routine is through the dispatcher or task selection routine. This is the heart of the selection logic in the supervisor and its responsibility is to determine the next task to use the CPU's resources.

The control blocks described in the previous article are used in depth in this process. For speed considerations, this process uses a "top-down" approach. That is to say, the first available partition is selected to run, if any, and then the task within the partition is selected. The code that accomplishes task selection is quite elegant and because of its inherent repetitive execution, also quite efficient.

In order to determine the highest priority partition ready to run, the dispatcher uses two major blocks called the Partition Selection String (PSS) located at X'258' and the PPRTYOWN table located at X'2C4'. Both of these blocks contain information for each partition in *priority* sequence and thus are rearranged by the VSE PRTY command as well as by the system partition balancer. The PSS string contains one bit for each partition in the system. If the bit is zero, the partition is not ready to run and likewise if it is set to one, the partition is ready to run. The PPRTYOWN table contains fullword pointers to each PCB in the system in *priority* order. The dispatcher first locates a non-zero bit in the PSS and then uses the relative location of the bit to index into the PPRTYOWN table to locate the PCB for that partition. An example of this technique is shown in Figure 3.

After locating the ready partition, the highest priority ready task *within* the partition must be located. For this the dispatcher uses the Task Selection String (TSS) located within the partition PCB + X'18'. Within the TSS there is one bit for each of up to 31 subtasks and the maintask. These bits are in *task priority sequence* within the PCB and correspond on a one-for-one basis with the Task ID String (TIDSTR) bytes located at PCB + X'20'. Like the PSS scan, the dispatcher again uses a Translate and Test (TRT) instruction to determine the relative task that is ready to run within the ready partition and uses this as an index to load the Task ID (TID) (see Figure 3). From the TID, all other necessary control blocks can be calculated using TIBATAB and so on and the task can be activated.

Through a graceful and speedy process the dispatcher has determined the task in the system that is worthy of its resources. It then initializes the major control block pointers including the PCB, TIB, TCB and PIB in the Fixed Control Area and additionally stores the TID at X'5A' into SYSCOM indicating the last task dispatched and the partition's COMREG at X'16' into low core. Then, the task's save area is addressed, the PSW that was saved at the moment of GENENT during the interrupt is restored to ERA and the supervisor exits its code and activates the task by restoring the task's general registers and issuing a load PSW instruction (see Figure 3).

Problem: BATCH and Application programs do not fully utilize VSAM buffers.

Solution: BIM-BUFF—Dynamic VSAM Buffer Management System.

For faster VSAM processing immediately, use BIM-BUFF

BIM-BUFF is a product which is designed to significantly increase the performance of VSAM in every DOS/VSE installation. It does this in a way which is transparent to the programs involved, does not alter any VSAM files, and does not make any modifications to VSAM itself. While each installation is different, experience with some DOS/VSE installations has shown potential savings to be astounding. The following savings have been achieved in benchmarks of real life applications:

	Physical I/Os	Elapsed Time
Typical sequential access	33%	10-50%
Typical random access	25-50%	40-60%
Clustered random access	99%	95%

In fact, the performance benefits can be so significant, that it may be possible in some cases to defer the purchase of new hardware. Perhaps best of all, these savings can be realized almost immediately. BIM-BUFF installs in minutes with no need to change any existing files, programs or JCL.

BIM-BUFF is priced at $2,400 for a permanent license, $1,200 on an annual lease or $120 on a monthly rental.

B I Moyle Associates, Inc. has been dedicated to providing cost effective software solutions, which improve system performance and user productivity, for 10 years. For more information on BIM-BUFF, or any of our other quality software products and services call Jim Kingsbury at 612-933-2885 today.

BIM

B I MOYLE ASSOCIATES, INC.
5788 Lincoln Drive
Minneapolis, MN 55436

612-933-2885
Telex 297 893 (BIM UR)
Member Independent Computer Consultants Assn.

CIRCLE #66 on Reader Service Card ▲

Haven't we been here before?

We have now gone full circle from task execution to interrupt to task execution again. The rather swift selection mechanism of the VSE dispatcher is necessary when you consider the literally millions of passes through this code during the life of an IPL. Surely you must agree that the VSE dispatcher is "a traffic cop extraordinaire."

Now, the control blocks written about as old friends have begun to take life in the heart of VSE: the supervisor. These articles are building to the ultimate discussion of VSE problem determination and debugging. In the next peek "under the covers," some aspects of VSE storage management and layout will be examined.

ABOUT THE AUTHOR

Eric L. Vaughan is president of Smartech Systems, Inc., a systems software firm specializing in the VSE market. Vaughan has more than 10 years of technical experience with VSE and VM operating systems and has worked extensively in the field of telecommunications. SMARTECH SYSTEMS, Inc., 10015 West Technology Blvd., Dallas, TX 75220, (800) 537-6278.

FIGURE 3

Example Of Task Selection Routine

```
*
* Determine the highest priority partition ready to run
*
         SLR      R1,R1                 CLEAR REGS . .
         LR       R2,R1                 . . FOR TRT
         TRT      PSS(LPSS),PREXTTAB    SCAN FOR READY PARTITION
         BZ       ALLBND                GOTO ALL BOUND IF NONE
         S        R1,APSS               CALCULATE DISPLACEMENT
         SLL      R1,5                    . TO THE PARTITION'S
         LA       R5,0(R2,R1)             . . PCB POINTER
         L        R5,PPRTYOWN-4(R5)     LOAD PCB ADDRESS
         USING    PCBADR,R5             SET TO DSECT
*
* Determine the highest priority ready task within partition
*
         TRT      TSS,PREXTTAB          FIND READY TASK
         LA       R8,TSS                CALCULATE DISPLACEMENT
         SR       R1,R8                 . . TO TASK'S
         SLL      R1,3                  . . . ID
         SRL      R2,2                  . . . . WITHIN
         AR       R1,R2                 . . . . . PCB
         IC       R1,TIDSTR-1(R1)       LOAD TASK ID
*
* Task is selected, exit supervisor and activate task
*
         L        RB,TCBSAVE            LOAD PTR TO SAVE AREA
         USING    SVEARA,RB             ADDRESS SAVE AREA
         . . . .
         . . . .
         MVC      ERA(8),SVEPSW         SETUP PSW
         LM       R9,R8,SVER09          LOAD TASKS GENERAL REGS
         MVI      RID+1,USERTID         IDENTIFY TASK
         LPSW     ERA                   ACTIVATE THE TASK
```


DB2 *from page 32*

quired. The IBM sample tables have been granted to PUBLIC, so anyone who has access to QMF can run queries against these tables.

QMF training requirements have proven to be minimal. I have written a users' manual and provide a three-hour introductory class for most users. If the user is at a remote site or timing for a class does not work out, IBM's *Learner's Guide* has proven to be useful. The guide is a self-paced, hands-on course that has the new user write queries from the sample tables and create forms for the results. It covers the basics but unfortunately falls short of explaining how to summarize and group data.

I have had the best response training people who have similar information needs rather than similar skill levels. This allows me to customize a class based on the data and concentrate on the QMF functions they will use the most. The functionality of QMF is extensive, but most users will only use a small portion of those capabilities to generate their reports.

One of the most satisfying aspects of my job is the feedback I get from new users and the users of new tables. Without fail, someone calls to tell me how much time they have been able to save by using QMF. They are delighted and usually full of ideas for new tables that we should create.

Having supported a number of other 4GL, user-friendly reporting tools, I can honestly say that I have been pleasantly surprised by my experience with QMF. I have seen real user acceptance of this product and I am continually impressed by the minimal support that is required for end users.

ABOUT THE AUTHOR

Mary Jo Cater is a senior systems analyst at Occidental Chemical Corporation. She has more than seventeen years experience in data processing, training and research. An active member of the Dallas area DB2 Forum since 1985, Cater is currently serving as secretary of that users' group. Occidental Chemical Corporation, PO Box 809050, Dallas, TX 75380, (214) 404-3668.

VSAM Optimization And Design

Free Space Selection, I/O Buffer Allocation, Space Allocation And Index Options

By Eugene S. Hudders

Organizational options, flexibility and performance are some of the reasons why VSAM is a widely-used access method. In addition, VSAM is relatively compatible between operating systems especially at an external level.

In this series of articles important areas of VSAM design and performance are explored. Control Interval (CI) Size for data and index, Control Area (CA) Size and CI/CA Splits and their effect on processing were described in detail in the January, 1989 issue of MAINFRAME JOURNAL.

Free space selection, I/O buffer allocation, space allocation and index options are addressed in this article.

Free Space

VSAM Key Sequence Dataset (KSDS) files provide the capability of addition and deletion of records as well as alteration of the records' length. In order to provide space for this to occur, the FREESPACE parameter can be specified. Space can be reserved at the CI and/or the CA level. The major objective is to avoid CI/CA splits. Splits, especially CA splits, take up resources and time. A CA split in an on-line environment can adversely affect response time. Thus, the positive effect of free space is to be able to add or expand records with minimum overhead. Also, during split processing the integrity of the dataset is in jeopardy until the index is updated on the disk.

There are some negative effects to using free space. Free space will increase the space requirements and, thus, sequential processing time. If a poor free space selection is combined with a poor CI and CA selection, then the results could be disastrous to the performance of the dataset. Therefore, specify free space only if needed. If you must specify free space, then use it judiciously.

One characteristic of free space is that it is *evenly* distributed across the dataset. Due to this, it is possible to have unusable space due to the insertion pattern. This leads to wasted space. Depending on the distribution of insertions, the user can vary the request for free space by specifying either CI or CA free space or both. If this does not fit a particular dataset, then the user should consider writing his own load program to create the free space where insertions occur. At the point where insertions are to be made, the load program can create dummy records. These dummy records are later deleted and the free space is created at the point where the insertions are to occur.

The selection of the free space percentage can also be a little confusing. The free space percentage is not necessarily based on the rate of growth of the dataset. CI free space is based on a percentage of the CI size, that is, bytes in a logical record, not dataset growth. A dataset may be growing at a 10 percent rate but if the CI size is 4K bytes and each logical re-

cord was 1,000 bytes, then a 25 percent CI free space would have to be requested to leave space for at least one logical record per CI. The CA free space is not much better because the idea is to leave completely empty CIs. Each CI can usually accommodate more than one logical record. Thus, using the above example (CI FSPC), if there were 150 CIs per CA and a 10 percent CA free space was selected, then 15 empty CIs would be left free. As each CI could hold up to four logical records, space for a total of 60 records has been allocated. This may be higher than the growth rate. When combined, CI and CA free space may actually reserve a much higher amount of space than the growth rate, especially considering growth rates are usually given in yearly percentages. If the dataset is reorganized monthly, then the actual space requirements for growth would only be one-twelfth of the yearly percentage based on an even distribution.

Unfortunately, a standard formula is not available to determine the percentage of free space to be used. Factors that will affect the free space percentage include the CI size, the logical record size, the CA size, the distribution of insertions, the growth rate and the frequency of reorganization. Trial and error is not necessarily the best method but it works. Use frequent LISTCATs to monitor results until a good balance is received between splits and free space reserved. The general guidelines are:

- If evenly distributed insertions and/or record expansions, use CI and CA free space (especially if on-line additions)
- If uneven distribution, use CA free space
- If additions are at the end of the dataset, leave no free space but allocate a larger primary
- If no insertions and/or record expansions, do not use free space.

