June 1992

FOCUS

The Magazine of the North American Data General Users Group

In Focus

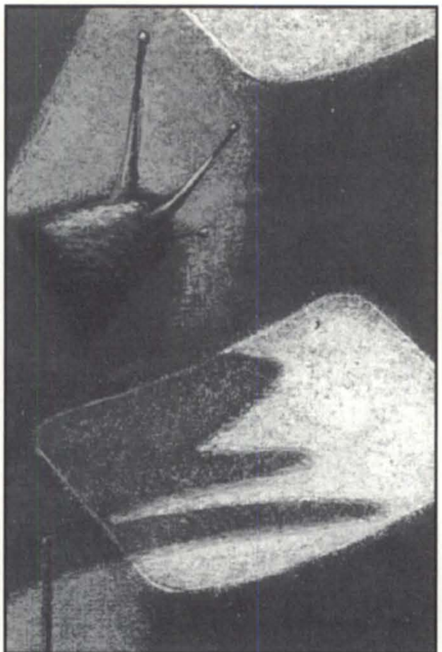
CPU idle time
Well-fed and humming along
Whetstones, Dhrystones, and MFLOPS

Plus

Happy tuning
Yes, *sar*
Real-world report writing
The new MVs

FOCUS

The Magazine of the North American Data General Users Group



Cover illustration by Tom Curry

FOCUS, the Magazine of the North American Data General Users Group (ISSN 0883-8194) is the official publication of the North American Data General Users Group (NADGUG) in cooperation with Turnkey Publishing. Editorial and business offices are located at Livingston Building, Suite 250, 3420 Executive Center Dr., Austin, TX 78731, phone 512/345-5316. NADGUG headquarters are located at NADGUG, c/o Danieli & O'Keefe Associates, Inc., Chiswick Park, 490 Boston Post Rd., Sudbury, MA 01776, phone 508/443-3330.

Postmaster: Send address changes to Subscription Department, Turnkey Publishing, Livingston Building, Suite 250, 3420 Executive Center Dr., Austin, TX 78731.

FOCUS, the Magazine of the North American Data General Users Group is distributed to members of the North American Data General Users Group. Membership fees are $60 per person. A one-year (12 issues) subscription to *FOCUS, the Magazine of the North American Data General Users Group*, costs $48. For memberships and subscriptions outside the U.S., add $50 to defray the cost of mailing.

The cost of single copies is $4. Requests to replace missing issues free of charge are honored only up to six months after date of issue. Send request to *FOCUS, the Magazine of the North American Data General Users Group*, c/o Turnkey Publishing.

Volume 8, Issue 6.

NORTH AMERICAN DATA GENERAL USERS GROUP

NADGUG busy-ness

by Dennis Doyle
NADGUG President

Greetings from Kansas City!

The spring board meeting was held here April 4 at the NADGUG flagship hotel, the Allis Plaza. The meeting and the executive board session immediately preceding it capped a busy six-week period for me as your representative in a variety of situations.

The intense six weeks of NADGUG business began in late February with a working review session at our management firm's offices in Sudbury, Massachusetts. Issues relating to both the spring board meeting and ongoing business were covered by Danieli & O'Keefe staff. Of particular importance was the definition and costing of the proposed projects in the area of membership growth and conference attendance, and their impact on the new fiscal year budget. I also met in Westboro with Data General support staff and Steve Baxter, vice president of corporate marketing. Past NADGUG president Frank Perry joined us, and we carefully revisited plans for the coming year.

In early March, Frank and I found ourselves traveling on the "wrong" side of the road on the way to Birmingham, England, to attend the United Kingdom DG user group's annual conference. The two-day event was provocative as well as educational. Many thanks to retiring Chairman Nigel Ockenden for his gracious hospitality. We look forward to introducing his successor, Jon Guthrie of Scotland, to many of you at NADGUG 92. It was also a great pleasure to meet Olga Kennedy, the outgoing chair of the Irish user group, as well as David Judge, the incoming chair.

We also met with Jean Mouleyre, general director of DG France and International Distributors. We discussed ways to promote NADGUG throughout France and in countries that his staff calls on. Ways to encourage European attendance at the Kansas City conference were also suggested.

Shortly after returning home, it was on to Kansas City for the spring board meeting. Attending were many familiar contributors, committee chairs, a past president or two, and some new folks. For copies of the proceeding's transcript, please contact Tim Boyer, NADGUG's recording secretary. Highlights of reports centered around plans to increase membership, the upcoming KC conference, current financials, and the budget. We will try to present details of these topics in *Focus* as space and time permit.

NADGUG's financial position is on solid ground. We anticipate that it will only get better as we move through the next fiscal year. Implementation of plans to grow the membership base will begin in early summer, and will stay well within budgetary guidelines.

Part of my report to those in attendance in Kansas City was an update on the request for enhancement (RFE) review process. It has begun in earnest, and my thanks go to Linda Klatt, Tim Boyer, Ed Lindberg, David Novy, and to members of their special interest groups for helping to get it off the ground.

Please join us at SIG meetings in Kansas City to participate in rating RFEs, and helping draft new ones.

Kansas City, here we come . . . . Δ

FOCUS

The Magazine of the North American Data General Users Group

NADGUG LEADERSHIP

President
Dennis Doyle

Vice President/ Conference Committee
Jan Grossman

Treasurer
Steve Pounds

Recording Secretary
Tim Boyer

Audit Committee
Calvin Durden

Membership Committee
Chris Thorpe

Planning Committee
Frank Perry

Publications Committee
Maggie Morris

RIG/SIG Committee
Bart Bates

FOCUS MAGAZINE

Publisher
Greg Farman, Ph.D.

Editor
Doug Johnson

Contributing Editors
Tim Boyer, Brian Johnson, Doug Kaye, Kim Medlin, David Novy

Contributors
Dennis Doyle, Rick Havourd, Edward Lindberg, Edward A. Sepich III, Thomas E. Soukup, Don Thomas

Advertising Manager
Michelle Sentenne

Art Director
Pat McMurray

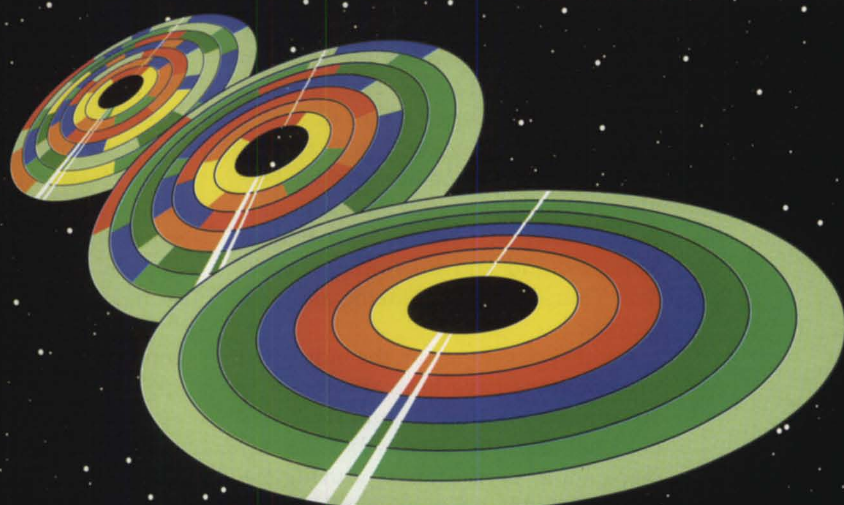
Production Artist
Casey Hunter

Office Manager
Paula Minton

Administrative Assistant
Tammy Agnew

To Boldly Go Where No One Has Gone Before

DISK_PAK™ OnLine!™



Now You Can Optimize Any Disk With All Systems GO!
No Downtime Required.

EAGLE
SOFTWARE, INC.

Call Today For Details 1-800-477-5432

P.O.Box 16 \ Salina, Kansas 67402-0016 \ Phone: (913) 823-7257 \ FAX (913) 823-6185

Lines of communication

by Ed Lindberg
AOS/VS SIG

SYNOPSIS
Get in on the STR and RFE process for AOS/VS and AOS/VS II. Your ideas will make a difference.

I would like to update any interested readers on the activities of the AOS/VS Special Interest Group, as well as provide any other general information that may assist users of Data General products.

The first thing that comes to mind is my recent trip to Kansas City for the NADGUG Spring Board Meeting. We had a very productive session, and I am looking forward to the annual NADGUG Conference in the fall.

As I am sure you will hear from a variety of sources, the meeting will be a little different this year. We will start the sessions on Tuesday, rather than Monday, and end on Friday, rather than Thursday. Start thinking ahead to October.

Another development: the individual presentations have been shortened to 50 minutes, with a 10-minute break between sessions. This will allow for many more presentations during the three-day period. Consider having more than one person from your company attend the meeting, so your business can take better advantage of the program.

Unfortunately, it was still necessary to schedule the Operating Systems Roundtable last in the program. Please plan to attend this session. Although it's the final NADGUG 92 session, it has been moved to earlier in the afternoon, so you will still be able to catch a plane for home if you need to travel on Friday.

We will want to discuss the new Software Trouble Report (STR) and Request for Enhancement (RFE) process at our Special Interest Group meeting. Currently, the SIGs are participating in the RFE process. I want to update you on our progress to this point in time. I have already received two packets of STRs, which earlier had been identified as RFEs to AOS/VS or AOS/VS II by the software developers.

The idea is to provide information regarding the importance of these requests to the development group. The presumption has been made that we might be able to influence the priorities of the developers if we detect an important request that they feel is not significant. Four of us have rated approximately 30 AOS/VS or AOS/VS II STRs that were originally classified as RFEs. Some of these we feel are "bugs," and as such should not be classified as enhancements. If it can be established that they are "bugs," they will be dropped from this procedure and corrected.

We rated the STRs on a scale of 1 to 5: 1 being "important to a lot of members of the SIG;" 5 being a "waste of time." These have been returned to the developers, and now we are waiting for their responses to our input.

