

DOWN\$IZING

Downsizing—moving from a few large computers to many smaller platforms—is a hot topic. More than most other IS technologies, downsizing has the reality, immediacy, and statistics to be a “silver bullet.” All the major benefits, from cost reduction to faster systems development and better running delivered systems, are realized by using downsizing techniques.

Perhaps the most obvious and believable benefit of downsizing is an enormous re-

duction in hardware costs. The cost for a million instructions per second (MIPS) on small platforms is \$200 to \$700 per MIPS, compared to over \$100,000 per MIPS on large mainframes. If downsizing allows a shop to replace a \$2 million mainframe with half a dozen \$5000 to \$10,000 PC-type workstations, a lot of hardware money can be saved.

Along with cheaper hardware comes software that is priced accordingly. If you price the software that is needed for a modern distributed network of workstations, you’ll find it significantly

cheaper than comparable capability on minis or mainframes. A new, lower budget will still allow you to get microcomputer and LAN operating systems, a graphical interface, and a fancy-looking, windows-based 4GL with companion relational DBMS.

Of course, by moving to PC and workstation platforms, one is moving right into the heart of “open systems” territory. Because of the large number of suppliers and new technologies, the user who commits to downsized platforms will enjoy a significant degree of vendor indepen-

© 1991 by George Schussel. All rights reserved.

A REVIEW OF THE ENABLING TECHNOLOGIES

by George Schussel

dence. Price and service competition are plentiful.

Savings in people and administration costs are a little tougher to document but are no less real. At recent conferences, spokespeople from several major corporations—including Texaco, Mobil, Hyatt, Echlin, and Harris—have recounted experiences where budgets for IS systems or departments were radically reduced after implementation of downsized systems. Of particular interest to those who specialize in application development, the building of new systems on PCs and worksta-

tions seems to be about twice as efficient as building the same applications on time-shared mainframe or minicomputer terminals.

For example, application developers enjoy the immediate turnaround possible from developing and debugging on PCs. Another improvement is the tools and environment advantage afforded by PC/workstation platforms. Most people find the idea of debugging with a multitasking, windows-based platform far friendlier than using a time-shared terminal. A developer can run the program source code in one window, a

debugger in a second window, and can watch the program output in a third window, while stepping through the program a line at a time.

OPEN SYSTEMS

Open systems are those operating systems that are freely licensed to any and all systems builders. In today's world, that means DOS, Windows 3.0 and OS/2 from Microsoft, and Unix from a variety of sources, including AT&T's USO, the Open Software Foundation, and The Santa Cruz Operation.



In the “old” days, computers were proprietary designs with customized, internally built operating systems. Attempts at standardization or portability of applications focused on computer languages such as FORTRAN and COBOL. Real-world portability, of course, was limited by serious differences in underlying architectures and operating system calls and services.

In the new downsized world, open systems are redefining the rules. Computer architectures are open, defined by merchant semiconductor chip manufacturers—such as Intel, Motorola, and Sun—who will sell to anyone. These firms hook up with open systems operating systems developers, such as Microsoft or The Santa Cruz Operation, and sell the CPU and associated operating system to systems integrators like IBM, DEC, Compaq, and a host of smaller competitors. The sales volume made possible by this approach far exceeds what was possible through the mechanism of proprietary approaches, even by IBM. Old-line manufacturers like IBM and DEC find themselves selling their older proprietary architectures next to new, open systems that offer price-performance that is typically 10 times better!

Since these open systems have (so far) been biased toward smaller systems, the term “downsizing” has been used to describe this greatly improved economics. However, the year-to-year performance improve-

ments of microprocessors are so much greater than for older traditional architectures that the new merchant microprocessor-based platforms are now rivaling the processing power of current minicomputers and mainframes. It’s clear that by 1995 microprocessor-based systems will—in terms of raw power—exceed what can be provided by traditional architectures like DEC’s VAX and IBM’s System 390.