The LISTCAT will provide information (splits) which will give you an idea as to the effectiveness of the free space selected. LISTCATs should be produced before and after dataset updates to measure results.

I/O Buffer Allocation

An important performance feature of VSAM datasets is the buffer space allocation that can be specified. An important fact is that the defaults assumed by VSAM in the cluster definition are not good for performance in most situations. For sequential processing, additional data buffers can be specified. For direct processing, additional index buffers can be specified. When accessing the dataset sequentially, VSAM starts at the highest level index following the chain down until the first sequence set index record is found.

VSAM will process all the CIs in the CA associated with the sequence set record. VSAM will then follow a forward pointer (Horizontal Pointer) in the sequence set record to locate the next sequence set (CA) to process. Sequence set records are not read ahead; therefore, additional index buffers are usually of no value. Data CIs are read ahead if additional data buffers are available. Having additional buffers usually improves performance. Too many buffers can increase paging and device/channel utilization

Is Your DASD Space Being Wasted????

THEN YOU NEED

DP VSAM/SR

THE **ONLY** PRODUCT THAT CAN **IMMEDIATELY RECLAIM UNUSED DASD SPACE** ALLOCATED TO **VSAM** FILES. NO BACKUPS, DELETES, REDEFINES AND RESTORES OF THE VSAM FILES ARE REQUIRED. (MVS ONLY)

FOR FURTHER INFORMATION CALL OR WRITE:

dpt

DATA PROCESSING TECHNIQUES, INC.
21 LAKEVIEW DRIVE
WEST ORANGE, N.J. 07052
PHONE: 201-325-3251

CIRCLE #25 on Reader Service Card ▲

CAN YOU AFFORD NOT TO TUNE YOUR VSAM FILES?

(VSAM problems don't just go away)

Unlike the highly visible 'explosive' problem which causes havoc and demands priority, VSAM problems tend to be 'corrosive' and often go unnoticed. The forgiving nature of VSAM will usually avoid a crisis, but can lead to expensive DASD and CPU inefficiencies.

ULTIMATELY, A SOLUTION IS NECESSARY!

Solution 1 — Acquire additional DASD, CPU power, technical people or add an extra shift ... This option is very expensive ... and only defers the problem.

Solution 2 — Acquire CBLVCAT ... This option involves a fraction of the cost, a nd can solve the problem in a fraction of the time.

CBL VCAT **VSAM Monitoring, Tuning/Optimizing and Modelling for any DOS, CMS, OS, XA system**

Call or write for a free trial and let us help you gain control of your VSAM files.
Tel: 416/746-4447 Compute (Bridgend) Ltd, 38 Guided Ct, Rexdale, Ontario, Canada

CIRCLE #38 on Reader Service Card ▲

which may affect the performance of the entire system. Too few buffers, such as the default value, can also affect the performance of the program especially if the CI size is small. The default data buffer space is twice the data CI size. After a certain number of buffers (for example, three or four), VSAM will attempt to read more than one buffer with only one physical read operation. This is a chained CCW I/O operation. In addition, if sufficient buffers are provided, VSAM will split the buffers so as to provide overlap processing and chained reading. Overlap processing is like having an alternate area available to process while filling up the alternate area.

Direct processing, on the other hand, makes more use of index buffers to search for the desired record. VSAM will use the extra index buffers to maintain the indices in storage for future reference and use. This will improve future data record access if the indices are in storage. Extra data buffers are usually of no value unless a spanned record is being read. The actual number of index records in storage depends on many factors such as the buffer pool management techniques (LSR versus NSR).

The default index buffer space is one buffer. Thus, for a three index level dataset, three reads to the index portion are required. Each index read destroys the contents of the previous level. This causes the indices to be re-read for every direct access. Ideally one would want to have extra index buffers in order to keep the higher level indices in storage so that they will not have to be re-read for every direct access.

There are several ways to specify extra buffers to a VSAM cluster. We will discuss some of the most common methods. The first method is to use the buffer space parameter (BUFFERSPACE) during the cluster definition. This is not recommended except for the base cluster when accessing an alternate index via the path in an on-line environment. The main disadvantages of this method are that any change in the CI size (data and/or index) will require a change in the BUFSP specification. The decision as to how many buffers of each type (data or index) is left to VSAM based on the mode of processing specified at dataset open time. This can be misleading if the dataset is opened for dynamic processing.

Another method is to run an IDCAMS ALTER to change the BUFFERSPACE allocation prior to execution and possibly after execution to restore the original values. This takes extra processing time and is not recommended.

A more dynamic alternative is to specify the required amount (BUFSP) in the execution JCL (// DD or // DLBL). This is the preferred method for batch processing. For DOS users, the amount of buffer space desired is defined through the BUFSP parameter in the // DLBL card. For MVS users, the BUFSP subparameter of the Access Method Parameter (AMP) can be used. The user has to determine the correct amount based on the CI sizes. Buffer space allocations in the JCL are also sensitive to change in CI sizes and mode of processing. This is similar to defining the buffer space at cluster definition and thus has the same disadvantages.

In order to better distribute the buffer allocations, MVS provides the subparameters BUFND and BUFNI also of the

Screen Painting and Conversation Prototyping for CICS and IMS/DC

Quick Screen 3270 is a new and easy to use PC based development tool. It will help you create screens and demo systems in minutes instead of hours!

Quick Screen 3270 features include:

- fast WYSIWYG (what you see is what you get) map entry
- advanced, powerful word processing and SPF type editing
- integrated map editing and conversation prototyping
- generate and import CICS/BMS and IMS/DC/MFS code
- 5 windows to work in
- context sensitive help and pop-up screens
- no programming background required for its use

Compare Quick Screen 3270 to SDF2 or to whatever you use today. Call or write for a free guide or a

FREE 60 Day Trial!

Integrated Systems Technology, Inc.
5 Chapel Hill Road, Short Hills, NJ 07078
(201) 376-3722

CIRCLE #68 on Reader Service Card ▲

VSAM-LITE

Automatic Data Compression for VSE

- Saves VSAM space by 50% or more
- Supports KSDS, ESDS and Managed-SAM
- Easy to use—*no* JCL changes, *no* program changes
- Reduces DASD I/O activity; Improves system performance
- Improves online response times; Speeds tape backups
- Analysis Utility forecasts actual space savings

Call now for your *FREE* copy of the Compression Analysis program that will show how much disk space your installation can save by using VSAM-LITE.

CALL...

Toll Free: **(800) 243-9542**
in Conn: **(203) 792-5100**

Universal Software inc
Brookfield Office Park
Brookfield, CT 06804

CIRCLE #112 on Reader Service Card ▲

AMP parameter in the // DD card. These subparameters provide the capability of requesting a specific number of buffers for the data (BUFND) and index (BUFNI). This selection is independent of the CISZ. Therefore, one can change the CISZ and VSAM will compute the necessary buffer space. These parameters also provide the user with a means of controlling the number of index and data buffers desired whenever the dataset is processed dynamically based on the user's knowledge of the dataset processing requirements.

A major point of discussion when dealing with buffering and sequential processing is the CISZ to be used. Are large CI sizes better than multiple data buffers? The answer lies in the amount of resources (for example, central storage) available. A CISZ in the 4-8K range coupled with 6-8 data buffers will yield good performance while maintaining adequate use of resources.

Considerations for on-line datasets will be discussed in the CICS/VS section. The request for buffers should be done through the File Control Table (FCT) utilizing the BUFNI, BUFND and STRNO parameters. The actual requirements will depend on the individual dataset being defined.

Space Allocation

When requesting space for a dataset, the important aspect is to ensure sufficient space in the primary so that no secondary allocations take place. The total space used after loading the dataset should be around 90-95 percent. This can be verified in a LISTCAT using the High Used RBA (HURBA) and the High Allocated RBA (HARBA). Additional space may be allocated if the dataset receives extensions. Secondary allocations should be taken as an indication that more space in the primary is required. Secondary allocations take up virtual storage and the dataset may be spread across a volume that may affect sequential processing. Also, when requesting space, be sure to request sufficient space for the secondary. Small secondary allocations (for example, one cylinder) can cause a high overhead when repeated secondary allocations are required.

Key ranges are another feature of space allocation where the user can distribute the data across several volumes. The idea is to provide a certain amount of overlap capability. There are certain things that must be observed. The split by key ranges has to be well planned to ensure a balance of total data and number of accesses. As each new volume is allocated, a new primary allocation is made. This can lead to unused space and/or over-used space if the data is not properly partitioned. The key ranges selected should include all key possibilities. Any key not included in the ranges will not form part of the dataset. If any overlapping is to exist, then the index records should be brought into storage. Otherwise, the access to the dataset will single thread through the index portion on disk.

Index Options

Several index options are available to the user. IMBED and REPLICATE make use of the disk areas to improve direct access. Another option is to bring in as many indices into virtual storage as possible. IMBED will take the Sequence Set Index (SSI) and write it with the data in its respective CA. As the whole first track of the CA is dedicated to the index, the

See VSAM Optimization *page 92*

The VSAM Performance

- **VSAM I/O Plus**™ slashes VSAM batch processing time up to 80 percent.
- **VSAM Assist**™ backs-up and restores VSAM files 70 percent faster than EXPORT & REPRO.
- **VSAM Data Compressor**™ cuts VSAM DASD space by 50 percent.
- **VSAM Space Manager**™ pools VSAM DASD and eliminates ABENDS due to lack of space.
- **VSAM Quick-Index**™ loads VSAM alternate indexes in one-fifth the time of BLDINDEX.
- **VSAM Mechanic**™ recovers or repairs broken ICF catalogs and VSAM data sets in minutes.

For more information call

800-638-9254

MVS only.

People

SOFTWORKS, INC.

CIRCLE #179 on Reader Service Card ▲

If...

One product and one hour could double VSAM performance

Then...

HYPER-BUF
VSAM buffer allocation expert

- **Shatters VSAM performance barriers**
  — Customers consistently report 25% to 80% reduction in EXCPs
- **Optimizes CICS buffers**
- **Installs in less than one hour**

Other VSAM products from Goal Systems, the Original VSAM Experts

/* **FAVER —** A VSAM file management tool with comprehensive reorganization and backup/restore capabilities.
VSAMAID — Tunes VSAM datasets automatically while eliminating inefficiencies. A cluster modelling, performance, and capacity planning tool.
MASTERCAT — An online and batch VSAM catalog navigation tool.
COMPRESSOR/MVS — Provides compression of VSAM datasets. Can triple disk capacity. Available for MVS.

To Find Quality Call 800-848-4640.

All products, except COMPRESSOR/MVS, are available for VSE and MVS.*/

Goal Systems International • 7965 N. High Street • Columbus, Ohio 43235 • 800-848-4640
Goal Systems International S.A.R.L. • 88 avenue de Wagram • 75017 Paris, France • Phone (1)42 67 55 55 • Telex: 641.094

CIRCLE #79 on Reader Service Card ▲

Capacity Planning Using Statistical Theory and Performance Data

Predicting Tomorrow's Response

By Philip C. Davis, III

ILLUSTRATION BY JEFF HILL

The inherent nature of an on-line system such as CICS implicitly provides a measure of the efficiency of a computer center and its service to the user community. This article addresses the problems encountered in performance analysis by presenting a technique that allows the assimilation of all data made available to the analyst. The theories presented here are the basic mathematical premise for a total tuning philosophy. The technique will allow its adherents not only to predict how their system will perform in the future, but also to help in pinpointing system problems before they severely degrade system performance. Only the basic foundation for the approach is presented here as a complete explanation of all the implications would require an article of substantial volume. The object in mind is the successful implementation of performance monitoring, statistical analysis and tuning to ensure the efficient use of resources resulting in faster response times, higher levels of productivity and enhanced user confidence in data processing.