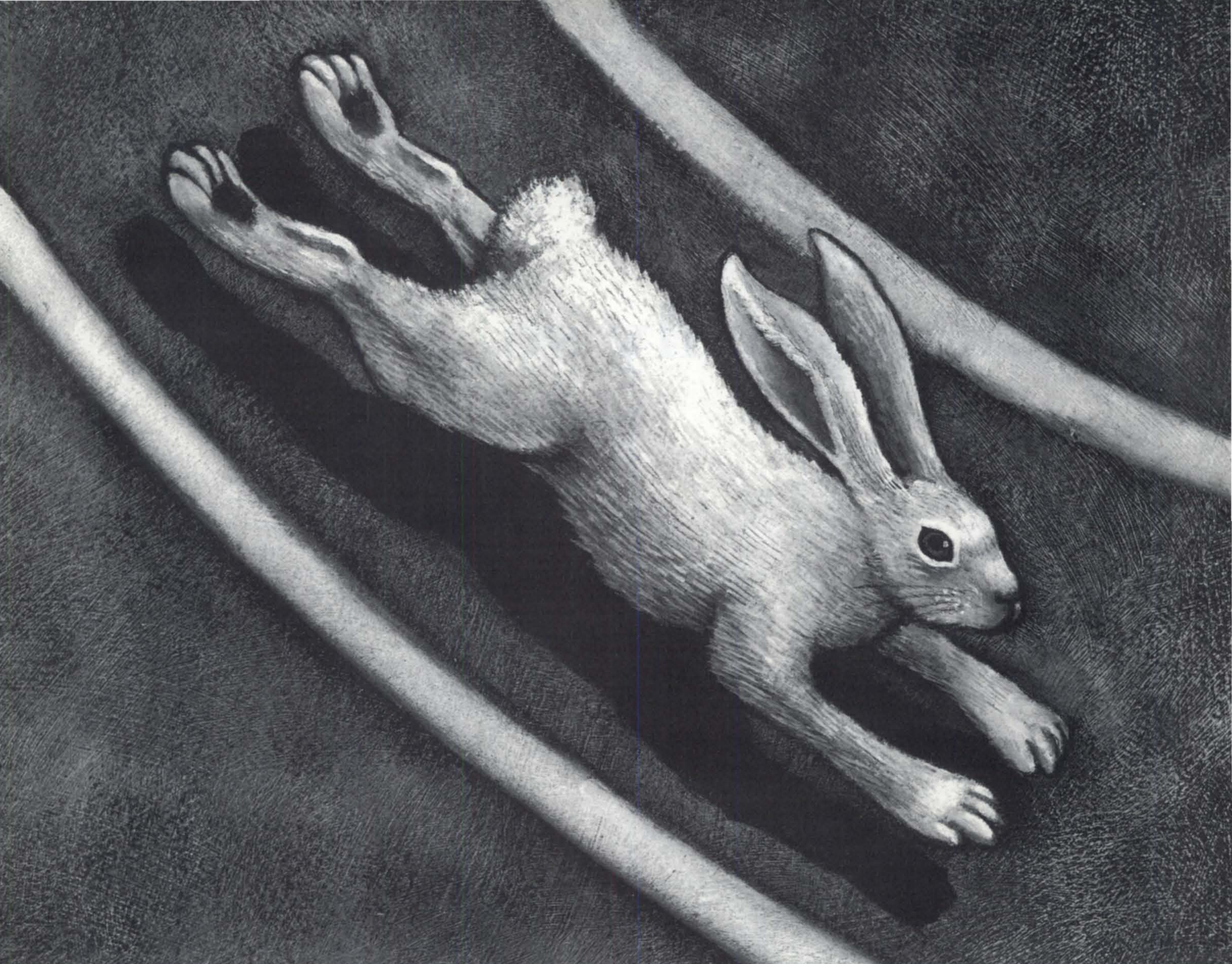
By the time you read this, we will have classified many more STRs, and we hope we will have established a good line of communication and an appeals process if we can't reach agreement with the developers.

I would like to comment on this concept and its importance. First, many of us have stopped submitting STRs against Data General software products. Now is the time to review our actions. This could be an opportunity to correct problems or get enhancements that we as users have needed for years.

Here at Western New England College, we have an MV/10000 and an MV/20000, as well as an Aviion. We are "out of revision" on AOS/VS, Cobol, and Infos II. I have not seriously considered running AOS/VS II, although we're licensed to run it on our MV/20000. With renewed lines of communications and the influence of your interest groups, we may be able to enhance your operations without increasing your costs.

How about participating? Δ

Ed Lindberg is chairperson of the AOS/VS SIG. He may be reached at 1215 Wilbraham Rd., Springfield, MA 01119; 413/782-1246.



Put Your ICOBOL Applications On The Fast Track.

With Choice!™ and Axis™ from Wild Hare. Only Choice!, Wild Hare's run-time system, lets you run your ICOBOL object programs and data files totally unmodified on any popular computer system. Without recompilation, translation or even reorganization. Any object file works on any platform. In any new market. With virtually unlimited sales potential.

And Axis, the world's fastest ICOBOL compiler, keeps you ahead of the pack by letting you develop your applications on almost any hardware, and under any popular operating system. Which means if you would rather work under UNIX than AOS/VS, or MS-DOS than VAX/VMS, no problem. Just order the specific Axis for your favorite development environment and you're off.

Choice! and Axis currently run on:

MS-DOS	**VAX/VMS**
OS/2	**AOS/VS**
PC NETWORKS	**MACINTOSH**
UNIX	**AViiON**
XENIX	**MCS**
AIX	**...and others**

Put Axis and Choice! on your ICOBOL application team and discover what thousands of others already know: When it comes to turning your application into a runaway success story, place your bets on the Hare.

Call Wild Hare at (303) 442-0324 for more information.

wild hare
COMPUTER SYSTEMS INC.

Wild Hare Computer Systems, Inc. 2820 Wilderness Place Boulder, Colorado 80301 USA Fax: (303) 440-7916 Tel: (303) 442-0324

Circle 40 on reader service card.

It ain't necessarily so

SYNOPSIS

Idle time on your CPU always means the operating system has nothing to do. That can be good or bad. You must be the judge.

by Don Thomas
Special to Focus

If you are reading this article, then chances are good that you have or had responsibility for an MV family CPU. If you have responsibility for a system, then you are concerned about getting the best performance from that system. The question is: what is best performance?

There is no black-and-white explanation about system performance. There are many performance gurus with as many ideas about what will increase your system throughput.

Throughput is what we are talking about here. I've always said that you will never get MV/40000 performance from your MV/9300. You *can* get MV/9300 performance from your MV/40000, if you aren't careful. The real objective is to get everything you are entitled to from the CPU you currently have.

The major goal for most system administrators is to increase system idle time. After all, the more idle time the better, right? It ain't necessarily so. Idle users may be a better gauge of system performance than idle CPU time.

System users are the ultimate system performance monitor. They are why the system exists. They are why you, as system administrator, have a job. The reason system users are not generally selected as a monitor of choice is because the information is difficult to formulate into charts and graphs.

Users will reschedule jobs outside peak periods to avoid a poorly performing system. This plays havoc with system idle time. You will see unexplained peaks in idle time when the users are rescheduling. It's wild!

Idle time means the system has nothing to do. The system is never *really* idle. Even when it's idle, it's check-summing memory, just waiting for an event to occur. This means that idle time is excess CPU time. Why do you want your system to waste time? No idle time means your system is performing work all the time, and that's good, right? It ain't necessarily so.

No idle time and poor system response may indicate the need for more hardware resources. While plenty (10 to 40 percent) of idle time and poor performance may indicate the need for more hardware resources. It sounds the same, but the solutions can be completely different.

More power

The first case—poor performance and no idle time—may indicate the need for a more powerful CPU. I would look at the percentage of system time and user time. A higher percentage of system time may be a symptom of an application bottleneck. Adding a more powerful CPU may solve the problem temporarily. Identifying the application bottleneck and solving that problem is the best long-term strategy.

Here is a scenario I have seen again

Setting New Performance Records with Disk, Tape, and Optical

Zetaco's newest peripheral controller, Model SCZ-6, implements fast SCSI-2 technology to support the latest and fastest SCSI disk, tape, and optical drives.

With its 10 MB/sec data transfer rate, SCZ-6 will help you set new MV system performance records in the key events of your daily data storage race—such as speedy transfers of large data blocks, less bus contention in multiple-drive subsystems, decreased backup times, and overall faster response time at the terminals. SCZ-6 is designed for highest performance in multi-drive subsystems.

Go for the gold—the SCZ-6 Multifunction Controller—and discover the speed and endurance you need to win the data storage race. Call us toll-free: **1-800-423-3020.**

Call the Authorized Zetaco Distributor nearest you for complete subsystems and service:

Eastern U.S.	*Western & Northern U.S.*	*Canada*
Design Data Systems	BLM Systems	Lasbert TRI-COM, Ltd.
Suite 200, 40 W. Gude Drive	Suite C, 145 Webster Street	3325 River Road, Unit 5
Rockville MD 20850	Hanover MA 02339	Gloucester, Ontario K1G 3N3
phone (301) 424-7870	phone (617) 982-9664	phone (613) 521-5152

ZETA

Zetaco, Inc.
Subsidiary of Carlisle Corporation
11400 Rupp Drive
Burnsville MN 55337
612-890-5135, FAX 612-890-0791

Circle 41 on reader service card

and again. A client has an application that needs to be rewritten, but the only solution ever proposed has been to buy a bigger and faster CPU. The client has the top-of-the-line CPU and the system performance is starting to deteriorate. What's to be done? They can't move up to a more powerful CPU because it's not available yet.

In comes another hardware vendor with facts, figures, studies, and testimonials about why their hardware outperforms the present gear. The client is sold, but has to convert the application to the new platform.

As promised, the application runs faster than ever on the new hardware. Yep, this was a good move, although an expensive one. Right? It ain't necessarily so.

It's just as likely that the same results could have been achieved by doing the application rewrite on the original hardware platform. The client would have saved the cost of the hardware, retraining, and confusion associated with changing hardware vendors.

Lots of idle time and poor performance means there is a resource bottleneck somewhere. It could be either hardware or software. Gee, I really narrowed that right down!

Disks are a good example of this type of bottleneck. I know you have perfectly balanced disk utilization, and that none of your disks are more than 65 percent full. Congratulations. You probably are not afflicted with this phenomenon. However, it is possible that all your application software must reference information on a particular disk, and that these requests tend to pile up. The users have nothing to do while the request is processed. The system has nothing to do while waiting for the requests to the disk unit to complete. When the system has no users to run, or no work to do, it goes into the idle loop. Now you have poor system throughput and great idle time.

More memory

Not enough memory can cause poor performance and increased idle time. If you add more work, then the system is out of tune and the symptom may be poor throughput. The solution may be more memory.

AOS/VS operates best when there is

just enough memory for all the jobs that need running at any given time. AOS/VS is a demand-paging operating system. That means it will try to keep only active pages in memory. Too much demand for memory turns ugly when AOS/VS must start swapping processes to satisfy the demand for memory. This places extra demands on the CPU and disk subsystems. The natural symptom is decreased throughput.

However, adding more memory might lead to a CPU bottleneck. Great! You then should buy a new, more powerful CPU, with tons of memory. Well, you *can* do that. However, you might find you now have a bottleneck with your disk subsystem.