What makes all this power usable, however, is the software. It’s the combination of multitasking, protected operating systems with industrial-strength database management systems and widely connecting communications software that allows us to use this new generation of hardware in combination with networks to produce robust systems that can replace traditional mainframes.

The two most important events leading up to the concept of open operating systems have been the creation of Microsoft, the world’s largest and most profitable software product company, and the fact that various openly licensed versions of Unix now run on every multiuser computer system sold in the United States, with the solitary exception of the IBM AS/400.

IBM AND MICROSOFT: THE OPEN SYSTEMS COMMITMENT

Microsoft was already on its way to stardom when IBM picked it as a partner to de-

velop an operating system for the IBM PC. A fundamental and critical decision that IBM made in creating the agreement for the development of MS-DOS was that Microsoft would license this operating system to any and all other systems vendors—in effect creating an open operating system and the clone business.

Blessed by the IBM endorsement, Microsoft profited from the sale of the DOS standard in over 50 million PC clones sold during the 1980s. The original joint development agreement (JDA) between IBM and Microsoft was expanded to include several efforts (Presentation Manager, LAN Manager, OS/2).

The formula of cooperation between IBM and Microsoft has proven spectacularly successful in dominating the market for PC technology. The essence of this approach has involved cooperative development of technology by the two giants, commercialization of proprietary products by IBM, and distribution of the same technology to clone vendors through Microsoft.

Now, however, things are changing rapidly in this partnership, including the fact that the two companies are supporting different technology approaches in several areas. In particular, Microsoft’s spectacular sales success with Windows 3.0 has directly challenged IBM’s attempt to make the partners’ OS/2 and Presentation Manager products the next desktop standard. Micro-

soft is the sole owner of Windows 3.0, which by this spring had sold over 6 million copies. Microsoft and IBM jointly own OS/2 and Presentation Manager, which by the same date had sold about 600,000 copies. Each will offer support for the other's standard bearer, but a definite divergence in strategic direction has been set.

Another split in strategic approaches by the two partners is evident in their approaches to the LAN operating system market. In an attempt to recreate their success with MS-DOS, IBM and Microsoft established a joint project to develop LAN Manager, an extension to OS/2 that converts OS/2's single-user, multitasking operating systems into a multiuser, multitasking operating system that can support client/server computing over a network.

The spectacular market success of the earlier DOS endeavor has so far totally eluded LAN Manager. There are many business and technical reasons for this relative failure, chief among them being

- ★ Novell, a well-established competitor, has been selling a product that has (so far anyway) a better technical reputation than Microsoft's.
- ★ The slow sales of the underlying OS/2 platform have meant that the potential market for LAN Manager is much smaller than for Novell's products.

In any case, until the spring of 1991, Novell appeared to be Microsoft's and IBM's mortal enemy and competitor in the networking business. At Boston's spring Networkworld show, however, IBM announced that it had agreed with Novell to remarket Novell's NetWare product and to build connectivity bridges to IBM's version of LAN Manager (LAN Server). Microsoft was noticeably absent from the announcement.

I doubt that a divorce is in the offing, but more fights are very likely. The IBM/Microsoft relationship has been fundamental to the emergence of the downsizing trend and is guaranteed to provide ongoing entertainment.

THE ROLE FOR UNIX

In the hierarchy of open operating systems, there is a natural progression from small to large: DOS, Windows 3.0, OS/2, and Unix. DOS, of course, is the low-end, 640K-limited, character-oriented environment that will stick around primarily because of the large existing base of 80286 computers that may be too slow to run graphical user interfaces effectively.

The logical upgrade for people who want to stay with the underlying DOS platform but want the advantages of a graphical interface with task switching for different applications is Windows 3.0. Win-3 is appropriate now for running the client side of client/server applications. In the future, according to Microsoft, it will

evolve to acquire many of OS/2's multitasking capabilities and will also be appropriate for server applications (which are multiuser in nature). While a single production application running in a Win-3 client is a reasonable goal, remember that Win-3's multitasking is unprotected. This means that a failure in the software running any application is likely to take down the entire machine.