Monitoring Tools — Collecting the Data

There are many software vendors who provide systems programmers with the tools they need to accomplish the goal of getting the most out of a system. However the questions arise, ''What do I look for in a monitor and once I have selected one, how can I use the information that it provides?''

An effective CICS monitoring tool must display data associated with system resources as they are being used. This feature makes it possible to ascertain how system resources are being utilized and under what circumstances a particular resource or transaction is degrading system performance. On-line, real-time intervention, to alleviate system problem events, is helpful to relieve system hangs and to prevent subsequent shutdowns. The dynamic character of CICS makes it difficult to evaluate the effect of any tuning effort on overall system per-

formance, thereby making evaluation of archived data essential.

A good CICS measurement tool must also be able to collect and archive system data so that reports can be generated to detail and summarize this data into daily, weekly and monthly reports. With this facility at hand, it is a simple matter to discover system trends and to effectively initiate system capacity planning. Although there are many facets of CICS that are reported on and archived, there are six system measurements that avail themselves as critical to predicting overall CICS performance. These are:

1. Transaction activity rate
2. File activity rate
3. Paging rate
4. File response time
5. CICS response time
6. CPU utilization.

Each one of these aspects of the system are related and affect one or more of the others. The problem now arises, "How do you take the information provided by a monitor and correlate it into a form that is useful?"

Mathematical Tools — Statistical Theory

Linear Regression and Correlation

Linear regression is a statistical method used to determine a linear equation that describes a straight line relating two variables. The linear equation is usually in the form:

$$Y = MX + B$$

Where:
- Y = dependent variable
- M = the slope of the line: $\frac{\text{RISE}}{\text{RUN}}$
- X = independent variable
- B = the Y intercept

Linear correlation is a statistical tool used to analyze the relationship between two variables and to determine the extent of that relationship. The linear correlation coefficient is found by using the equation in Figure 1.

A resulting value near one indicates a strong relationship between the two variables. A value of near zero indicates that the variables are only minimally related and a value near minus one indicates a strong inverse relationship. Table 1 gives you the corresponding significance between two variables.

Using, for an example, the value of R = .45 for 30 observations, it is found that the value is between the 95 percent and 99 percent columns giving us a greater than 95 percent significance between the two variables.

Assuming acceptable correlation between two variables, further analysis can be conducted and the exact nature of the relationship can be expressed. Using the following form of the linear equation, we can predict a Y value given an X value as in Figure 2.

Once your prediction is found using the linear equation above, you must qualify the prediction with a confidence interval. This can be expressed as the degree of certainty that your predicted mean will fall into a given range. It is found by using the equation in Figure 3.

FIGURE 1

$$R = \frac{N\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{N(\Sigma x^2) - (\Sigma x)^2}\sqrt{N(\Sigma y^2) - (\Sigma y)^2}}$$

Where:
- N = the number of paired samples
- Σ = the operation of summation
- X = the independent variable
- Y = the dependent variable

TABLE 1

"R-TEST" Table

# of Samples	80%	90%	95%	99%	99.9%
29	.245	.311	.367	.471	.579
30	.241	.306	.361	.463	.570
31	.237	.301	.355	.456	.562

Note: If the correlation coefficient is negative, use the absolute value but keep in mind that the relationship is inverse.

FIGURE 2

$$Y = \left[\frac{\frac{\Sigma x_i \Sigma y_i}{N} - \Sigma x_i y_i}{\frac{(\Sigma x_i)^2}{N} - \Sigma x_i^2}\right] X \text{ (Hypothetical or previously calculated X)} + \left\{\frac{\Sigma y_i}{N} - \left[\frac{\frac{\Sigma x_i \Sigma y_i}{N} - \Sigma x_i y_i}{\frac{(\Sigma x_i)^2}{N} - \Sigma x_i^2}\right]\left(\frac{\Sigma x_i}{N}\right)\right\}$$

Where:
- Y = predicted value of dependent value
- X = hypothetical or previously calculated value of independent variable
- Y_i = singular occurrences of Y } Data
- X_i = singular occurrences of X } Data
- N = number of paired occurrences of X and Y

FIGURE 3

$$\mu = \bar{X} \pm \frac{S_x}{\sqrt{N}} Z$$

Where:
- $\bar{X}$ = your predicted mean value
- S_x = standard deviation expressed as:

$$\sigma = \sqrt{\frac{\Sigma(x - \bar{X})^2}{N}}$$

- N = number of paired observations
- Z = value from table of Z-scores

Z-SCORES TABLE

Degree of Certainty	Z-Score Value
60	.84
65	.94
70	1.04
75	1.15
80	1.28
85	1.44
90	1.65
95	1.96
99	2.58

Predicting Next Month's Response — What to Do with the Data

Now that you have been severely beaten around the head with statistical theory and equations, what, you may ask, can I do with all of this? The answer is to plug and chug. That is, plug in your data and chug through the equations.

Table 2 is a typical month's worth of performance data provided by a typical monitor. We will use this data as an example.

TABLE 2

PERFORMANCE DATA

DATE	C.I.C.S. RESPONSE TIME	FILE I/O RESPONSE TIME
03/01/88	0.4301 = = = =	0.0330 = = = = =
03/02/88	0.7862 = = = = = = = =	0.0343 = = = = =
03/03/88	0.3682 = = = =	0.0310 = = = =
03/04/88	0.3311 = = = =	0.0287 = = = =
03/05/88	0.2062 = =	0.0310 = = = =
03/06/88	0.1834 = =	0.0262 = = = =
03/07/88	0.3390 = = = =	0.0249 = = = =
03/08/88	0.3280 = = =	0.0281 = = = =
03/09/88	0.3204 = = =	0.0293 = = = =
03/10/88	0.3480 = = = =	0.0291 = = = =
03/11/88	0.3131 = = =	0.0256 = = = =
03/12/88	0.1831 = =	0.0297 = = = =
03/13/88	0.1661 = =	0.0243 = = = =
03/14/88	0.3366 = = = =	0.0263 = = = =
03/15/88	0.3578 = = = =	0.0276 = = = =
03/16/88	0.3481 = = = =	0.0257 = = = =
03/17/88	0.3440 = = = =	0.0267 = = = =
03/18/88	0.3202 = = =	0.0283 = = = =
03/19/88	0.1837 = =	0.0299 = = = =
03/20/88	0.1931 = =	0.0266 = = = =
03/21/88	0.2981 = = =	0.0236 = = =
03/22/88	0.3305 = = = =	0.0243 = = = =
03/23/88	0.6927 = = = = = = =	0.0340 = = = = =
03/24/88	0.3394 = = = =	0.0299 = = = =
03/25/88	0.3492 = = = =	0.0303 = = = =
03/26/88	0.1783 = =	0.0295 = = = =
03/27/88	0.2431 = = =	0.0258 = = = =
03/28/88	0.3353 = = = =	0.0249 = = = =
03/29/88	0.3128 = = =	0.0268 = = = =
03/30/88	0.3743 = = = =	0.0297 = = = =
03/31/88	0.3292 = = =	0.0283 = = = =
TOTALS	0.3583 = = = =	0.0282 = = = =

For reasons of brevity I will not cover the process of multiple regression that is required to establish the relation between the six elements for which we have collected data. To illustrate the application of the theory, I will use CICS response time as it relates to file response. This will require the use of the equation for finding the linear correlation coefficient discussed earlier, but first you must break down the elements as required by the equation:

N = the number of paried observations = 31
ΣX = the summation of file I/O times = .8734 secs.
ΣY = the summation of CICS response times = 10.1693 secs.
ΣXY = the summation of the products of XY = .2925295 secs.
ΣX^2 = the summation of (I/O times)^2 = .0248424 secs.
ΣY^2 = the summation of (CICS response)^2 = 3.8489436 secs.

We now plug our values into the equation in Figure 4.

FIGURE 4

$$R = \frac{(31)(.2925295) - (.8734)(10.1693)}{\sqrt{(31)(.0248424) - (.8734)^2}\sqrt{(31)(3.8489436) - (10.1693)^2}}$$

$$R = \frac{9.0684145 - 8.8818666}{\sqrt{.7701144 - .7628276}\sqrt{119.31725 - 103.41466}}$$

$$R = \frac{.1865479}{(.0853628)(3.9878052)}$$

$$R = .5480091$$

Using Table 1 you will find the linear correlation coefficient of .548 for 31 samples gives a greater than 99 percent significance.

Now that you have established that the relationship between file I/O and CICS response is greater than 99 percent significant, plug the values into the linear equation which will then serve as a model for the relationship (see Figure 5).

Reduction of this expression results in a more familiar form, giving us an equation that best describes the line correlating file I/O times and CICS response for the month of given data (see Figure 6).

FIGURE 5

$$Y = \left[\frac{\frac{(.8734)(10.1693)}{31} - .2925295}{\frac{(.8734)^2}{31} - .0248424}\right] X + \left\{\frac{10.1693}{31} - \left[\frac{\frac{(.8734)(10.1693)}{31} - .2925295}{\frac{(.8734)^2}{31} - .0248424}\right]\left(\frac{.8734}{31}\right)\right\}$$

FIGURE 6

$$Y = \left[\frac{.2865118 - .2925295}{.0246073 - .0248424}\right] X + \left\{.3280419 - \left[\frac{.2865118 - .2925295}{.0246073 - .0248424}\right](.0281742)\right\}$$

$$Y = \left[\frac{-.0060177}{-.0002351}\right] X + \left\{.3280419 - \left[\frac{-.0060177}{-.0002351}\right](.0281742)\right\}$$

$$Y = 25.596342\ X + \left\{.3280419 - (25.596342)(.0281742)\right\}$$

$$Y = 25.596342\ X + \left\{-.3931146\right\}$$

$$Y = 25.596342\ X - .3931146$$

We can now use this equation to predict CICS response given any file response by replacing the X variable with a hypothetical or previously predicted value for file response.

Say that we have predicted by using similar techniques as explained above a mean file I/O time of .030 for next month. If we just plug that value into the X variable, we can then determine what our CICS response for next month should be:

$$Y = 25.596342(.030) - .3931146$$
$$Y = .3747757$$

Validation of this prediction is accomplished by providing a confidence interval within which the prediction will fall (see Figure 7).

FIGURE 7

$$\mu = \bar{Y} \pm \frac{S_y}{\sqrt{N}} Z$$

Where:
$\bar{Y}$ = predicted CICS response mean = .3747757 secs.
S_Y = standard deviation in the form:

$$\sigma = \sqrt{\frac{\Sigma_i (y_i - \bar{Y})^2}{N}}$$

Where:
Y_i = each observation of CICS response
Y = mean of CICS response = .3280419 secs.
N = number of observations = 31

$$\sigma = \sqrt{\frac{.5129867}{31}}$$

$$\sigma = .1286389$$

* Z = Z-Score derived from Z-Score table: we will use a 99% degree of certainty, as the degree of significance in the predicted value is > 99% = 2.58

$$\mu = .3747757 \pm \frac{(.1286389)}{\sqrt{31}} \quad (2.58)$$

$$\mu = .3747757 \pm .0596089 \quad @\ 99\%$$

This expression is telling you that with a 99 percent degree of certainty we can expect, given a file response mean of .030, that our CICS response mean will fall into the range between .31517 and .43438. (Note: traditionally the degree of certainty is a subjective call based on factors concerning confidence in the measurement of the data collected.)