We are looking at a phenomenon where too much memory can cause an increase in system overhead. Bill Means offers the theory that we might be able to improve AOS/VS or AOS/VS II throughput in certain large memory configurations by modifying the way the operating system handles memory pages. If you would like to participate in this project, call me.

There are few quick solutions to performance issues. Anyone offering a canned answer without a thorough investigation of your system configuration, operating environment, application, and business growth potential, should be avoided. It takes work and effort to manage a system that is a business productivity tool. Never make a decision about your system without an eye on the future.

You are the best-qualified person to answer the question, "What is good system performance?" Because you see what is happening with your system daily, you (and your system users) are the best judge of system performance. You will need to find a reliable professional to help you identify bottlenecks, and recommend long-term solutions to your throughput issues.

Remember, idle time can indicate many things. It always means the operating system has nothing to do. That can be good or bad. You have to be the judge. Δ

Don Thomas is president of NSTS, Inc. He may be reached by phone at 404/923-1383, or by fax at 404/923-3998.

Whetstones, Dhrystones, and MFLOPs

by Thomas E. Soukup
and Edward A. Sepich III
Special to Focus

SYNOPSIS
Freedom of choice in open systems means doing a little comparison shopping—and that leads us directly to benchmarks.

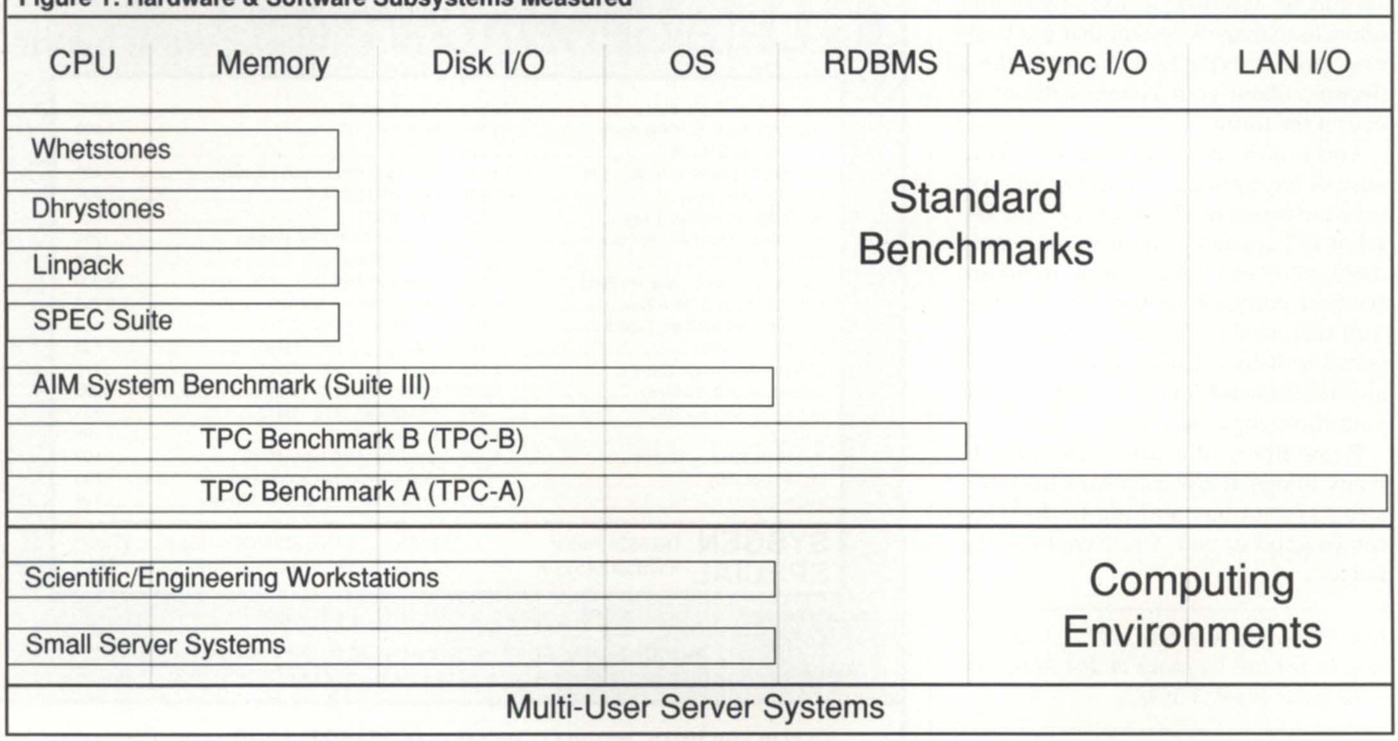
In today's world of high technology and the rapidly advancing environment of computer systems, a business' accurate selection of a computer system is the key to its success. Unlike proprietary computer systems of the past, which limited a customer's hardware and software options, open systems now afford users the freedom of choice.

Once a final decision is made to move to an open systems platform, decision-makers face the dilemma of which open systems vendor to choose. Many standard benchmarks attempt to rank open system platforms, but have they made the choice any easier?

Each standard benchmark measures different aspects of open systems platforms. Some benchmarks measure only the hardware subsystem, such as:

- Central processing unit (CPU)—how fast the computer processes instructions
- Memory system—how fast the CPU interfaces with memory (primary storage)
- Disk input/output (I/O)—how fast it interfaces with disk (mass storage)

Figure 1: Hardware & Software Subsystems Measured



• Terminal input/output—how fast it processes characters to and from the screen/printer and other asynchronous devices
• Network input/output—how fast it processes network traffic to and from other computers.

Meanwhile, other standard benchmarks measure software subsystem-related aspects, such as:

• Operating system efficiency—how fast the computer executes applications, stores and retrieves files, and manages multiple on-line and batch applications running simultaneously
• Relational data base system performance—how fast it runs data base applications
• Multi-user performance—how fast it manages a certain mix of on-line and batch tasks.

Each standard benchmark measures different aspects of open systems platforms.

Rules of thumb

Open systems vendors are continually working to improve operating systems and the data base products they support. Thus, standard benchmark results change so rapidly that there is no fixed result. Simultaneously, as vendors continue improving software and hardware, each revision of a standard benchmark may change the way the benchmark tests a certain subsystem, or may begin testing an entirely different set of subsystems.

How to proceed in such an environment? A few rules of thumb:

• Select standard benchmarks that best resemble your current or proposed computing environment
• Choose a vendor that best addresses your present and future goals, one whose solution addresses system expansion
• Choose a vendor that demonstrates interoperability (communicating within a multi-vendor environment)
• Choose a vendor whose service organization will assist in meeting your short-term and long-term goals on an ongoing basis.

Data General Corporation's open systems platform, the Aviion family, as well as the company's proprietary Eclipse MV product lines, continue to outperform the competition in the multi-user server and network server markets. The results of the AIM System Benchmarks, TPC-B and TPC-A Benchmarks, prove this market advantage.

Benchmarks

Benchmarks are set of one or more application programs used to measure and predict the performance of open systems platforms.

When using standard benchmarks as a basis for predicting how an application or data base product will perform, you should select those measuring subsystems based on your particular application needs.

Standard CPU benchmarks

Workstation users often use standard CPU benchmarks to measure a workstation's pure number-crunching ability. These benchmarks predict how CPU-intensive applications will perform.

Whetstones—"KWhets"

KWhets (thousands of Whetstone instructions per second) is a measuring rod for system performance in scientific or engineering application environments. Operations measured include floating-point and integer calculations, transcendental functions, conditional branching, and array manipulations.

Whetstones is antiquated. The Dhrystones benchmark is currently the industry's trend for predicting how CPU-intensive applications will perform. *Subsystem(s) measured: CPU. Example computing environments: scientific and engineering research environments.*

Dhrystones:"Dhrys" Revision 2.1

Dhrys (Dhrystone instructions per second) is a relative measure of computing power along with compiler efficiency. *Subsystem(s) measured: CPU. Example computing environments: scientific and engineering research environments interested in CPU/compiler performance.*

Linkpack:"MFLOPS"

MFLOPS (millions of floating-point operations per second) is a relative measure of floating-point throughput for both single- and double-precision arithmetic. It predicts CPU speed and Fortran compiler efficiency. *Subsystem(s) measured: CPU. Example computing environments: scientific and engineering research environments depending heavily on Fortran.*

Spec Suite Revision 1.2

Ten compute-intensive programs: Six floating-point F77 double-precision and four non-floating-point "C" programs.

Specmark reflects computation speed. Specthruput reflects computation capacity. Specint reflects integer speed, and Specfg reflects floating-point speed. Floating-point and the compiler's pre-processor performance heavily influence the results. *Subsystem(s) measured: CPU. Example computing environments: scientific and engineering environments depending heavily on "C".*

Standard multi-user benchmarks

Time-sharing users often use standard multi-user benchmarks to measure a server's ability to handle several hundred users performing a certain mix of tasks. These benchmarks predict the number of users a system can handle in a server-like environment.

AIM System benchmark (Suite III) Revision 3.1

The AIM Performance Rating reflects the peak performance of an open system platform measured in AIM multi-user performance units. Performance ratings of a wide range of Unix systems can be compared using available AIM performance reports. *Subsystem(s) measured: CPU, memory, disk, and operating system efficiency. Example: computing environments: custom recordkeeping, accounting, and inventory applications written in "C", Cobol, and other languages.*

Standard data base benchmarks

Data base users often use benchmarks to predict the performance of a specific data base product. Data bases commonly tested include Oracle, Informix, Sybase, Progress, Ingres, and Unify, as well as others.