The role for OS/2 has been championed by IBM and, to a lesser extent, Microsoft. This role is now defined by IBM to be the standard for both client and server when the machines are Intel architecture (80386, 80486). Sales of OS/2 have not lived up to the expectations of its progenitors because most users have failed to see the benefits that they would receive after investing in the hardware, software, and staff training necessary for an upgrade to OS/2.

With the availability of Win-3, the role for OS/2 has become much clearer. The protected, multitasking environment that it offers and its compatibility with DOS make OS/2 an outstanding development environment for PC- and/or mainframe-based applications. In addition, OS/2's capabilities are fine for hosting the server and communications portions of a database client/server application.

So what is left for Unix? In fact, Unix represents both a partner for the Win-3, DOS, and OS/2 environments and a complete alternative. Unix



with the Open Look or Motif graphical interfaces is an appropriate client-side operating system. And certainly Unix as a mature time-sharing, multitasking operating system is the closest open environment to mature proprietary operating systems and makes a lot of sense on your server.

Unix is the logical server upgrade path for users who have decided on client/server approaches and have run out of capacity on OS/2 running on Intel CISC (80386, 80486, etc.) microprocessors. If a product like Oracle Server or Sybase's SQL Server was used for the DBMS, the applications will not change as the OS/2 box is removed and replaced with a Unix box. Unix server manufacturers, such as NCR, Pyramid, Sequent, Corollary, and Netframe, are selling machines that run multiprocessing versions of Unix and that can deliver hundreds of MIPS of computer power at a price that is about 5 percent of IBM's mainframes.

Historically, Unix has been criticized as being an "academic" product, suitable for use in universities but not robust enough for industrial-strength, high-uptime commercial transaction processing. While this may have been true a few years ago, nothing could be farther from the truth today. Recent additions to Unix capability have included high-level security, fault tolerance, multiprocessing, and support for transactions across a network.

Many markets are standardizing on Unix. The U.S. gov-

ernment, for example, now requires Unix in all new procurements. Other markets that are settling on a Unix standard include workstations and scientific computers.

Since both Unix and OS/2 are suitable for open system servers, it is interesting to compare the two. OS/2 is probably the better choice for shops that are wedded to IBM technologies. Although IBM is aggressively selling AIX, its version of Unix, OS/2 is the downsized operating system that is part of IBM's SAA, the set of portability and connectivity standards for IBM users. Database, CASE, and connectivity solutions will be delivered to OS/2 customers years before the same features appear in AIX.

Unix's advantage over OS/2 is far greater scalability. The smallest computer that will run OS/2 is one built on an 80386 chip, and the largest (available today) is one made with an 80486 chip. Unix, on the other hand, will not only run on these same Intel architectures, but will also run on much larger computers up to and including supercomputers. If you want scalability in the OS/2 world, you have to develop to IBM's SAA standards and port up to a proprietary IBM environment such as VM or MVS.

One other issue of concern to some developers is the fact that, based on "apples to apples" benchmark comparisons of database and compute-type applications, Unix appears to offer about twice the through-

put on the same server as OS/2 does.

LAN OPERATING SYSTEMS

The architecture behind PC and workstation networks is different from the star configuration of typical terminal-to-mainframe situations. In the mainframe world, the central computer is constantly connected to terminals and responds to messages from them. Terminals can communicate with each other only by sending a message to the mainframe, which switches it to the appropriate target.

Communication in the downsized world is more typically "peer to peer," with computers talking to each other rather than just terminals displaying mini or mainframe messages. The network in a downsized world may be star wired to a server (as in the mainframe world), or it can be ring or bus wired. In the ring topology the network physically goes through each device (such as a PC), while in the bus topology each hardware device hangs off a backbone wire that acts as the communication medium.