What inferences can be made with this information? Suppose that the actual mean value of the predicted month for file I/O

See Statistical Theory *page 94*

Problems:

1. **CICS, IMS, VTAM and TSO users need access to an online editor and JES.**
2. **General user access to ISPF is limited because of required use of TSO and its associated system overhead.**

Solution: **BIM-EDIT/MVS—Online Editor System**

Online Editing and Access to JES from CICS, IMS, VTAM or TSO

BIM-EDIT/MVS has been designed to eliminate many of the problems associated with using TSO/ISPF. Because TSO is required to use ISPF, general user access to it is sometimes limited because of system overhead restraints or the fact that the user simply does not have access to TSO. BIM-EDIT/MVS runs in its own region independent of any MVS subsystem and can be accessed directly from VTAM, CICS, IMS or TSO concurrently. Unlike TSO/ISPF all BIM-EDIT/MVS users run in the single BIM-EDIT/MVS region. Due to single region implementation, BIM-EDIT/MVS should support as many as four times the number of users as TSO/ISPF, using a similar amount of system resources.

Also, BIM-EDIT/MVS is a very powerful, flexible and easy to use editor which contains many features which should greatly enhance the productivity of your programming staff as well as other end users needing access to an on-line editor.

Here are just a few of the many reasons you should look at BIM-EDIT/MVS:

- Commands can be entered in any environment, eliminating the need to exit one function and enter another. For example, while editing a member, a command can be entered to display the JES queues.
- Commands may be chained together in any combination, executed as a group and can cross "modes".
- Commands, command groups and BIM-EDIT procedures can be assigned to PF or PA keys for execution by user.
- Up to 9 active concurrent BIM-EDIT/MVS sessions per user, rotate between them via a PF or PA key.
- Access to JES spooling queues. Jobs and job output can be listed, altered and purged. JES queue directories can be listed.
- Access to partitioned data sets (PDS's). Members can be created, listed, edited, and purged. Directories can be listed.
- Stack area provided to facilitate moving text between members.
- BIM-EDIT/MVS can be accessed on-line from VTAM, CICS, IMS, TSO, and also from its batch utility and from the user application interface, concurrently.
- Single-region implementation of BIM-EDIT/MVS, unlike ISPF which requires a region for each user, should allow BIM-EDIT/MVS to support up to four times the number of users as ISPF utilizing a similar amount of system resources.
- Tight source control with member auditing and stamping, purge control mechanism and checkin/checkout function.
- A powerful easy to use procedure processor.
- Flexible and straightforward security.
- Outbound screen data compression for optimum response time, including suppression of characters previously at the same screen position.
- Automatic disk space compression.
- Full function Electronic Mail System.
- Maximum compatibility with BIM-EDIT/DOS.

BIM-EDIT/MVS is priced at $11,200 for a permanent license, $5600 on an annual lease or $560 on a monthly rental.

B I Moyle Associates, Inc. has been dedicated to providing cost effective software solutions, which improve system performance and user productivity, for 10 years. For more information on BIM-EDIT/MVS, or any of our other quality software products and services call Jim Kingsbury at 612-933-2885 today.

BIM

B I MOYLE ASSOCIATES, INC.
5788 Lincoln Drive
Minneapolis, MN 55436

612-933-2885
Telex 297 893 (BIM UR)
Member Independent Computer Consultants Assn.

CIRCLE #115 on Reader Service Card ▲

BEFORE

your systems support becomes terminal...contact ISS.

Some of our services include:

- System generation DOS/VSE, MVS VM, CICS, IMS, VTAM
- Conversion CPU upgrades: 43xx, 303x, 308x, 309x
  Telecommunications: BTAM, VTAM
  OEM Software Packages: Disk and Tape Management
- Performance evaluation & tuning
- Resource capacity planning
- On-demand technical support
- 24-hour emergency support
- Site survey (software & hardware)
- On going systems support
- Remote systems support

USA (818) 966-0227
LONDON 01-642-4242
BRUSSELS 01-736-9713

A Galliard Company

ISS

Managing DASD Performance Can Be a Pleasure Trip... Not a Survival Course

DASDMON will revolutionize the way you manage MVS DASD performance.

Traditional DASD tuning techniques are being outpaced by faster CPUs and increasing user demands for more online data. Although cache controllers, data spaces and hiperspaces will relieve some of the strain, they also create new complexities. Tuning DASD can feel like struggling through a survival course.

DASDMON takes you away from it all

DASDMON offers you system managed performance today. By cutting through the complexities of conventional tuning, managing DASD performance is more automatic and streamlined than ever before. Once you've set response time objectives, just sit back and...

- **DASDMON** will identify problems by continually measuring individual I/O response times and comparing them to your objectives.
- **DASDMON** will determine the cause of problems by gathering and analyzing data. No longer will you spend hours scanning reports.
- **DASDMON** will present solutions to your problems in precise English, and simulate their effects to ensure that a new problem is not introduced.
- **DASDMON** will select the best possible set of solutions so you can implement recommendations with confidence.

And, if you have any questions or concerns along the way, our experienced technical support staff will be glad to guide you—24 hours a day.

To maximize your system's overall performance, use DASDMON with our DASD Performance Optimizers. These products eliminate DASD I/Os, resulting in automatic response time reduction.

For more details, call **(800) 323-2600** or **(412) 323-2600** in Pennsylvania. Ask for Extension **333.**

Performance and Optimization products from Duquesne Systems. **Enjoy the results.**

DUQUESNE SYSTEMS

Two Allegheny Center
Pittsburgh, PA 15212

CIRCLE #80 on Reader Service Card ▲

ISPF Techniques *from page 39*

little common sense, the additional resources consumed by ISPF are more than offset by improvements in programmer productivity.

When running ISPF in TSO, two factors can really make the difference between a responsive, low overhead system and a slow resource hog. Both significantly reduce DASD I/O. Commonly-used portions of TSO and ISPF should be in the Link Pack Area (LPA). All datasets should be optimally blocked. For 3380 DASD, this normally means the largest block size possible that is less than or equal to 23,476 characters (32,760 is the largest ISPF can handle). Although not an insurmountable problem, attention should be paid to programs that will read the dataset to be sure they can accept blocks that large. This usually means modifying buffer size parameters for compilers and utilities and removing any hard-coded block sizes for in-house COBOL programs. MVS' rule about concatenating datasets with JCL DD or TSO ALLOC statements, in which no dataset can have a larger block size than the first dataset of the list, is another area to watch. Coding a DCB = BLKSIZE = 32760 on the first DD of a concatenated set is the safest solution to that potential problem.

Your Turn

You have had a chance to read some of the ISPF techniques I have collected from my own experience and from those people I have worked with over the last few years. Now, it is your turn. If you know or find some ISPF techniques that could help make other readers more productive, please take a moment to tell us about them here at MAINFRAME JOURNAL either by the Feedback Card or by letter. If there is sufficient material, we will keep you updated on newly-discovered techniques in future issues.

ABOUT THE AUTHOR

Jon E. Pearkins, CSP, CDP is president of Certified Software Specialists Ltd. in Edmonton, Alberta, Canada. He has spent almost all of his 15 years of data processing in technical support both as a hands-on manager and systems programmer. Certified creates and supports its own line of IBM mainframe systems software. Certified Software Specialists Ltd., 54015 Range Road 212, Ardrossan, Alberta, Canada, T0B 0E0. (403) 998-0607.

FDR • ABR • CPK • FDR • ABR • CPK • FDR • ABR • CPK •

VSAM Optimization *from page 81*

SSI record is repeated the number of times that it will fit into the track. This reduces rotational delay as any record on the track would have the same information. The cost is one track of every CA in the dataset. If the CA size was specified incorrectly, then the cost can be high. If the CA size is set to the maximum (one cylinder), then the cost will vary from around three percent (3350) to eight and one-third percent (3375). On a 3380, IMBED represents almost seven percent of a cylinder.

If DASD space is critical, IMBED may make it worse especially if a poor CA size is defined. As discussed earlier, IMBED is not recommended for datasets with less than one cylinder in size when the CA is made to its maximum size. This is because only one index record is generated and should be in storage. Also, if the dataset occurs in many splits, the updating of the index will take additional time because a full track has to be updated rather than an individual record.

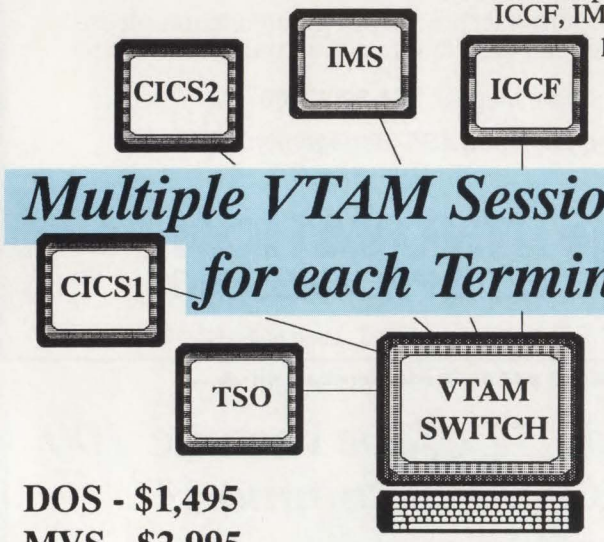
REPLICATE instructs VSAM to assign one full track to each index record in the index portion of the dataset. Since each index record is assigned its own track, the index record is repeated the number of times it fits into one track reducing rotational delay in the access. Once again, extra disk space is used.

By far the best option, especially for on-line processing, is to load the indices into virtual storage. This reduces physical accesses to the disk and will improve response time. The cost in this case is virtual and processor storage. If these are available, then this is the preferred option. The minimum area allocated should be at least one index buffer per index level in the dataset. Ideally, the entire Index Set should be resident in storage for optimum performance. Index records are brought into storage as required (no read ahead). The number of buffers recommended will be reviewed in the CICS/VS area in greater detail since direct processing usually is more frequent in an on-line environment.

The use of the different index options will depend on the environment. If DASD space is available, then IMBED and REPLICATE can be used. If virtual storage is available, then allocate additional index buffers. If REPLICATE is to be specified, then we suggest that IMBED be specified. With REPLICATE, the price of IMBED is paid but all of the advantages are not achieved. IMBED provides the potential reduction of arm movement between the SSI and the data depending on whether the arm is still in position or has been stolen by another request. This will depend on the disk activity.

In the next issue, the following areas will be examined: CICS/VS and on-line VSAM processing and miscellaneous considerations.

VTAM/SWITCH allows users to switch from VTAM application to VTAM application (CICS, TSO, ICCF, IMS, TESTCICS, etc) by pressing a PF/PA key. Multiple sessions of the same VTAM application are also allowed. Applications can be connected automatically. Security can be specified at the user, application, physical and logical terminal level. Predefined LOGON procedures can be set up for each user or for groups of users.