TPC benchmark B (TPC-B)

TPC-B is an on-line transaction processing data base stress test that simulates a banking industry application. It reports results as transactions per second (TPS) and a five-year cost per TPS. *Subsystem(s) measured: CPU, memory, disk and operating system efficiency, with a specific relational data base product. Example computing environments: recordkeeping, accounting and inventory applications written in a specific relational data base language.*

TPC Benchmark A (TPC-A)

TPC-A is an on-line transaction processing data base stress test that simulates a complete automatic teller machine system. The TPC-A Benchmark reports results as transactions per second (TPS) and a five-year cost per TPS. *Subsystem(s) measured: CPU, memory, disk and operating system efficiency, with a specific relational data base product, local area or wide area network terminal I/O. Example computing environments: record-keeping, accounting and inventory applications written in a specific data base language with heavy terminal and local or wide area network terminal users.*

How to obtain specific vendor results

Open systems vendors can supply the most up-to-date standard benchmark results for their product lines. In addition, many Unix publications and specific benchmarking test organizations provide vendor competitive reports. A few quick references:

- Whetstones: Published in *UNIX Review*, March 1990, & other Unix publications.
- Dhrystones: Published in *UNIX Review*, March 1990, & other Unix publications.
- LINKPACK: Published in various Unix publications.
- SPEC Suite: SPEC publishes results in the SPEC newsletter. Mail orders to: SPEC c/o National Computer Graphics Association (NCGSA), 2722 Merilee Drive, Suite 200 Fairfax, VA 22031; 703/698-9600, ext. 318
- AIMS System Benchmark Suite III: AIM Technology publishes a quarterly *Unix System Price Performance Guide*. To order, call 800/848-UNIX, or write to: AIM Technology, 4699 Old Ironside Drive, Suite 150, Santa Clara, CA 95054.
- TPC Benchmark B (TPC-B): Published by the TPC Council, c/o Shanley Public Relations, 777 North First Street, Suite 600, San Jose, CA 95112-6311; phone 408/295-8894, or fax 408/295-2613.
- TPC Benchmark A (TPC-A): same as for TPC Benchmark B. Δ

Thomas E. Soukup is the America Sales and Service Division Benchmark Coordinator, and a member of the Western Area Technical Services Group at Data General Corporation, Schaumburg, IL. Edward A. Sepich III is founder, chairman, and president of Total Environmental Systems Services Corporation, (Schaumburg, IL), a management advisory company.

Well-fed and humming along

SYNOPSIS

Don't forsake common sense when it comes to tuning your computer system. And when common sense is lacking—throw more memory at it.

by Rick Havourd
Special to Focus

The whole issue of system performance and tuning really comes down to what your own needs and desires are for the computer system you manage. What are you doing with your machine? What do you expect to get from your machine?

How you use it will play the biggest role in the system's overall performance. Using your workstation for desktop publishing or application development certainly is different from using a server as a data base engine or traditional timesharing system.

There are a number of issues common to all of these environments, and that's what I'll be covering in the rest of this article. You may then test these ideas (and some of your own). Keep the ones that work for you, discard those that don't. But you shouldn't be expecting miracles. Computer performance is a lot like that of an automobile: when you first purchase it you're amazed at the speed, agility, and overall handling. But as time goes on it just doesn't seem as fast as it used to. Tune-ups, small repairs, and high-quality gasoline all improve the vehicle's performance, but in isolation none will amaze you.

The memory issue

With the Aviion and with RISC systems in general, there is a very simple and limited instruction set. With 51 different instructions it takes quite a bit to piece them together into something meaningful, i.e. your program. Because of this, your machine is going to run through 16 million to 33 million instructions every single second. The problem is that 33 million instructions probably take up most if not all of your system's memory. For those who missed it, our first keyword is: *memory*.

With complicated programs made of simple instructions, it's going to take quite a bit of memory to keep the processor well-fed and humming along. Save all that money trying to figure out where machine cycles are going and load up your machine with memory.

In an earlier *Focus* article, I said I was running on an AV 310c with 12 MB memory. You want to talk about a performance dog? Well, this one came with fleas! I was never quite alone, as my disk drive kept me company with its incessant chatter (paging entire applications between memory and disk makes a lot of noise).

The other day we finally broke down and purchased more memory: 16 MB from SCIP systems cost us less than $1,500 and it's the best thing I ever did. So now with 28 MB my machine has been transformed from the swapmeister into the real screamer the salesperson sold me.

A little experimenting has shown us that approximately 5 MB is used by DG/UX; 4-6 MB for X11 and Motif. Add to that other, noncritical but useful utilities for 2-4 MB, and you're sitting at a bare minimum of 12 MB. Once you break the 16 MB barrier, life improves rapidly .

To test your own memory utilization, here's a command you can use: *sar -r 1 1* will give you the instantaneous free memory pages and free swap blocks.


Presenting the
ECLIPSE MV/60000 HA™
It's the dawn of a new age of corporate computing!
Welcome to the 6th generation of ECLIPSE®-the generation that reaches new heights in corporate computing while protecting your software investments. The MV/60000 HA gives you:
MORE POWER–100 real MV MIPS supporting over 1000 CEO® or database users with up to 4 CPUs and 1 GB of memory.
GREATER SYSTEM AVAILABILITY–including integrated diagnostics, N+1 power and cooling, AOS/VS II, MRC, and the NEW H.A.D.A./MV disk array.
BREAKTHROUGH TECHNOLOGY–single board 27 MIP CPU, latest Motorola gate arrays, 600 MB/sec asynchronous bus, mid-plane design.
ALL LEADING INDUSTRY STANDARDS–supports SCSI II, TCP/IP, NetBEUI, AppleTalk,® Novell,® OSI, SNA, X.25, and more!
MEETING YOUR NEEDS NOW–whether you require increased processing power, system consolidation, or mission critical confidence, the MV/60000 HA will take you into a new age of corporate computing.
Call 1-800-DATA GEN
Data General
ECLIPSE MV/60000 HA is a trademark of Data General Corporation.
ECLIPSE and CEO are registered trademarks of Data General Corporation.
AppleTalk is a registered trademark of Apple Computer.
Novell is a registered trademark of Novell, Inc.
© 1992 Data General Corporation
Circle 10 on reader service card.

This is useful in determining overall memory/swap utilization. Each page is 4,096 bytes, and each block is 512 bytes. The more free pages avalable, the better. As fewer free pages are available, swapping will increase.

The disk issue

The next major subsystem on your machine is the disk. The Aviion integrates a state-of-the-art SCSI (small computer system interface) controller directly on the system. This allows us to use the latest in disk technology—very fast disk drives that typically have an access time between 12-15 ms. Between the computer and the disk drive, data passes over the bus at 2 MB per second. This provides a high-quality, high-speed storage solution. For the most part disks won't be a major bottleneck for you, but let's consider a couple of situations that could use a little help.

A data base server is a prime candidate for disk bottlenecks. Many of today's RDBMSs feature transaction control mechanisms and data integrity features, such as two-phase commit and transaction rollback facilities. Quite a bit more disk activity per record update occurs than with earlier record management techniques. By utilizing multiple disk drives, you can potentially improve performance.

The simplest method for doing this is to place the data base file on one spindle, and the before-image or data integrity file on a different spindle. ("Before-image" is the term used by Progress to indicate a secondary file the RDBMS uses to store an image of the record prior to making modifications—substitute your own terminology here.)

Moving your swap file system to another drive can help if your system does a fair amount of paging. I'm really not very keen on spreading individual file systems across disks because the segments spanning multiple disks are allocated one disk at a time. So if you have a 300 MB file system, with two 150 MB segments on two drives, the second 150 MB will not be used until the first has been filled. When the first fills, all new allocations will occur on the secondary drive, thus moving all that activity from one disk to the other—not balancing the load the way we want. With multiple disk drives, you want to attempt to balance the data access evenly across as many drives as possible. Nobody really cares if you have equal amounts of storage allocated. It's the number and size of data accesses that will affect your performance.

The terminal issue

One seldom considers the impact that ASCII terminals may have on a system. Each and every time you press a key, the processor must be interrupted to service it. At the very least, the character is buffered and then echoed back to you. Put a bunch of touch-typists on your machine and this can become quite a chore. The use of VDA/VDCs (direct connect terminals) don't provide you with much choice in the matter, but this type of access is pretty basic and incurs little overhead.

On the other hand, in our modern networked society we have the terminal server. A terminal server is a device used to connect ASCII terminals to a computer via ethernet. It's a great device that provides a lot of flexibility, but it brings its own problems as well. Terminal servers on a TCP/IP network use telnet protocol to communicate with the host. This is an expensive method because each key you press is packaged up in a TCP packet and fired off. The host must receive the packet, interpret the data, pass your single keystroke off to the terminal handler, take the echoed character, create a new packet, and send it back to the terminal server. As you can see, this is quite a bit more involved than a direct connect.

For sites with high terminal counts, and yet need the benefits of a terminal server, DG offers a nifty little device called a VME Terminal Controller. It allows terminal counts close to "direct connect" numbers without significantly affecting performance. This is accomplished by offloading the networking overhead associated with telnet sessions to the controller, so the computer need only deal with the individual characters.

The common sense issue

Let's not forget everything we've learned about computers and performance. Just because you've got this rocket sitting next to you doesn't mean it'll never run out of power. There will always be jobs to run that aren't as necessary or time critical as others. DG/UX provides those time-tested notions of batch processing and job priority.

Jobs can be scheduled to run at off-peak hours using the *at* and *cron* programs. The *nice* and *renice* commands allow you to increase or decrease the scheduling priority of individual processes. Typing *renice 19 -p 123* will adjust process #123 so that it will run only when the computer has nothing else to do. On the other hand, *renice -20 -p 123* will cause this process to run at the expense of all other jobs. Let's not go crazy with these commands; it is a bad idea to monkey around with the priorities of system processes. You may feel comfortable adjusting them so long as a little common sense is used. For example, if your data base application seems a bit sluggish, raise the priority *(nice--5 or renice -5)* of the server; do *not* raise the priority of the people using it.

That's all I have for this time. If you get anything out of this, I hope that it's a respect for common sense. And when common sense is lacking—throw more memory at it. You may also wish to spend some time getting acquainted with *sar*, the System Activity Reporter. Congratulations, you can now safely go out and tune-a-Aviion but, I'm sorry to say that you still can't tune-a-fish! Δ

Rick Havourd is a partner with Micro Sage Software Systems and may be contacted at 130 South First Street, Ann Arbor, MI 48104; 313/663-0444.

Brian Johnson

Happy tuning

SYNOPSIS
BJ returns to system performance analysis and tuning, because it's one subject that bears repeating.

It's appropriate that the theme of this month's issue is system performance analysis and tuning. Just last week I got reminded that just because I covered some aspect of system performance in detail during the last five and a half years, that doesn't mean that it doesn't bear repeating occasionally. So, this month I'm going to do a review of some of the basic principles involved with AOS/VS [II] system performance analysis and tuning.

If you want more detail on any particular issue, you can check the index for all my past columns, available on the :SYSMGR BBS, and then download the particular columns involved.

Unless specifically noted, everything I'm about to tell you applies equally to both AOS/VS and AOS/VS II. In fact, it also applies generally to just about every other operating system on the planet. Galaxy-wide, I'm less sure.

:BACK_TO_BASICS

In any situation where multiple consumers share a limited set of resources, clearly the problem of supply versus demand is critical. The job of any time-sharing operating system is to apportion access to the shared resources among competing users according to a set of rules that determine how sharing takes place and how to resolve conflicts.

The first step is to identify the shared resources. There are lots of them, but the three major ones we're going to concern ourselves with here are memory, CPU, and disk I/O.