Since PCs emerged as single-user devices, the question has arisen about how to manage, from a software point of view, a network of single-user devices operating in concert as a multitasking, multiuser creation. The answer has been to hang another server on the ring and provide network services to the other users through software known as a LAN operating system.

In effect, the combination of single-user operating systems running on clients, a LAN network, and a LAN operating system can allow the network and its constituents to emulate the functionality of a star-wired mainframe computer. Of course, we're using the word "emulate" loosely here because, while the network will handle the transaction work load of the mainframe, it will do it with a total hardware/software cost that is far, far less. And the network will supply graphical interfaces, run Lotus 1-2-3 and Word for Windows, and do a host (pun intended) of things that the mainframe can't handle.

As a further guide to the functions of a LAN operating system, the following services are provided:

- ★ Remote administration of the server
- ★ Access to multiple servers
- ★ Interprocess communication across the LAN
- ★ Security through passwords and other devices
- ★ Performance monitoring
- ★ Audit trails
- ★ Accounting for network and resource usage
- ★ Assignment of tasks to idle workstations

The three most widely sold product sets for LAN operating

systems are supplied by Novell, IBM/Microsoft, and Banyan. Novell's NetWare, with over 50 percent of the total market, has been the winner so far. Novell has targeted the office systems market that originally came about because of a desire to share files, printers, and the like. Novell supplied a quality product and one that used minimal network resources. While still dominant, it is now coming under attack from Microsoft and a host of Microsoft VARs (the largest of which is IBM).

NetWare is vulnerable because, as an office support product, it wasn't designed for the types of robust database applications that are necessary to replace minis and mainframes. In particular, NetWare's structure runs applications in an "unprotected" environment. This means that everything is running in a single partition and a failure in one program is likely to take down the whole network. Worse, debugging such an environment is hellish because it may be impossible to reproduce the conditions at the time of failure exactly.

This opens a window for Microsoft and IBM to take away market share with their LAN Manager/LAN Server products. These products run on top of OS/2 and take advantage of the protected, multi-tasking environment that OS/2 provides. The problem with these technologies to date has been that, other than the protected environment, they have offered no advantages over NetWare. In fact, just the op-

posite has been the case: most users have felt that NetWare is more mature, easier to use, more reliable, and faster than LAN Manager.

The third major player is Banyan. Its product, Vines, is based on Unix in much the same way as LAN Manager/Server is based on OS/2. Banyan has carved out the high ground in this fray by offering a higher level of management services for networks than its competitors. This is especially true for companies that wish to run wide area networks with many local area drops. Banyan takes advantage of Unix's built-in connectivity features to interface to the largest number of foreign environments. To date, Banyan's disadvantage has been a benchmark record that shows it processing transactions decidedly slower than its competitors.

Whatever LAN operating system is chosen, your downsizing architect will have to integrate it carefully with a choice of platforms, operating systems, and client/server database management systems.

CLIENT/SERVER COMPUTING

The most critical technology for downsizing applications is SQL-based client/server computing. Once you decide to re-think your applications in the context of workstations using shared databases located on servers and connected by networks, you've made the essential decision to build an applications architecture that will be



economical, flexible, and portable for a long time into the future.

A client/server computing environment consists of three principal components: client, server, and network (see Figure 1).

The client is where the application program runs. Normally, the client hardware is a desktop computer such as an IBM PC or a clone or an Apple Mac. The application program itself may be written in a 4GL or in a 3GL such as C or COBOL. There is a whole new group of "Windows 4GLs" that allows painting of applications under the leading desktop windows-based operating systems.

These products support both windows-oriented appli-

cation development and execution. Leading examples include Powersoft's Powerbuilder, Ingres's Windows 4GL, and Gupta's SQLWindows. Using any of these application-building approaches results in a runtime configuration in which the I/O and application control come from the client, while the database and associated semantics run on the server. At the desktop level, most software will support the emerging windows-based standards. This means Windows 3.0 for DOS, Presentation Manager and Windows 3.0 for OS/2, and Open Look and Motif for Unix.