Multiple VTAM Sessions for each Terminal

DOS - $1,495
MVS - $2,995

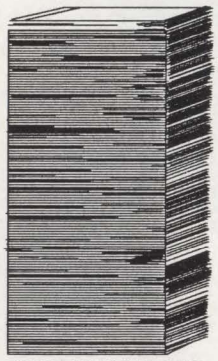
OTHER SOFTWARE

Send MESSAGES to CICS users with CICS/MESSAGE

- Does not affect user's transaction
- Saves screen buffer, Tran-id, and Commarea and displays the message
- Message can be sent to one terminal or Operator (OPID), set of terminals or Operators, all terminals or console
- Messages can be sent using a supplied CICS transaction or by LINKing to the message program

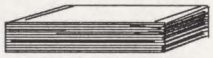
$695 for purchase

ELIMINATE IDCAMS LISTCATS
DOS - $495 OS, MVS - $695



IDCAMS LISTCAT

Recapture wasted disk space, save time and paper by using LISTCAT PLUS. Lists VSAM entries (including ICF VSAM) in one-line-per dataset format. Flags VSAM inefficiencies. Datasets needing attention can be flagged. Lists non-VSAM entries in ICF catalogs, too.



LISTCAT PLUS

Over 30 other products to choose from for CICS and BATCH.
Each available for FREE 30 DAY TRIAL, purchase, or annual lease.

MacKinney Systems (417) 882-8012
2740 South Glenstone, Suite 103 Springfield, Missouri 65804

CIRCLE #3 on Reader Service Card ▲

ABOUT THE AUTHOR

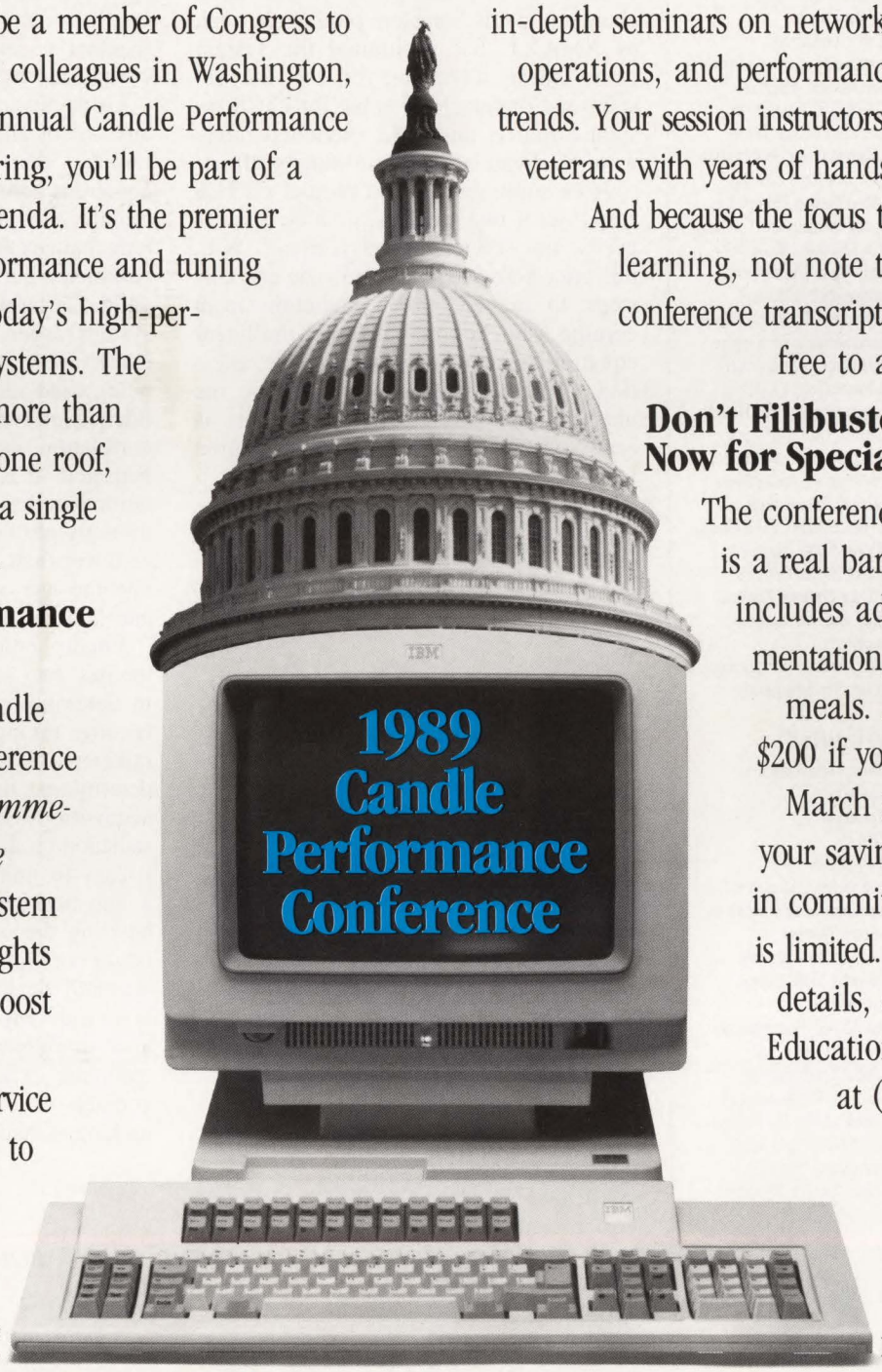
Eugene S. Hudders is president of Multiple Computer Services, Inc. (MCS). In 1983, he became a Certified Data Processor (CDP) and Certified Computer Programmer (CCP). He is also a Certified Systems Professional (CSP). During 15 years at IBM, he was a systems engineer, SE manager, marketing manager and branch manager. He has also been general manager for MAI del Caribe. Multiple Computer Services, Inc., G.P.O. Box 1697, San Juan, PR 00936-1697, (809) 721-7200.

Capitalize On Greater Performance

You don't have to be a member of Congress to meet distinguished colleagues in Washington, D.C. At the third annual Candle Performance Conference this spring, you'll be part of a history-making agenda. It's the premier forum for the performance and tuning experts who run today's high-performance online systems. The conference offers more than 60 sessions under one roof, all for the price of a single conventional class.

Watch Performance Blossom

Only the 1989 Candle Performance Conference offers this much *immediately applicable* information on system performance...insights that can quickly boost your data center's productivity and service levels. In addition to the environment-specific sessions, you'll also benefit from the scores of in-depth seminars on networking, automated operations, and performance management trends. Your session instructors are all industry veterans with years of hands-on experience. And because the focus throughout is on learning, not note taking, detailed conference transcripts are distributed free to all participants.



Don't Filibuster — Enroll Now for Special Discounts!

The conference fee of $1,500 is a real bargain because it includes admission, documentation, and all hosted meals. And you'll save $200 if you register before March 30, so don't let your savings get hung up in committee. Enrollment is limited. For registration details, call the Candle Education hotline today at (213) 442-4075.

!Candle®
Candle Education
1999 Bundy Drive
Los Angeles, CA 90025

April 30 — May 5, 1989

Hyatt Regency Hotel
Bethesda, Maryland

CIRCLE #116 on Reader Service Card ▲

Automated Operations

Taking the industry by storm!

- Automatic MVS & JES2 operator commands.
- Automatic System IPL and System Shutdown.
- Automatic Operator Command Completion Validation.
- Automatic Operator Directives (highlighted messages).
- Automatic Job and/or Command Schedule Generation.
- Automatic Insertion of Execution Dates (Run-Date).
- Automatic Predecessor Validation/Verification.
- Automatic Management of Tape Drive Availability.
- Automatic Management of predetermined Initiator Availability.
- Automatic Interception of JCL-ERRORs and ABENDs.
- Automatic Interception of "NOT CATLG-2" Type Errors.
- Automatic Interception of Maximum Return Code Errors.
- Automatic Detection of Librarian, Panvalet, or PDS Libraries.
- Automatic Prompting for Required JCL or Data Changes.
- Automatic In-stream SYSIN Data Modification Prompting.
- Automatic Job Submission / Command Issuance and Tracking.
- Automatic Job Selection Based Upon Multiple Calendars.
- Automatic Selection Based Optionally Upon Day-of-week.
- Automatic Holiday Adjustment (Optionally by Country).
- Automatic Critical Path Job Stream Support Through Priorities.
- Automatic Job or Operator Command Serialization Support.
- Automatic Data-Entry Requirements Displayed Separately.
- Automatic Interface to any Restart System (e.g. CA-11).
- Automatic Initial Build and Daily Maintenance of Job History.
- Automatic JCL or Data Changes based upon day-of-week.
- Automatic JCL or Data Changes based upon week-of-month.
- Automatic JCL or Data Changes based upon month-of-year.
- Automatic CPU Affinity for multiple CPU installations.
- Automatic Dependent Job Affinity (Better than CPU Affinity).
- Automatic Required Dataset Validation & Job Triggering.
- Automatic Support for Exclusive Use of In-Use Datasets.
- Automatic Support to DUMMY JCL for Optional Datasets.
- Automatic Support to Skip Jobs Conditional on Datasets.
- Automatic ASF Comprehensive Security.
- Automatic SYSOUT Capture/Browse/Archive (coming soon).
- Automatic Incident Reporting/Problem Tracking (soon).

Simplicity Personified!

- No New Concepts, Procedures, Editors, Techniques, etc.
- Full ISPF and ISPF/PDF Implementation.
- ISPF Split-Screen (No Dedicated Terminals).
- Full Context Sensitive Interactive HELP.
- Run Documentation On-Line via ISPF/PDF.
- Uses ISPF/PDF EDIT to Define Scheduling Requirements.
- Schedule Commands or Jobs for Execution on a Future Date.
- Dynamic TSO/ISPF Multi-System Status Display.
- Support for 100's of Duplicate Jobs/Commands per day.
- Interactive, Dynamic Calendar Creation & Maintenance.
- Separate Scheduling and Execution Controls.
- One Single Daily Batch Job Supports All ASF Requirements.
- Dynamic "DEMAND" support for Jobs or Commands.
- Operator Console Interface for Multiple Command Lists.
- Optional External Security (ACF2, RACF, Top Secret, etc).
- Multiple "Independent" Separate Copies of ASF Per Processor.
- Multiple Centralized and Decentralized (End-User) Schedulers.
- Extremely low system overhead (typically less than 2%).
- NO System Hooks, NO mods, NO Exits, NO IPL Required!
- 7-Day 24-Hour Support. --- 60-Day Free Trial.

ASF

"Designed for the '90s"

Chaney Systems Support
W18180 Janesville Rd
Muskego, WI 53150
(414) 679-3908

CSS

Statistical Theory *from page 88*

time turns out to be .030, but CICS posts .5891. This represents a value outside of the predicted range. The increase in CICS response cannot be attributed to file response because the predicted value of .030 was realized. Therefore, the problem must be attributed to another source. Perhaps your CPU is peaking out or maybe a change in a CICS system parameter, such as AMAXT, has prohibited the system from taking advantage of the file response. Conversely, a value for CICS response that is under the predicted range would indicate a successful tuning effort.

Now suppose that you predict a CICS response of two seconds, a value that exceeds your service level agreement. With this prior knowledge at hand, you can take steps to prevent your prediction from coming true. You may also use the linear equation in reverse answering the question, "What file response will give me unacceptable CICS response?" This is accomplished by solving the equation for X:

$$MX + B = Y$$
$$MX = Y - B$$
$$X = \frac{Y - B}{M}$$

Where:

X = File response mean
Y = Hypothetical CICS response = 2 secs.
B = .3931146
M = 25.596342

Using your data and a hypothetical, unacceptable value of two seconds for CICS response you get:

$$X = \frac{2 - (-.3931146)}{25.596342}$$

$$X = \frac{2 + .3931146}{25.596342}$$

$$X = .0934944$$

This tells you that an average file response of .0935 will deliver a mean CICS response of two seconds.