The second step is to get your hands on a piece of software that will help you accomplish the job of analyzing and monitoring system performance. AOS/VS [II] has the unfortunate distinction of being one of the few operating systems that requires that you answer questions (like system cache size), and then gives you no way to find out how good your answer was. Rather than mention my product and have to give equal time to my competitors, I'll just leave it to you to figure out what's available and from whom.

:QUEUING_THEORY

Classic queuing theory offers interesting insights into what happens to systems that involve a shared resource, and consumers who make random demands (e.g., interactive users).

There are many ways to organize a system, but one of the simplest to analyze is what the queuing theorists refer to as an M/M/1 system. The assumptions made for an M/M/1 system are:

1. Requests for service arrive at an average of *t1* time units with a Poisson distribution.
2. The time it takes to honor a request for service is also described by a Poisson distribution with an average of *t2* time units.
3. The service is not preemptible; if the service provider is busy when a new request arrives, the new request enters a queue to wait for service using a simple FIFO priority.

The graph describing average queue length and its standard deviation (a measure of expected variability) is shown in Figure 1 (page 20).

Based on the predictions that the theory makes about the behavior of M/M/1 systems, we can safely state the following rules:

- If demand exceeds supply, an infinite length queue of requests is the result
- As demand rises, the length of the queue of waiting requests grows according to an inverse function (i.e., slowly at first, very quickly as the service provider gets busier)
- The unpredictability of the average queue length grows according to an inverse square function (i.e., stability gets worse faster than the average queue length does).

> The rule for the memory resource is simple: you gotta have whatever you need.

Probably the most non-intuitive result from queuing theory is that in a system involving random requests for resources, you can never use all of the available resources. In fact, you have to limit your use to a relatively small fraction of the total resource in order to ensure that consumers will receive both timely and predictable response time. "Timely" means that the average queue length is reasonably short; "predictable" means that the queue length is generally the same size over time.

:MEMORY

Because users tend to acquire memory at the beginning of their lifetimes and hold onto most of it until they terminate, memory violates assumption No. 1 for an M/M/1 system. The results of queuing theory generally don't apply until you run out of memory and stealing and/or swapping starts to occur.

In addition, the relative cost of resolving a memory resource conflict (i.e., the time it takes to steal a page and load its contents) are orders of magnitude more expensive than when no conflict exists (zero cost), so it is generally impractical to have less memory than the total of what's required by all users at peak times.

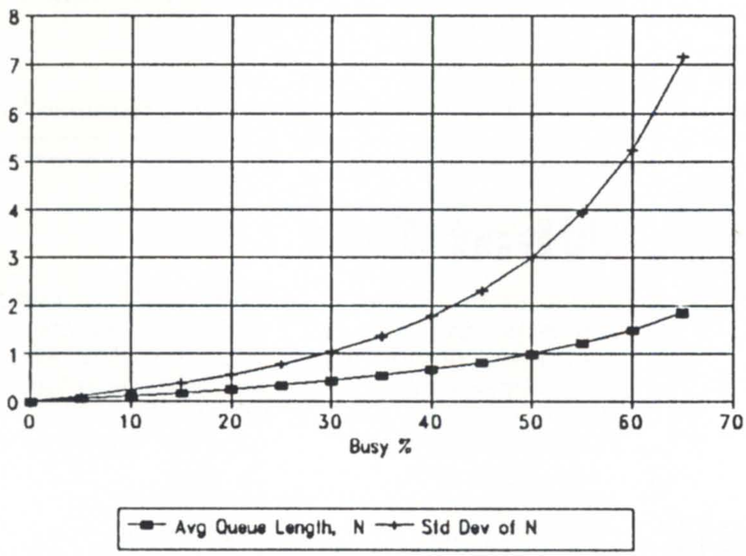
Based on this, the rule for the memory resource is simple: you gotta have whatever you need. The cost of violating this rule is terrible, and common strategies to minimize its impact are generally ineffective, and result in nowhere near the speed-up that is obtained by simply adding more physical memory.

:CPU

Most operating systems include fairly

sophisticated algorithms for resolving competition for the CPU: simple FIFO as assumed for an M/M/1 system is rarely the algorithm used. In addition, CPU violates assumption No. 3 because it is relatively inexpensive to interrupt servicing one request in order to honor another (presumably) higher priority request. The net effect of these modifications is to move the knee of the curve in Figure 1 down and to the right, making the corner occur much lower and later (at about 85 percent), and be much sharper. As a result, you can generally run preemptive resources much closer to their limit than non-preemptive resources before long queues occur.

Figure 1: M/M/1 Behavior



In order to postpone CPU contention for as long as possible, the AOS/VS scheduler allows three kinds of prioritization:

- Explicit process priorities
- Adjustment within a priority based on interactive versus batch-like process behavior (i.e., heuristic scheduling)
- Process class (or group) priorities and/or allocation by percentage (requires the optional CLASP package).

In addition, the AOS/VS scheduler contains some sophisticated and effective logic that attempts to keep the scheduler overhead from being far less than a linear function of PID count.

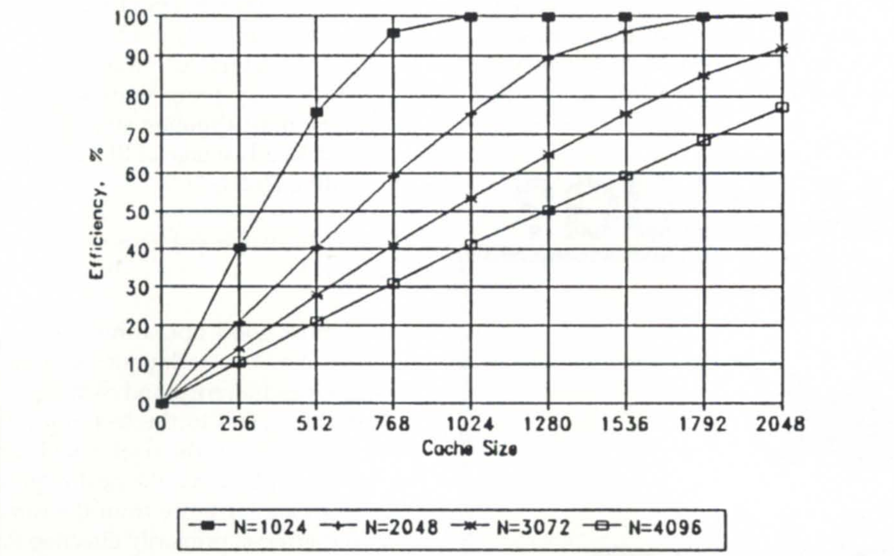
One thing is important to bear in mind about CPU priority schemes: they have no effect unless contention is occurring. Think about that for a second; if there generally is idle CPU, then everyone is getting all they want by definition. Even if there's no idle CPU, things still might be all right.

Consider the case in which 100 PIDs are using 50 percent of the CPU and several batch jobs running at an explicitly lower priority are using the other 50 percent. Except for some tiny delays imposed by virtue of queues that can form at the entrance to single-threaded paths in the operating system, the on-line users (i.e., the ones who collect a salary) are going as fast as if batch jobs were not running.

I've conveniently ignored contention due to both on-line users and batch jobs beating on the same disks, but we'll talk about that shortly. The point I'm trying to make here is the importance of running batch jobs at a priority below the on-line users, in order to keep the on-line users running as fast as possible.

So what happens if the on-line users alone are trying to consume more than 85 percent of the CPU? Easy: response time starts to go to hell, rapidly and massively. Will explicit prioritization solve this problem? Only if you're one

Figure 2: Cache Behavior



of the lucky ones who get the high priority. The operating system can't manufacture CPU out of thin air, or trade it off for any other resource.

The only way to solve a problem where the on-line users are persistently attempting to consume more than 85 percent of the CPU is to a) buy a faster CPU; b) reduce the CPU consumption of the user programs; or c) keep taking users out to the parking lot and shooting them until 15 percent average idle is achieved.

:DISK_IO

Of the three system resources, disk I/O is the one that most closely behaves the way that an M/M/1 system does. Requests arrive pretty much randomly and are not preemptible (you can't cheaply switch to another request in mid-request).

Because of the relatively large cost of disk I/O (measured in tens of milliseconds as opposed to nanoseconds), the effect of doing excessive disk I/O is gruesome. The designers of early operating systems quickly recognized this fact and took steps to reduce the susceptibility of systems to disk bottlenecks. The most common strategy they use is to take advantage of BJ's Rule No. 1 Regarding Disk I/O:

***RULE 1:** The fastest disk access is one that is avoided.*

Like most good operating systems, AOS/VS uses two techniques to avoid disk accesses: buffering and caching.

Buffering refers to the technique of reading more than the user asked for, on the assumption that the next request will be asking for more from the same area. Buffering is primarily effective for sequential I/O.

Caching refers to the technique of trading off physical memory for disk activity by keeping copies of the most recently accessed N blocks in memory, and managing them using an algorithm that attempts to predict which blocks are most likely to be referenced again. In addition, there are both read/write (e.g., AOS/VS system and shared page LRU cache) and write-thru caches (e.g., AOS/VS II disk data cache). Read/write caches achieve lower overall physical I/O by trading it off against the increased possibility of file system damage after a crash. Caching is primarily effective for random I/O.

In the case of buffering, the optimum strategy is to read or write as much as possible each time you access the disk. How this is accomplished depends on the language you're using. Consult the section of the manual on optimization for clues. Of course, everything has a point of diminishing returns, and in the case of sequential I/O that point is usually in the neighborhood of 12 to 16 kilobytes at a time (24 to 32 disk blocks). Needless to say, massive buffering makes sense only if you have enough spare memory to do it without causing paging or swapping.

In the case of caching, the general behavior of LRU-based schemes (the kind used by AOS/VS) is shown in Figure 2 (page 21). This graph was produced using a simulation program where the number of blocks competing for the cache was 1024, 2048, 3072, and 4096, and requests were subject to a statistically normal distribution. Here again, we see that there is a point of diminishing returns in the tradeoff between memory (cache size) and disk activity reduction (efficiency). In addition, an oversized cache imposes a slight increase in CPU consumption due to the overhead of searching an excessively long buffer list for each failed request.

The high efficiencies that read/write caches typically are capable of (98 percent or better versus 30 percent to 50 percent for a write-thru) is both a boon and a bane. On one hand, the reduction in disk activity is formidable (50:1 in the case of a 98 percent efficiency), but on the other hand even a slight change in efficiency of 1 percent when running at 98 percent can either double or halve the amount of disk activity due to cache misses. It is critical that you monitor efficiency closely and respond to any significant change quickly.

In the case of caches, the optimum strategies are to a) make sure the cache is sized at least as large as the point of diminishing returns; and b) reduce the number of blocks competing for the cache as much as possible.