The server is responsible for executing the SQL statement received from the client. Sometimes the data request is not

pure SQL, but it can be a remote procedure call that would then trigger a series of already compiled existing SQL statements on the server.

The server is responsible for SQL optimization, determining the best path to the data, and managing transactions. Some server technologies support advanced software capabilities such as stored procedures, event notifiers, and triggers. The server is also responsible for data security and validation of the requestor, and it also handles additional database functions such as concurrency management, deadlock protection and resolution, logging and recovering, and database creation and definition.

The network is responsible for connecting client and

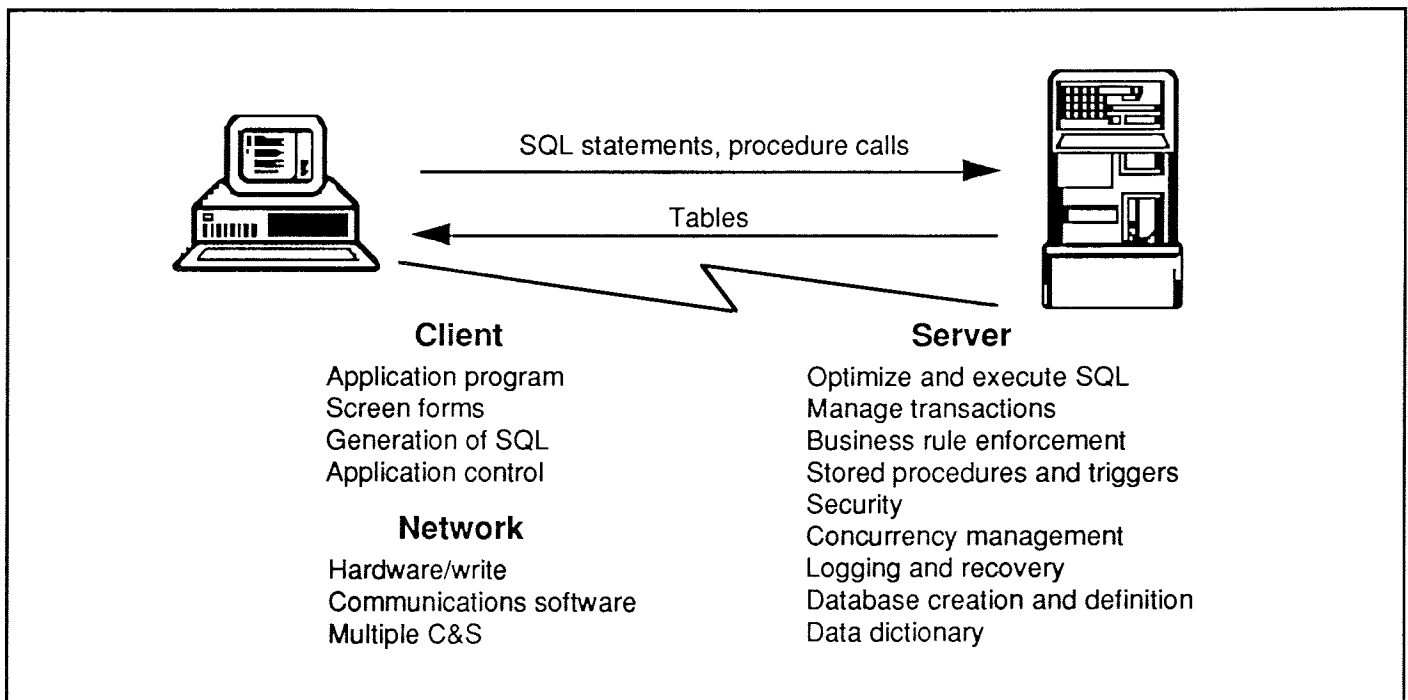


Figure 1: Client/server functions

server. Normally, the network consists of some kind of wire along with the communications card in both the client and server boxes. The communications software typically handles different types of communication standards such as LU6.2 and TCP/IP. Most network environments provide support for multiple clients and servers.