Conclusion

The material that is covered here is only meant to serve as an introduction to statistical theory as it applies to predicting performance. It must be realized that the relationship between file I/O times and CICS response is but a fraction of the factors that must be taken into account. However, by considering the relationships between the six previously mentioned system parameters and using the statistical theory presented here, one can chain together a series of equations that will closely represent your system.

Using this, you can accurately predict any one or all of the six performance parameters that define your system. These equations should be recreated every month. Any alteration made on your system changes the relationships between the parameters, rendering your equations invalid. Certainly not all factors concerning system resources and performance are linear. However, considering the time frames being used relative to the data, the confidence interval being used and the proven correlation, any error is adequately compensated. It is convenient that the calculations are based on data collected over a monthly period of 30 daily observations, as fewer than 30 observations would dictate the use of another set of equations and tables.

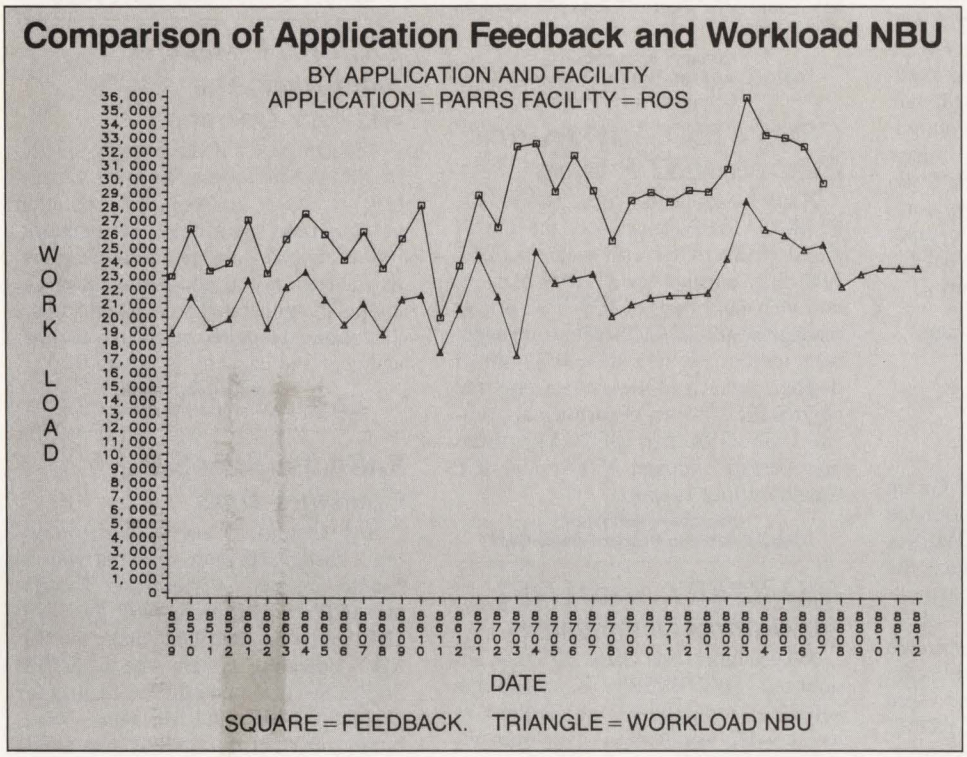
Finally, I have used this technique over the past two years and find it to be useful in determining what part of the system requires tuning (that is, CICS system parameters, disk I/O subsystem), as well as determining the effects of new hardware acquisitions both before and after their installation on the system. Also I have found it easy to implement the calculations on a statistical calculator. This allows, after entering the data, retrieval of the correlation coefficient, the linear equation, the standard deviation and even the predictions with only a few keystrokes. You may also implement the equations on your computer either by writing the programs yourself or using Lotus, SPSS or SAS making statistical analysis even easier.

ABOUT THE AUTHOR

Philip C. Davis, III has been a systems programmer with Delaware Trust Company for the past three years. Prior to that Davis was a systems programmer with Villanova University. Delaware Trust Co., 900 Market Street Mall, Wilmington, DE 19801, (302) 761-1368.

Capacity Planning *from page 57*

FIGURE 1



medical facility and each installation granule (unless the granule is the facility as a whole).

To capture the Actual (versus Forecast) NBU data, we ensure that for each major AF installed, a corporate data source can be found of how many NBUs caused work for that AF. This provides data for updating the capacity planning database and avails feedback by which forecasts are tuned.

For instance, to provide the Actual WNBUs for PARRS, ADT and the three MRMS AFs, the ADT system reports the number of admissions for each medical facility and the medical statistics system reports the number of outpatient visits and X-ray visits for each facility.

Additionally, MRMS periodically reports the number of medical record sign-ins and so on. PARRS reports the number of appointments booked. These real-world feedbacks of the application's workload are the sort of metrics that the NBUs are supposed to track. If NBUs are plotted against real-world feedbacks, the shape (but not necessarily the magnitude) of the two curves should match reasonably well. Figure 1 illustrates this for one of the medical facilities served by PARRS. The triangular data points show the "patient visit" WNBU and the square ones show the "patient appointment" feedback, the work that PARRS actually did. Note that the NBU is forecast into the future but the feedback is not. (The numbers on the Y-axis have been altered for this article.)

The major difference between NBUs and feedbacks such as appointments is that we currently have no reasonable basis for forecasting the feedbacks; we do for the NBUs. After a few years' feedback data have been accumulated, it is entirely possible that this will become susceptible to reliable forecasting. If this comes about, it may be a wise decision to switch over to using this "new" WNBU for capacity planning henceforth.

In summary, the crucial point is that application developers must contribute to the capacity planning process by finding and accounting for business events called NBUs that will relate in a linear fashion to computer resource consumption.

In the next issue, I will describe the capacity planning database in detail, give some examples of its use and relate some hard-earned practical experience.

ABOUT THE AUTHOR

Michael Snyder is senior technician in the Application Development Group at Kaiser Foundation Health Plan, Inc. His background includes 20 years of application development concentrating on analysis, design and coding.

Automated Systems

Nothing Else Like It Anywhere!

- Dynamic Job Class Enforcement by User-id/Jobname.
- Dynamic JOBCAT/STEPCAT Restriction and Authorization.
- Dynamic WTO Message Processing Subsystem.
- Dynamic Console Message Suppression.
- Dynamic INTRDR Restriction and Authorization.
- Dynamic Accounting Number Validation and Verification.
- Dynamic Addition of TIME= to Job Cards by Class.
- Dynamic Validation of TAPE usage by Job Class.
- Dynamic Addition of EXPDT = for TMS (CA-1) environments.
- Dynamic Validation of BLP usage by Program Name.
- Dynamic Operator Commands Triggered by Any Message.
- Dynamic JES2 Remote Start, Stop, & Status Messages.
- Dynamic DASD Allocation Redirect via UNIT = Substitution.
- Dynamic Volume Affinity Enforcement by USERid/GROUP.
- Dynamic DASD Block Allocation Compatibility Support.
- Dynamic CPU & WAIT Time Extension for Production Jobs.
- Dynamic MSGCLASS = Addition for TSO Submitted Jobs.
- Dynamic Job Start/End Notification for TSO Users.
- Dynamic Placement of Step Return Codes on Joblog.
- Dynamic NOT-CATALOGED-2 Interception & Notification.
- Dynamic DASD Device Block Counts by DDNAME.
- Dynamic Addition of Distribution Code on Separator Page.
- Dynamic Adjustment to BUFNO=1 for VIO requests.
- Dynamic TSO User Job Failure Notification.
- Dynamic Job Identification with RACF and TSO/E Systems.
- Dynamic Elimination of Predefined Messages from Hardcopy.
- Dynamic Highlighting of Installation Specified Messages.
- Dynamic TSO Redirect (ECHO messages to TSO User).
- Dynamic Cancellation of Jobs Issuing Msgs (IEF287I).
- Dynamic Reply to WTORs Providing Greater Automation.
- Dynamic CICS Journal Offload and Validation.
- Dynamic SMF Dataset Offload.
- Dynamic Production-Job Failure Notification (Abends, JCL-errors, Return codes (non-rollable highlighted messages).
- Dynamic Authorization Support for Installation Written Programs Validates Caller by Location (LPA/CSA or Region).

Don't Write Another MVS Exit Again!

- Dynamic Changes to DSS Specifications (no IPL).
- Dynamic Loading of SMF, DFP, & TSO Exits (No IPL).
- Dynamic Interception of Abends in System Exits (No Job or System Failure if/when an Exit Abends).
- Dynamic Support for Multiple Concurrent Exit Modules (2 IGGPRE00s, 5 IEFUJIs, 3 IEFACTRTs, etc).

Specialized Support Utilities Included!

- IPL Volume Space Compression Eliminates SMP X37 Abends.
- Real-Time Update of DSCBs (Allows Rename Even if In-use).
- JCL Reformat/Validation Utility which Supports User Exits.
- Dynamic DASD Space Reclamation and Reporting Utility Scratches Work Datasets, Compresses PDSs, Releases Unused Space, Migrates non-current GDG datasets (if DFHSM is installed), and Reports on Uncataloged/Miscataloged Datasets.
- MVS Support for VM/CMS File Structures Simplify Conversions.
- Convert multiple CMS Files to MVS PDS members by File-type.
- 7-Day 24-Hour Support. --- 30-Day Free Trial.

DSS

"The Dynamic Support Subsystem"

Chaney Systems Support
W18180 Janesville Rd
Muskego, WI 53150
(414) 679-3908

DSS

CIRCLE #140 on Reader Service Card ▲

PRODUCT UPDATE

CheckOut/VM Provides Higher System Availability

Duquesne Systems (Pittsburgh, PA) recently introduced CheckOut/VM, a software product that verifies the availability of both hardware and software components in a VM system configuration. It is designed to provide higher system availability by automatically probing the VM system for down components, restarting the down components it finds and notifying designated support staff of component failure.

For more information
CIRCLE #200 on Reader Service Card

ADMS Reduces Time To Construct Data Management Requests

Advanced Software Products Group (Naples, FL) has announced Automated Data Management System (ADMS), a system said to drastically reduce the amount of time necessary for constructing data management requests. All requests are transmitted on-line through TSO/ISPF screens. It also provides the ability to transmit, print, confirm and view historical request data. ADMS supports the MVS operating system.

For more information
CIRCLE #201 on Reader Service Card

Personal REXX 2.0 Available

Mansfield Software Group (Storrs, CT) has announced Personal REXX 2.0, a major upgrade to their REXX language processor for the IBM PC, PS/2 and compatibles. Personal REXX is the first full PC implementation of VM's acclaimed REXX programming language. Major enhancements in 2.0 include improved performance with some REXX programs running twice as fast; and increased storage capacity so that Personal REXX can use all available memory below 640K as well as EMS expanded memory.

For more information
CIRCLE #202 on Reader Service Card

Best/1 I/O Diagrammers Pinpoint System Bottlenecks

BGS Systems (Waltham, MA) has announced the BEST/1 I/O Diagrammer for MVS and for VM. These two self-contained products are designed for systems and operations personnel responsible for managing and tuning DASD I/O subsystems. With these standalone I/O Diagrammers, mainframe installations can quickly and easily find out what is going on in their DASD systems. They can draw their I/O configurations and identify which are the "hot" or high utilization devices representing bottlenecks that are prime candidates for tuning. Standard MVS and VM data sources are used as input.