The first part is easy: you try various cache sizes and plot the efficiency to find the point of diminishing return.

The second part is trickier: you have to identify which kinds of blocks are competing for the cache, and then try to reduce the number of them. In the case of the system cache, the culprits on most systems are random index blocks (RIBs), directory, bitmap blocks, and IPC spool blocks (in order of activity). In the case of the shared page LRU cache, the culprits are program file (.PR) and shared data base pages.

Reducing random index blocks is easy: simply change the element size of all active files to have no more than one index level. Contiguous files (zero index levels) are only marginally better in terms of reducing cache activity, and generally not worth the system management headaches they cause (e.g., Error: insufficient contiguous blocks).

On AOS/VS Classic, reducing directory blocks is a bit trickier. Changing the hash frame size is helpful, but pales in comparison to the effect of the sheer size of the directory itself. Unfortunately, convincing users to break up huge directories into a series of smaller subdirectories is an ongoing battle for most system managers.

The internal directory structure used by AOS/VS II is quite a bit different from that used by Classic, and some early experiments I did by varying the File Information Table (FIT) element size for the LDU showed that the cache activity is relatively immune to changes in the FIT size, and more a function of the sheer size of the directory. This makes it effectively a parallel to the situation under Classic (unfortunately).

Activity due to bitmap and IPC spool blocks is generally an order of magnitude less than that due to RIBs and directory blocks, so they generally aren't worth worrying about (yes, Virginia, there *are* exceptions).

By now you probably noticed that I haven't said anything about disk space. The reason is because disk space is not really a performance issue in most cases, it's simply a management headache. While it is true that disks that are more than 90 percent full are significantly slower to access than those less than 90 percent full, the benefits do not continue as the percent full drops below about 80 percent.

The simple reason for this is that real-world disk requests tend to violate the assumptions of M/M/1 in a couple of subtle ways: first, requests tend to arrive in bursts as opposed to more evenly spaced (e.g., it might take 5 or 10 I/Os to open a file); and the requests tend to cluster tightly on the disk (e.g., an open hits a relatively small area of a single directory file).

This explains the apparent anomaly that you see when you look at average seek distances on DISCO. If requests to the disk were truly random, we would expect to see an average seek distance of 1/3 the number of cylinders in use (e.g., about 250 for a 90 percent full 354 MB disk). Instead the average seek distances are typically a fraction of that; more like 100 or less.

Check it out for yourself on your own system. I just did, and the disk I'm sitting on right now—which incidentally hasn't been reloaded in more than three years—has an average seek of 52 cylinders. It is precisely this effect that makes the seek time for a disk relatively immune to how full it is and how well it's organized.

The only exception to this is during single-user sequential processing, like batch and DUMP/LOAD, where caching is relatively ineffective due to the low probability of reuse. As a result, average seek distances usually go up during these activities.

:FILE_SPACE_EXHAUSTED

Well, I'm outta space again and I still haven't gotten to the bulging file of fresh clippings from the wild and wacky world of the trade rags, so I guess that'll have to wait until next month. Happy tuning. Δ

BJ is the President of B.J. Inc., a San Francisco based consultancy specializing in system auditing, system management, and performance analysis. :SYSMGR is a division of B.J. Inc. BJ can be reached at 109 Minna St., Suite 215, San Francisco, CA 94105, 415/550-1444 (voice) or 415/550-1072 (fax). The :SYSMGR bulletin board number is 415/391-6531 (300/1200/2400 with optional MNP class 5, CHAR/605X/CHARLEN=8/PARITY=NONE/AUTOBAUD) or 415/550-1454 (voice).

David Novy

Yes, *sar*

SYNOPSIS
Although DG/UX system performance measuring tools are not spectacular, they are adequate.

This month's issue of *Focus* is dedicated to performance monitoring and tuning. At present, there is not a broad suite of performance monitoring tools available for DG/UX. All that's officially available is the Unix *sar* command.

The *sar* command is a batch process that is not capable of doing real-time analysis of Unix system performance. It does give the system manager the capability of capturing system performance for a number of intervals over a specific period of time, so *sar* can help a system manager determine the need for more CPU, memory, or disks.

However, *sar* is a DG/UX feature badly needing a facelift. Today's system managers expect real-time data capture and analysis, preferably displayed in graphical form; *sar* cannot do this.

Fortunately, help is on the way. It appears that the DG/UX development team is working on enhancements to *sar* that will modernize it and increase its usefulness to those concerned with performance measurement and tuning. It would also appear that the *sar* enhancements will be available before the end of 1992.

If the end of 1992 comes and the *sar* enhancements are still not available, then blame the author and not Data General. The time estimate was made by the author based on pure emotion. DG has given no official response as to when the *sar* enhancements will be available. However, the DG/UX development team has been very responsive to user requests, and the request for *sar* enhancements was a primary concern to the attendees at the 1991 NADGUG Conference.

If you need to use *sar* in its present form, a very good explanation of how to do so can be found in the man pages. The man pages explain the use of each of the *sar* switches. They also explain how to run *sar* as a background process.

If you want to give *sar* a try, here are a couple of sample commands that will give you some useful information regarding system performance. The first command is:

```
sar -o temp 60 10
```

This command starts the *sar* process, and has it run for 10 intervals of 60 seconds each. This command's output is CPU usage on the basis of user, system, and idle. The data are also sent to a file named *temp*.

Another useful *sar* command:

```
sar -d -f temp
```

This command gives an analysis of disk usage information using the same *temp* file created by the first *sar* command example.

Managing DG/UX machines

It should be noted that, although the DG/UX system performance measuring tools are not spectacular, they are adequate. Also, since open systems memory and peripherals are not as expensive as DG MV memory and peripherals, then system performance is not as critical as it is with DG proprietary equipment. Some concepts found to be useful in helping me manage my DG/UX machines are as follows:

1) The *sar* command can be used to determine CPU utilization.

2) The *dfk* command may be used to determine disk capacity at any given time. For optimum performance, you should try to keep disk capacity below 70 percent. Disk performance degrades dramatically if this level is exceeded.

3) Use the *sar* command to determine disk memory and swap disk utilization. The most efficient systems are those in which there is no swap disk utilization. This is because accessing information from disk is at least 1,000 times slower than accessing information from memory. Installing additional memory is probably the least ex-pensive means of increasing system performance. When increasing Unix workstation memory from 16 MB to 32 MB in a CAD/CAM environment, I have observed 100-percent improvements in system performance. Going from 32 MB to 48 MB resulted in significant performance increases.

> The *sar* command is an adequate tool for determining the system performance of DG/UX, but it needs to be made easier to use and it needs to be able to give real-time performance information

4) Consider using third-party memory. Third-party memory generally costs at least 50 percent less than proprietary memory, and it usually comes with a lifetime warranty and overnight replacement. You need to realize that memory has become a commodity item. Both proprietary memory boards and third-party memory boards use the same chips. DG does make very good memory products, but so do the larger third-party memory vendors.

5) When making recommendations to users regarding system configuration cost estimates, err on the side of too much. Never err on the side of too little. It is easier to allocate more money than you actually need for the purchase of computer equipment if it is part of the official budgetary process, than to overcome performance bottlenecks by trying to find additional money outside of the official budgetary process. I have had customers tell me that I let them down when I told them they needed to spend an unbudgeted $5,000 to overcome a bottleneck. This $5,000 was within 10 percent of my original cost estimate, so I originally felt rather satisfied with my forecasting ability, However, my customers soon informed me that forecasting 10 percent too little is a lot worse than 25 percent too much. If you are going to fight a budget battle to obtain funds for a computer system, make sure that you only have to fight the battle once.

The *sar* command is an adequate tool for determining the system performance of DG/UX, but it needs to be made easier to use and it needs to be able to give real-time performance information. Fortunately, the cost of open systems hardware and peripherals is much lower than that of proprietary hardware, so system tuning is not as critical. Fundamental DG/UX performance considerations are to have plenty of memory, do not overload the disks, and purchase a CPU large enough to do the job. Δ

David Novy is a technical computing specialist at 3M in St. Paul, Minnesota. He is past chairman of the AOS/VS special interest group and current chairman of NADGUG's SIG/UX.

Kim Medlin

Real-world report writing

SYNOPSIS

This article is the sixth in a series, examining a software development project using RAD tools and techniques.

In this series of articles, we are following a logical progression of software development steps for a real-life application called the Loan Tracking System. Specifically, the tools we are using allow for "rapid application development," or RAD. Previously, we explored the analysis and design phases of development using a CASE tool (computer-aided software engineering). Next, we began developing the individual application functions using a fourth-generation language (4GL).

Now, we continue in this vein with a look at one of the most powerful report-writing tools on the market today: Oracle's SQL*Reportwriter. Bear in mind that other 4GLs contain report writers, also. I am merely using Oracle here as an example of the types of functionality that can be found in current RAD tools.

It is hard to overstate the impressive abilities of current 4GLs. Real-world application software can be generated in a fraction of the time expected with a 3GL like Cobol or C. On top of that, the quality and stability of functions written with a 4GL are often superior to that of a 3GL.

Here at Data General's Systems Integration Services (formerly Solution Services), we write custom software for Data General customers every day. I personally have been designing and writing software applications for 12 years, and I am truly amazed at the speed and precision with which a good 4GL creates software.

The simple case

Let's take a look at how we would tackle a simple reporting function with SQL*Reportwriter. Let's assume that all we need to do is list the contents of the student status table. This table contains only two fields: the student status code and description. The table contains only a small number of records. Our objective is to print all records on the page.

Figure 1: The Simple Case

```
Student Status Codes

Code    Description
-----   -----------
A       Active
C       Collections
F       Forgiven
I       Inactive
J       Judgment
P       Paid
*** End of Report ***
```

Because 4GLs are written to take advantage of the structured query language (SQL; pronounced like "sequel"), the development of this report will take only a few minutes. The heart of the effort is in specifying the SQL statement that will retrieve the desired records:

```
select
        sst.code,
        sst.descr
    from sst
    order by sst.code
```

This SQL statement is as straightforward as it appears. "sst" is the name of the table; "code" and "descr" are the names of the two fields. The statement simply selects the fields we require

Figure 2: One More Step

04/01/1992

Student Status Codes — Grant Status Codes

Code	Description	Code	Description
A	Active	0	N/A
C	Collections	D	Medical Doctor
F	Forgiven	M	Ministry
I	Inactive	P	Paid
J	Judgment		

*** End of Report ***

from the appropriate table and specifies the order in which they should be returned. This is a key benefit of SQL (and 4GLs, too). A SQL statement declares *what* information you need. It is the responsibility of the SQL engine to determine *how* to physically fulfill your request.

After adding a report title line (which takes about two more minutes), we have generated the report that appears in Figure 1 (page 28).

One more step

The status report example is about as simple as they get! Almost any report writer could handle that requirement with ease. Now, let's say we want to modify the report and print the "grant codes" to the right of the student status codes, and on the same page.

No problem. Just add another SQL select statement to the report definition. In this case, we would add the statement:

```
select
      grnt.code,
      grnt.descr
   from grnt
   order by grnt.code
```

We would also specify that the group of records selected from this query should appear to the right of the first set of records. The entire time spent to add the second list to the report is no more than five minutes. After some simple formatting adjustments, the report now appears in Figure 2.

It is important to realize that this seemingly simple functional enhancement would have blown some supposably production-quality report writers right out of the water! That is because some SQL-based report writers allow only one select statement per report definition. In contrast, we have never found even a single report that SQL*Reportwriter cannot produce.

This emphasizes the importance and value of selecting an appropriate 4GL for the requirements of an application. Bear in mind that evaluating the major 4GLs on the market is no trivial task. Before selecting a 4GL for serious development, it is wise to seek the advice of a consultant competent in such matters.

A real-world report

No more kid stuff. Let's see what a more realistic report definition looks like. From the Loan Tracking System, I will select the Payments Due Schools By Date report.

This report lists due dates and amounts for payments to individual schools. This allows the application administrator to project upcoming allo-

Figure 3: Payments Due Schools By Date

Due Date	School Name	Due to School
12/01/1992	Anderson University	1,000.00
	East Carolina University	6,800.00
	High Point College	1,400.00
	University of N.C. at Charlotte	8,750.00
	University of N.C. at Greensboro	7,250.00
	Due Date Total:	25,200.00
12/15/1992	Brigham Young University	1,000.00
	St. Andrews Presbyterian College	900.00
	University of N.C. at Wilmington	900.00
	Western Carolina University	4,000.00
	Wingate College	2,000.00
	Winston-Salem State University	750.00
	Due Date Total:	**9,550.00**

*** End of Report ***

cations from the foundation's checking account.

Just as in the simple reports, we specify with a SQL select statement which records should be reported. The select statement required in this case will extract records at the same time from two individual tables; the loan due table *(ldu)* and the school table *(sch)*. The act of retrieving records from more than one table is called a "join."

The following select statement retrieves the desired data:

```
select
  to_char(1ldu.due_date, 'mm/dd/yyyy')
due_date,
  sch.name,
  sum(ldu.due_amt) tot_due
from ldu, sch
where ldu.sch_code = sch.code
  and ldu.due_date between :beg_date and
:end_date
group by ldu.due_date, sch.name
order by ldu.due_date,
```

Once again, the statement first lists the individual fields to be returned. This is very straightforward except for the due date. The "to_char" function returns the date as a character item in the format of "mm/dd/yyyy." Had this function not been used, only a two-digit year would have been returned. Notice also that SQL is calculating a total for each date and school combination with the "sum" and "group by" clauses. The second part of the "where" clause ensures only records within a user-specified date range are selected.

When the data structures were specified during the design phase, a one-to-many relationship was defined between the *sch* table and the *ldu* table. That is, for each *sch* record there exists one or more *ldu* records. The information that relates these tables together is the common field *sch_code* (school code). It is this ability of SQL to relate tables with common fields that causes us to refer to this type of data base as "relational." The actual join is defined with the statement, *where ldu.sch_code = sch.code.*

After adding several report formatting parameters, the finished report appears in Figure 3 (page 31).

If it looked simple to create this report, you are right. Once you know SQL and a good 4GL reporting tool like SQL*Reportwriter, a task like this is easy and quick to implement. This particular report took less than an hour to write from scratch. When compared to writing a report program in Cobol or C, I think you will agree that this meets any realistic expectation of "rapid application development." Δ

Kim Medlin is a Senior Consultant with Data General's Systems Integration Services group in Atlanta, Georgia. Systems Integration Services specializes in custom software design, development, implementation, and consulting. His address is 3617 Parkway Lane, Norcross, GA 30092. He may be reached at 404/448-6072, ext. 2007.

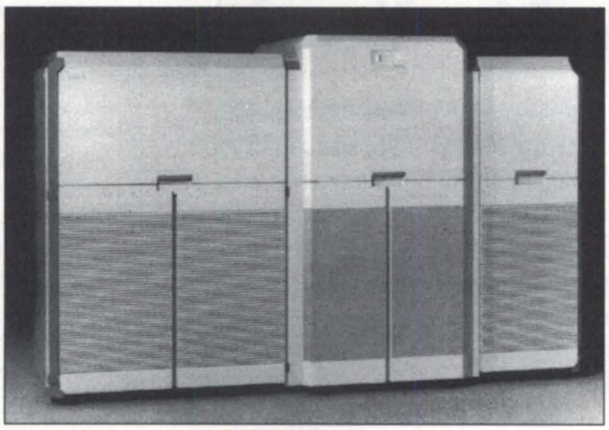
New family members

SYNOPSIS

Six generations of anything is quite an accomplishment. For Data General and its recently announced mid-range and high-end MV minicomputers, six generations represent a commitment to a product line and a long-range strategy.

by Doug Johnson
Focus staff

Trek backward six generations in an American family and you could run across a Revolutionary War veteran. Such a journey reaches back a long time. But genealogy races swiftly in the computer industry. Six generations of high technology may flash past in . . . oh, say about a dozen years.



MV/60000 HA

Six generations of anything is quite an accomplishment. And it should at least indicate something about commitment and long-range goals. Data General Corporation, once a brash upstart with no family connections to speak of, now boasts six generations of the Eclipse MV minicomputer line it began producing in 1980.

"The fact that you're at that high a number means this is not just some short-term strategy," said Dave Ellenberger, DG's vice president of Eclipse marketing. "I think one thing it means to our customers is that they continue to have a combination of leading-edge technology and a protection of their investment."

A gala event

Exhibiting its commitment to the continuing MV product line, DG on April 8 did just that: put on an exhibition. During a much-publicized executive business symposium the company hosted and sponsored at Harvard University, attendees heard leading economists and technology futurists discuss America's ability to compete in tomorrow's global marketplace, and observed demonstrations of the two newest MV systems, as well as a new MV disk array subsystem.

In action were the MV/60000 HA high-end system, the MV/35000 mid-range system, and a mass storage subsystem for the entire MV family called the H.A.D.A./MV (high availability disk array).

"These products demonstrate how Data General has taken advantage of technological change to keep our MV family on the leading edge," said Joel Schwartz, DG vice president and general manager.

Ellenberger termed the event "a first-class unveiling."

"I guess I would be boasting if I said that it was the best announced event in our history, but I have had that feedback," he said. "I think we had well over a hundred customers, top executives from our top customers around the world, represented at the event. And we had a set of world-class business speakers." The theme, "How to Win in a Changing World," featured Lester Thurow, dean of MIT's Sloan School of Management; *Megatrends* author John Naisbitt; *2020 Vision* author Stan Davis; and Tom West, DG's senior vice president of advanced systems development.

Product demonstrations were conducted right after lunch.

"That [session] was packed, with people trying to get in to see those machines," Ellenberger added.

MV/60000 HA

A new, multiprocessor high-end computer system representing one of Data General's largest-ever product developments, the MV/60000 HA is intended for customers supporting large numbers of users, and those seeking to consolidate current computer applications and operations onto a single system.

"We've done more simulation with this machine than any other in [our] history," says Ellenberger.

According to Schwartz, "The MV/60000 HA doubles the performance of our high-end MV/40000, offering mainframe power with a single-board CPU."

Based on Motorola's ECL gate array technology, the MV/60000 HA employs a scalable, multiprocessing architecture that provides 27 to 108 Dhrystone MIPS (million instructions per second). The cache design, memory architecture, and high-performance system bus increase performance incrementally as more processors are added, enabling the MV/60000 HA to support more than 1,500 users in a commercial environment.

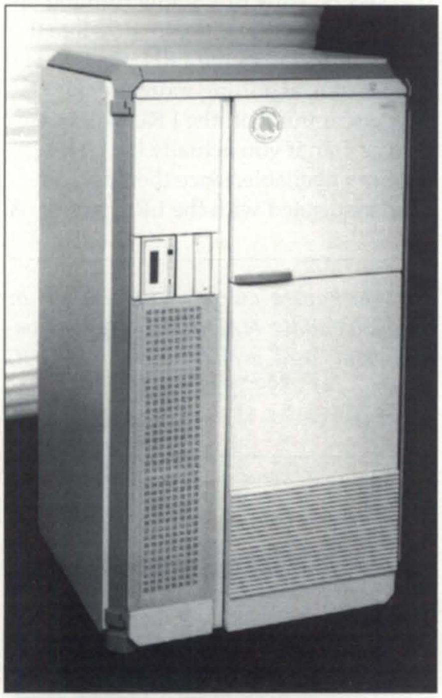
"To reach those levels, you need to make sure that your software is tuned," said Ellenberger. "So a lot of effort was involved there, in software development, and ensuring that we can reach these unprecedented levels of support."

The MV/60000 HA can accommodate from one to four job processors, from 128 MB to 1 GB of memory, 2 to 6 I/O channels, and up to 720 GB of direct mass storage. The system will also offer full interoperability with DG's Aviion systems and Dasher PCs. Pricing for the MV/60000 HA starts at $750,000, and will begin shipping in August.

MV/35000

The MV/3500 is a mid-range system designed for multi-user systems and servers. It offers the distinction of being the first DG system designed for six-way multiprocessing, while providing substantially increased performance over previous mid-range MVs. An improved version of DG's customer CMOS microprocessor provides 40 percent or greater performance than a comparably configured MV/30000.

A scalable multiprocessor, the MV/35000 is configurable with up to six CPUs, 512 MB of memory, three I/O channels, and up to 360 GB of direct mass storage. System performance ranges from 9 Dhrystone MIPS using a single processor, to 54 Dhrystone MIPS using all six processors.



H.A.D.A./MV

Existing MV/30000 users can upgrade their systems by changing boards. Those with other mid-range DG systems can trade in and retain complete hardware and software compatibility. Pricing for the MV/35000 system starts at $177,000. It will be available beginning this month.

H.A.D.A./MV

The newest member of Data General's growing disk array family, the H.A.D.A./MV, brings to the MV family capabilities that Aviion users already enjoy. It is a 2.5 GB to 30 GB high-performance, transaction-oriented disk array supporting multiple levels of RAID (redundant array of inexpensive disk) technology.