For more information
CIRCLE #203 on Reader Service Card

CPMS/SYSD Accesses And Controls JES2 Printers

H&W Computer Systems' (Boise, ID) CPMS/SYSD Release 6.1, the CICS-based productivity tool, features ISPF/PDF-like edit/browse, SDSF-like report viewing/routing, CICS/JES2 printer management and CICS debug/management utilities. With the new JES2 printer display, authorized users can access and control JES2-driven departmental printers. LISTVTOC and LISTCAT utilities now feature indexed VTOC and ICF Catalog utility support.

For more information
CIRCLE #204 on Reader Service Card

DB2-XPERT Assists DB2 Application Development

XA Systems (Los Gatos, CA) has announced DB2-XPERT, a mainframe software productivity tool designed to assist DB2 application development. DB2-XPERT provides capabilities to edit, browse, extract and load DB2 tables under TSO/ISPF. It runs as a dialog under ISPF using menu-driven, ISPF-like displays and a command set very similar to ISPF.

For more information
CIRCLE #205 on Reader Service Card

VSAMVIEW Selects Optimal Parameters

Design Strategy's (New York, NY) VSAMVIEW uses catalog and user-supplied information to select optimal parameters for a new or existing VSAM cluster. Clusters can be tuned for improved performance or space utilization. Menu screens are similar to those within ISPF. New features in Version 3 include: list volumes containing VSAM files and/or suballocated space, list user catalog aliases, and list VSAM clusters on a specified volume.

For more information
CIRCLE #206 on Reader Service Card

VTAM-EXPRESS Eliminates Separate Compression Product

SofTouch Systems (Oklahoma City, OK) recently announced VTAM-EXPRESS that performs terminal data-stream compression at the VTAM level. No longer is a separate compression product for each different VTAM application needed. Both inbound and outbound compression is provided for every application in the network.

For more information
CIRCLE #207 on Reader Service Card

EXPLICIT Analyzes Mainframe And Network Operation

Technetronic's (McLean, VA) EXPLICIT is said to provide an easy, painless method of analyzing the operation of your IBM mainframe and network. Its automated data gathering features, its modern mouse and menu interface, and its PC-generated color graphs make it a unique integrated performance management and capacity planning tool.

For more information
CIRCLE #208 on Reader Service Card

SAS/CPE Start Set Combines Tools

SAS Institute (Cary, NC) is providing a faster and more efficient way to handle Computer Performance Evaluation (CPE) applications with the SAS/CPE Starter Set. Available under the OS/MVS operating system, the SAS/CPE Starter Set is a menu-driven facility for evaluating SMF and RMF data — combining many of the SAS system's data management, analysis and reporting tools into a packaged CPE solution.

For more information
CIRCLE #209 on the Reader Service Card

DB2/Batch Cures Batch DB2 Blues

Relational Architects' (New York, NY) DB2/Batch is said to cure many of the problems caused by abends in DB2 batch programs masked by the TSO Terminal Monitor Program and DSN command. DB2/Batch solves these problems by letting you run your DB2 programs directly in batch with conventional JCL. With the product, batch DB2 programs can reportedly work compatibly with job scheduling packages and multi-step jobstreams, as well as with Checkpoint Restart and the abnormal disposition of datasets.

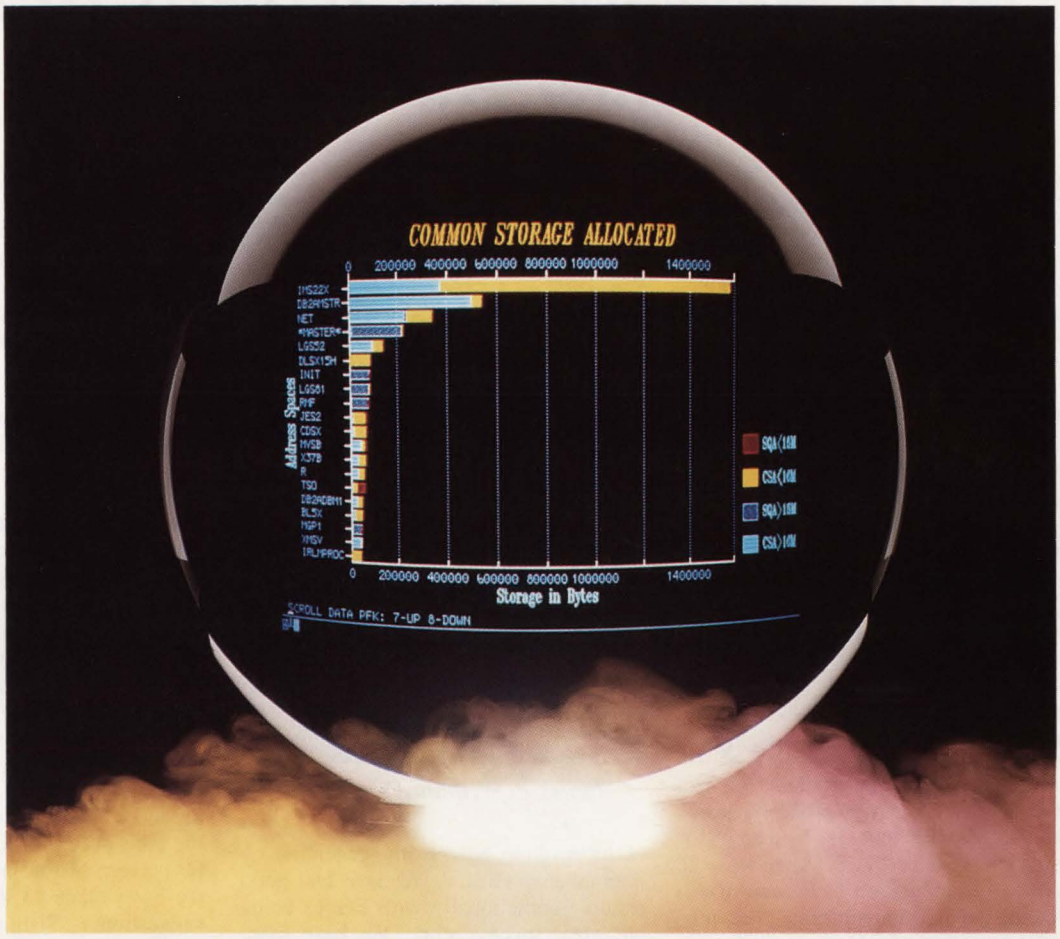
For more information
CIRCLE #210 on the Reader Service Card

MVS Users Get LOADLIB AUDIT

D. L. Brickey & Associates (Rochester, IL) recently announced LOADLIB AUDIT for MVS. It produces reports from detail analysis and content cross referencing of an executable program library. With LOADLIB AUDIT, the detail content and structures of a library are exposed and processed, producing very enlightening information.

For more information
CIRCLE #211 on the Reader Service Card

INTRODUCING THE FIRST REAL-TIME MONITOR FOR CSA.



Introducing the new Common Storage Monitor for RESOLVE PLUS, the first in-depth tool for monitoring users of CSA and SQA on MVS/XA. And the most effective way to reduce time consuming, and costly, IPLs caused by common storage constraints.

Control CSA creep.
Common Storage Monitor helps you control "CSA creep", the slow, hidden increase in common storage allocation that can ultimately result in system degradation and even failure. It is the only monitor that allows you to account for common storage usage by individual user. So you can identify applications that are abusing common storage and recover wasted CSA held by terminated tasks.

Pinpoint CSA usage.
Common Storage Monitor displays allocated storage for each address space by job name. Operating in either ISPF or command mode, you can display as much detail as you need to identify the user responsible for the allocation, and to analyze how the storage is being used. Password-protected RESOLVE PLUS services are also provided to free areas of CSA once they have been identified. Online color charts give you an easy-to-interpret overview of common storage allocation.

For more information on the Common Storage Monitor for RESOLVE PLUS Version 3.0.0, call Marty Johnson today. In California: 800-624-5566. Outside California: 800-822-6653. Boole & Babbage, Inc. 510 Oakmead Parkway, Sunnyvale, California 94086.

Boole & Babbage

International sales and support provided through The European Software Company and a worldwide distribution network.

NOW AVAILABLE IN A STANDALONE VERSION!

CIRCLE #160 on Reader Service Card ▲

VENDOR PROFILE

Altai Software: The Experts In Data Center Automation

Designing software that enables a computer system to run itself is the objective of Altai Software, an Arlington, TX based software company specializing in data center automation products for the IBM mainframe environment.

Altai's two main products — Zeke, "The Scheduler that Works," and Zack, "The Operator's Operator," provide data center operations personnel with the tools necessary to automatically schedule and operate IBM's VSE and MVS operating systems.

"Teaching the computer to run itself is our mission," says James P. Williams, president of Altai. "It is just not practical to expect people to keep up with these giant beasts and effectively schedule, operate and manage them."

Broad Scope of Automation

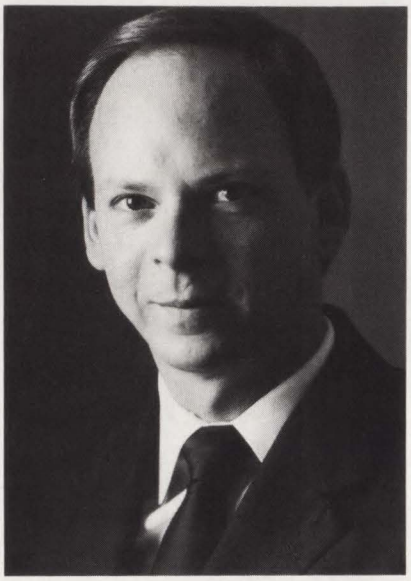
Altai's view of the problems of data center automation may be broader in scope than that of other vendors. Instead of focusing on VSE or MVS, Altai sees the problem of automation to be one of automating the IBM mainframe data processing environment. Altai's solutions for automated operations are applicable to both VSE and MVS, just implemented slightly differently.

"Conceptually, both operating systems have the same problems that relate to automation. In fact, many companies have a mix of installations running VSE and MVS and are therefore looking for a product that provides a solution for both operating systems — one that solves the problems of automating data processing, not just MVS or VSE," Williams points out.

First Product — Zeke

Altai's first product, Zeke, was first installed in June of 1982 at Henkel Corp. (now ITG Corp.) in Minneapolis, MN. "Our first two installations in June of 1982 were both converting from NCR machines to IBM mainframes. They were used to having a scheduling system and did not want to implement their IBM systems without one. We installed Zeke in each of those accounts and they are still running Zeke today — although both have since migrated from VSE to MVS," Williams says.

Altai began distributing products internationally in 1983 with about 34 percent of its revenue now being generated by international sales. "We have had good results internationally with France being our great success story. We have more than 100 Zeke and more than 50 Zack installations in France. This has happened for two reasons — the quality of our products and the caliber of our French distributor, Faster SARL," Williams explains.



James P. Williams, president

In 1987, Datapro Research Corp. conducted a user survey as part of its evaluation of Zeke. That survey rated Zeke higher than any other scheduler on the market in terms of customer satisfaction. Today, about 600 installations worldwide use Zeke to automatically schedule their computer operations.

Zack Gains Quick Acceptance

Altai's second product, Zack, is relatively new, having been generally released in January of 1988. Zack automates console operations for both the VSE and MVS operating systems. Zack allows you to program console operation using a REXX-like language. Zack can automatically respond to and suppress routine console messages. In addition, Zack can issue system commands or messages at predetermined times, call user-written programs and perform many routine functions faster and more accurately than ever before possible.