The system provides a high degree of fault tolerance, to virtually eliminate system downtime from disk failure. The H.A.D.A./MV provides concurrent support for RAID 0, RAID 1, and RAID 5 levels, as well as independent drive operation. It is designed for flexible configuration and scalable storage capability, supporting both 1 GB and 500 MB Winchester disk drives. The system can be configured to include industry standard 8 mm, 4 mm DAT, and QIC tape drives for data backup.

Pricing for the H.A.D.A./MV starts at $42,000, available this month.

More on the way

"I don't see it slowing down," said Ellenberger of future MV product development. "We're bringing out new MVs more rapidly than we have at any point in our history. So basically, what that says is you have to move quickly to keep up with technology." Δ

Bits and bytes

Performance not there

From: Jim Bageant

We have recently upgraded our MV/20000 to a Mod 2 and are not getting the increase in performance we expected (in fact, it feels a little slower at the user console). We looked at all three hardware areas before the upgrade (CPU, memory, disk) and decided to upgrade the CPU first. BJ's :PERFMON always reported some free memory and a large LRU (free averaged about 1 MB and LRU 10 MB). Disk drives are old Argus and Fujis (on BMX-3s), but disk wait queues and seek numbers are within BJ's suggested parameters. After the upgrade, free memory is always 0. The LRU chain is still averaging over 10 MB. Why did this occur? Does the second processor partition the memory somehow? Our DG salesperson says we are out of memory, although my tests show swapping is nonexistent. Additional memory is not a large dollar issue, but I don't want to do another upgrade and not have a performance increase visible at the user console. Any ideas?

From: Matt Koch

Sounds to me like your engineer is right. We had a machine that always showed sufficient memory in the LRU chain, but no free memory. Averaging the :PERFMGR reports over 1 minute revealed the problem. When the reports were averaged over 2 minutes, the problem was not apparent. As to why memory has become a problem now, I would think that with the second CPU there are now more active PIDs, increasing the demand for memory.

From: Doug Rady

Your 1 MB free memory was probably consumed by the private data of that second copy of VS now running on your second CPU. If you're down to 0 free with 10 MB of the LRU, then you should look at adding more memory. Always running off the LRU will be slower than if you actually have free memory available, since there is overhead associated with the LRU pages. Δ

Do you have a question, comment, or answer? Call the NADGUG/RDS electronic bulletin board, available to all NADGUG members, 415/499-7628. No fees other than phone charges.

The latest products for DG systems

High-performance PCs

PC

Westboro—Data General Corporation announced two additions to its PC product line.

The Dasher II-386/33L, a high-performance IBM PC-AT compatible featuring the Intel 33 MHz 80386 processor, offers a base configuration including the CPU, 4 MB of memory, a socket for an optional 80387 math coprocessor, and a 64 KB external cache for improved processor performance.

The Dasher II-486SX/20A, based on the 20 MHz Intel 80486SX processor, provides sufficient mass storage and available I/O slots to serve as either a local area network (LAN) server or Unix platform. It may be upgraded to a 33 MHz 80486DX system by switching the CPU chip. Prices for the Dasher II-386/33L begin at $2,445, while the Dasher II-486SX/20A starts at $2,545.

Data General Corporation, 3400 Computer Drive, Westboro, MA 01580; 508/898-4288.

Circle 45 on reader service card.

4 mm DAT tape drive

MV AV

Westboro—Data General has introduced the model 6762 4 mm DAT (digital audio tape) cartridge tape drive for Eclipse/MV and Aviion systems. Using the helical scan recording method, this single-ended SCSI-2 tape drive offers speed and storage capacity needed for backup of large disk subsystems.

A 1 MB onboard data buffer and high-speed search/rewind operations support an average sustainable data transfer rate of 183 KB per second (native mode). High-speed ECC and read-after-write verification are used to ensure data reliability. Priced at $5,500 for an an add-on drive, the model 6762 4 mm DAT tape drive is also available as packaged storage subsystems, starting at $11,000.

Data General Corporation, 3400 Computer Drive, Westboro, MA 01580; 508/898-4246.

Circle 46 on reader service card.

VT emulation

AV PC

Columbia, MD—Rhintek, Inc., introduced its enhanced version of EMU/470, incorporating VT320 and

Continued on page 39

AD INDEX

PRODUCTS AND SERVICES

ON-LINE HELP

***Who to call for answers about* NADGUG *and* FOCUS**

NADGUG address:
c/o Danieli & O'Keefe Associates, Inc.
Chiswick Park
490 Boston Post Rd.
Sudbury, MA 01776
FAX: 508/443-4715

FOCUS Magazine address:
c/o Turnkey Publishing, Inc.
Livingston Building, Suite 250
3420 Executive Center Dr.
Austin, TX 78731
FAX: 512/343-7633

NADGUG

Membership, RIGs, SIGs
NADGUG staff **800/253-3902**
(Outside the U.S.) **508/443-3330**

Electronic bulletin board
(300 or 1200 baud modem)
Rational Data Systems **415/499-7628**

FOCUS Magazine

512/345-5316

Editorial comments, article suggestions .... Doug Johnson
(please send product announcements to the address listed above)

Information about advertising .... Michelle Sentenne

FOCUS back issues .... Turnkey Publishing staff

Products and Services: continued from page 37

VT100 emulation. Version 4.0 provides compatibility with Data General's Aviion workstations and Digital Equipment Corporation's VAX minicomputers. This revision supports six of the most popular network interfaces, including PC/TCP (FTP Software's TCP/IP kernel).

EMU/470 enables users to unite the diverse requirements of PC networks, remote dial-in, and direct connect with minicomputer systems. Users guide operations with EMU's intuitive, multi-tiered menu system. EMU/470 Version 4.0 retails for $249 (foreign orders $269 US).

Rhintek, Inc., 8835 Columbia 100 Parkway, Columbia, MD 21045; 410/730-2575.

Circle 48 on reader service card.

MV-to-PC

MV PC

Rancho Palos Verdes, CA—Now MV users can convert and transfer data to a DIF or ASCII format readable by PCs.

Xport from R.B. Zack & Associates, Inc., works in two steps, in which the user is guided by the utility. The "strip" step allows the user to select the desired content by omitting unwanted information; the "create" step reformats information. Xport is available at an introductory price of $1,995.

R.B. Zack & Associates, Inc., 29000 S. Western Avenue, Suite 401, Rancho Palos Verdes, CA 90732; 310/833-0211.

Circle 49 on reader service card.

New Genisys options

MV

Salt Lake City, UT—A new version of Genisys, the data base management system from DMS Systems, Inc., has added macro options to increase the user's ability to customize applications, allowing branching, looping, and conditional statements.

Other enhancements include form-level aggregate functions to utilize user-defined memory variables, setting a string variable to a given value, and the ability to hot key to the CLI or Wordperfect Office.

DMS Systems, Inc., 1111 Brickyard Road, Salt Lake City, UT 84106; 801/484-3333. Δ

Circle 47 on reader service card.

Data Specific

DG's new chief financial officer

Arthur DeMelle

Arthur W. DeMelle has been appointed vice president and chief financial officer at **Data General.** DeMelle, who brings nearly 30 years of corporate financial experience to his new position, will report to DG President and Chief Executive Officer **Ronald L. Skates.**

"Art DeMelle brings a broad range of corporate financial experience and expertise to Data General," said Skates. "He will be a great addition to our management team."

DeMelle will direct DG's worldwide financial operations, investor relations, and the company's internal information systems management group.

For the past two years, DeMelle has served as senior vice president of finance and administration at a privately held company, Chep USA of Park Ridge, New Jersey. Prior to that, he was executive vice president and chief financial officer at Emery Air Freight for two years; executive vice president of finance at Purolator Courier Corporation from 1980 to 1987; and vice president of finance at Interpace Corporation from 1978 to 1980.

From 1963 to 1978, DeMelle worked at Price Waterhouse & Company in Newark and Morristown, New Jersey, where he was an audit partner. DeMelle holds an MBA degree from Rutgers University, and a bachelor's degree in economics from Bowdoin College.

A large family

In the dozen years since Data General introduced the MV/8000 system, marking the company's entry into the 32-bit computing world, more than 50 different MV class machines have been brought out to augment the product line. This vast array was depicted quite strikingly in a wide, wide timeline diagram across the bottom of several pages in *Data General Universe*, the Eclipse Business Unit's quarterly publication.

The timeline shows a proliferating number of models through the 1980s, filling an ever wider variety of computing niches, culminating with the most recently announced MV/35000-1 through 4 and MV/60000 HA-1 through HA-4. At the far end beyond 1992 stand several more, intriguingly labeled, "Future." No doubt more family additions are on the way.

Workforce reduction

Data General announced April 1 that it will record a charge of approximately $46 million against its second quarter, which ended March 28, as a result of costs associated with a worldwide workforce reduction of about 1,000.

The company also said that while revenues for its Aviion family continue to be strong, preliminary indications are that total second-quarter revenues will be lower than what most industry analysts have projected. This is due to weakness in DG's other product lines, competitive pricing pressures, and a weak worldwide economy. The company said it may report an operating loss for the quarter, prior to the restructuring charge.

DG reports that its workforce reduction will include a significant number of administrative positions being eliminated from international operations.

The workforce reduction "is a painful but necessary step," said Ronald L. Skates, president and chief executive officer. "This action will make Data General an even more efficient producer of high-quality products, and will not impact our ability to provide service to our customers."

DG's financial position continues to be strong, said Skates. The company had cash and marketable securities of $249 million at the end of the first quarter. DG employed 8,100 at the end of that quarter, which ended December 28, 1991. Δ

6 great reasons to dash to Data General!

1. DASHER II–386/33L™

2. DASHER II–486/33LE™

3. DASHER II–386SX/20A™

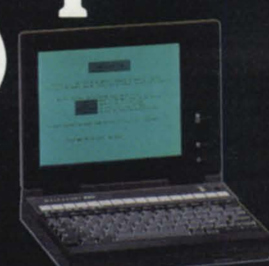
4. DASHER II–486SX/20A™

5. DASHER II–386SX/16z™

6. DASHER II–486/33TE™

intel inside™

(And 1 super reason to walk.)



WALKABOUT®/320

The six Data General DASHER® PC models and the Data General WALKABOUT/320 Notebook PC are all you need to run in a variety of application environments. They are all IBM compatible using the latest Intel® technology at prices that leave the competition in the dust! Our total lineup means a single-source total solution: Computer systems. PCs. Software. Installation. And *ON-SITE* service by the same people who support your existing DG systems. Data General PCs are fully Novell®-certified. We even provide a 1-year warranty on every model. If you want to win the PC productivity race, then dash, walk, whatever, just get to your phone and call **1-800-DATA GEN**

Data General
Where the World is going!

Circle 11 on reader service card.

DataGeneral
TLC