"This frees operations personnel for handling the exceptions to the rules. As console message rates continue to increase, it will become more and more difficult for human operators to keep up. Zack can be a valuable aid," Williams points out.

Altai has sold more than 120 Zack systems since the product's introduction. "The quick acceptance of Zack in the market surprised us. However, Zeke and Zack together make a powerful team and a large percentage of our Zack sales have been to existing Zeke clients. This demonstrates the degree of loyalty Altai's customers feel toward the company. We listen to them, we react to their needs and we support them. That is why we are still here," Williams comments.

Altai users meet regionally four times a year to share their experiences and to inform Altai of their needs. At each meeting, the users vote on suggested product enhancements, thereby prioritizing their needs for the company. "About 80 percent of the product enhancements we have made to Zeke have been user suggestions," Williams says.

Upcoming Products and Enhancements

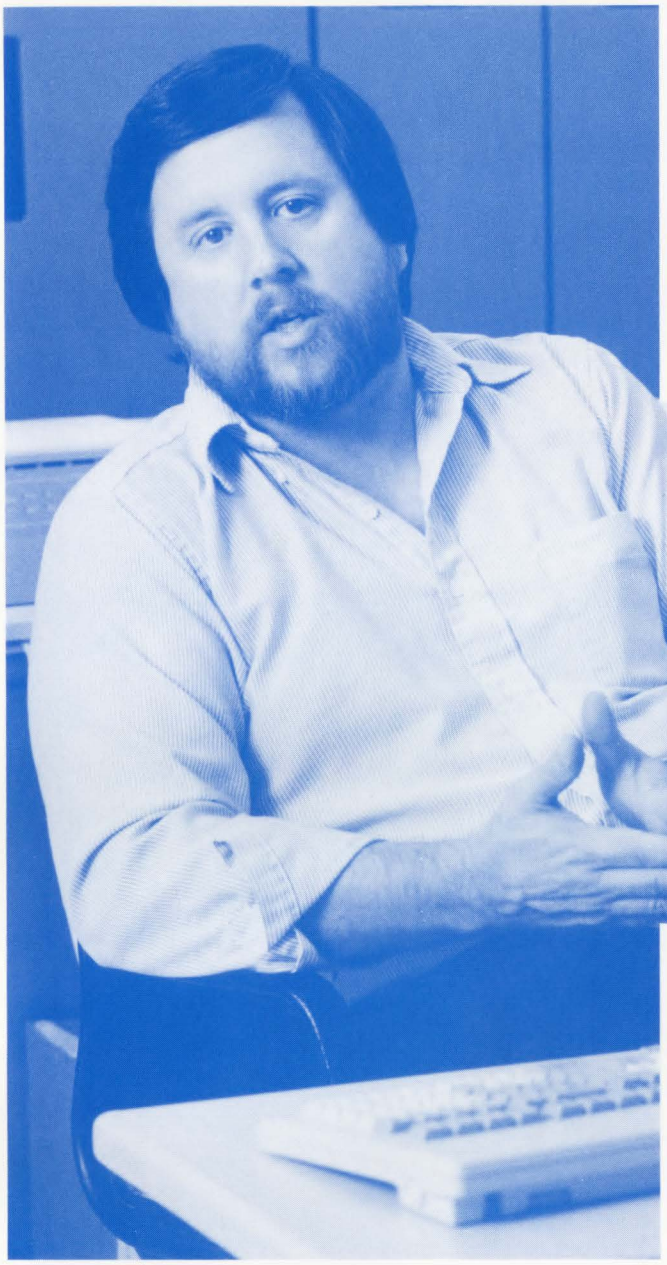
Altai has several new products and major enhancements to existing products planned and in development. Upcoming versions of Zeke and Zack will provide scheduling and management for a network of mainframes, including mixed MVS/VSE environments. Zeke/VM is scheduled for release during mid-1989.

Altai's newest product, Z/CAT2, is a utility that prevents NOT CATLG 2 and duplicate dataset errors from occurring under MVS. Z/CAT2 is currently in Beta test sites and will be available for general distribution on March 1. Altai is also developing a restart/rerun management system that will enter the early testing stages in March. Additional products are planned to further expand Altai's family of data center automation software.

Key to Success

Williams says that a key to Altai's success has been its concentration on a single area of expertise: data center automation. "We aren't trying to be the biggest, just the best at what we do," concludes Williams.

Vendor Profile is a regular forum whereby a vendor is given the opportunity to introduce the company and its products to MAINFRAME JOURNAL readers.



Lee Veal On VSE Console Support...

"Anything IBM gives you, DOCS makes five to ten-fold better."

DOCS™

With 17 years systems software experience, Lee Veal, Software Support Supervisor for The City Of Garland, Texas, has seen a lot of software come and go. But there's one product Lee won't let go—DOCS (Display Operator Console Support) from SMARTECH Systems. And here's why...

I recommended DOCS because it looked like it was going to save us a lot of money. With the ability to call up the VSE system console from multiple terminals, we're able to de-centralize our access. And that cuts down on the phone calls received by the operational staff, because programmers have access to DOCS in their building ...and at home.

How easy it is to install DOCS...

It's not like re-inventing the wheel to bring DOCS up. You don't have to change your operational layout or procedures just because DOCS is in place. In fact, it actually makes the system come up faster.

Plus DOCS allows the terminals to be shared back and forth between CICS or any other on-line software. They can be dual purpose—and we need those terminals for other functions.

On DOCS security features...

DOCS allows us to use password securities and "read only" type securities whenever the DOCS consoles come up. We can even include or exclude partitions on each individual console. We use these features in the programmer area—for the terminals that the programmers have access to.

How DOCS Dynamic System Status Display (DSSD) increases efficiency...

It tells us if a job is running or why it stopped. This makes a difference when we have a problem. We can know what the program is waiting on and address the problem saving us a lot of time. If we didn't have this, we'd be "flying blind."

How DOCS reduces keying time...

With DOCS we're able to set up our PF-Keys on-line to store commonly used responses like IGNORE, CANCEL and RETRY. Also, there are features that let us "pre-answer" questions. One we use more than any is the "call back" feature where you can call back your previous reply and answer that same thing again for the next response. Plus, we have the ability to recall any line on the screen for input. It saves the operators time from sitting and pecking out those characters.

On the need for DOCS in today's environment...

It simply makes your operation run easier with it than without it. Most of the benefits of DOCS you really can't say in words how good they are—you just have to experience it.

Why not experience it for yourself? Try DOCS FREE for 30 days. It takes just 30 minutes to install. Call us for more information.

1-800-53-SMART

Outside the U.S., call (214) 956-8324

SMARTECH SYSTEMS, INC.

Turning high technology into *SMART TECH*nology.℠

10015 W. Technology Blvd., Dallas, TX 75220
Telex: 9102503110

VSE and IBM are registered trademarks of the International Business Machine Corp. SMARTECH and DOCS are trademarks of SMARTECH Systems Inc. Copyright © 1988 SMARTECH Systems, Inc. All rights reserved.

CIRCLE #65 on Reader Service Card ▲

VSAM DATA COMPRESSION

Without the CPU Overhead

IAM REDUCES THE SIZE OF YOUR VSAM FILES BY 30 TO 70%

IAM'S FILE STRUCTURE — SAVES 20 TO 40%
IAM uses an advanced file structure which is far superior to VSAM. IAM's supercompressed index, freespace concepts and blocksizes make much more efficient use of disk space.

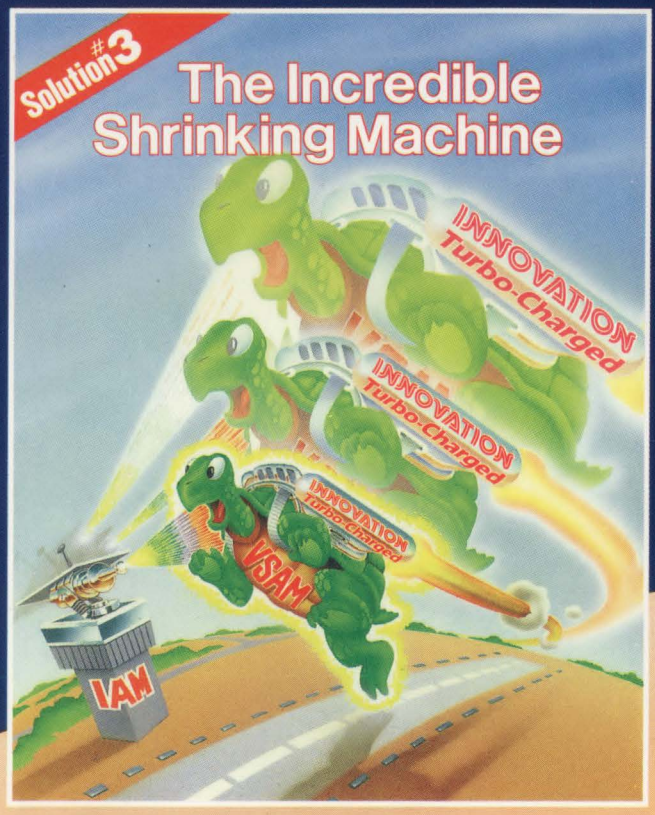
IAM'S DATA COMPRESSION
SAVES AN ADDITIONAL 20 to 50% DASD SPACE
Most files contain records with unused fields or repeating sets of characters. When IAM applies its proprietary compression techniques, the result is an additional 20 to 50% reduction in file size.

IAM's CPU time is dramatically less than competing compression products. In fact, since IAM's CPU time is normally much less than VSAM, IAM with data compression takes less CPU time than normal VSAM processing.

TRANSPARENT
Online systems (CICS), BATCH jobs, TSO, SMP/E and other applications make extensive use of key indexed VSAM (KSDS) files.

IAM is a transparent alternative to VSAM KSDS files, which substantially reduces the impact of VSAM processing in your installation. There are no modifications to programs or JCL to use IAM files in place of VSAM.

AUTOMATIC RELEASE OF UNUSED SPACE
IAM takes the guessing game out of VSAM space allocation. Large amounts of disk space are wasted when users overestimate how much space VSAM requires or how many records a file will contain. VSAM cannot release overallocated space.



FREE VSAM SPACE SAVINGS ANALYSIS*

SPACE SAVINGS

DATA SET NAME	VSAM TRKS ALLOC	VSAM TRKS USED	IAM TRKS STD	IAM TRKS COMPR	% SAVINGS STD	% SAVINGS COMPR	TOTAL RECORDS	AVERAGE LARGEST	MAX LRECL
BIG.CLUSTER	37155	37155	27855	15600	25	58	4754670	233 502	580
CICS.FILE.MASTER. TABLE.CLUSTER	21000	19005	12720	9495	33	50	5068165	150 150	150
NAME.ADDRESS.FILE	9315	8985	6465	1875	28	79	428529	680 2090	2090

The VSAM simulation report displays the current size and data characteristics of your VSAM files and how much space IAM will save you with and without data compression.

To see your VSAM usage, send for INNOVATION's free VSAM reporting programs.

Call for a Free No Obligation 90 Day Trial

Supports MVS, MVS/XA and MVS/ESA

Makers of FDR and ABR

INNOVATION DATA PROCESSING

275 Paterson Avenue, Little Falls, NJ 07424 • (201) 890-7300