An important benefit that the set-oriented SQL language provides is network efficiency. When using traditional file serving PC LAN approaches, the entire data file must be transmitted across a network to the client machine. Using SQL as a basis for database management solves this problem since only the necessary query response data (a table) is transmitted to the client machine.

SQL on the server also allows the implementation of advanced facilities such as triggers and automatic procedures in the database. As relational DBMSs evolve, they will provide the ability to build rules directly into the database engine. The systems that are built with this approach will be more robust than traditional application-based logic approaches.

Although client/server computing is being planned for environments that use minicomputers and mainframes as the server, the largest market is likely to develop with a mix of OS/2, Windows 3.0, and MS-DOS on the client and either Unix or OS/2 as the base for the server. OS/2- or Unix-based SQL server software will

provide mainframe-level security, recovery, and data integrity capability. Functions such as automatic locking and commit rollback logic along with deadlock detection and a full suite of data administration utilities are available on the server side. Another way of looking at this is that SQL client/server technology allows cheap PCs to become "industrial-strength" computing engines.

PERFORMANCE

If you have had a chance to build PC-based database applications in the last few years, you may be suspicious of any claim that a PC hardware environment could be capable of performing on a level comparable to minicomputer technology. However, it is important to remember that the processing capability of a typical PC has increased by a factor of 10 from 1984 to 1990. A PC

built around the Intel 80386 microprocessor chip running at 33 MHz has 30 times the computing power found in a PC/XT.

Benchmarks audited by Digital Consulting have shown that a 80386-based PC can handle about 10 TPC-A (Transaction Processing Council, database Benchmark A) transactions per second while running under OS/2. This level of service can provide on-line transaction processing capability at a cost of \$5000 per transaction per second (TPS). This cost is much less per TPS than existing minicomputer and mainframe systems can provide. Using proprietary minicomputers, you can expect to spend from \$25,000 to \$40,000 per TPS. IMS-based MVS mainframe environments typically yield a cost of \$50,000 to \$75,000 per TPS. Alternatively, using the combination of MVS and DB2 as a transac-

DCI ANNOUNCES FIRST DOWNSIZING CONFERENCE AND EXPO

A major trade show and conference focusing on downsizing trends will be held in Los Angeles on September 10-12. The DCI Downsizing Expo will have a very different focus from Comdex and PC Expo, two existing shows that cover PC-oriented products. The Downsizing Expo will focus on education and products for the corporate, educational, and government user. The leading consultants in this emerging field will be in Los Angeles to discuss trends and issues. Management issues related to downsizing computer systems will be stressed. Seminar sessions will be held by Cheryl Currid, Larry Deboever, Rich Finckelstein, George Schussel, Will Zachmann, and over four dozen other industry leaders. The expo floor will feature over 100 exhibits by the leading server, PC, workstation, DBMS, CASE, and tool vendors. For more information on this conference, you can call DCI at 508/470-3880.



tion processing engine will typically result in a cost of \$125,000 per TPS.

What all this means is that based upon full development, maintenance, hardware, software, and staff costs, SQL client/server computing is likely to result in finished systems that cost only a fraction of what building transaction systems has cost in the past.

Of course, there are many applications that are simply too large to contemplate running on even a fast PC. Client/server architecture allows you to design the application once and, without change, port that application to whatever hardware server has the database processing power to manage your database. This fact allows development on PC-style servers and then porting to the new generation of "superservers," minicomputers built to run open operating systems and powered by multiprocessing versions of merchant CPU chips.

The approach is to take microprocessor-based technologies and combine them with high-speed buses, channels, and parallel computing architectures to create platforms that can run with the fastest minicomputers. Vendors such as Compaq, Pyramid, and Sequent are building parallel processing machines using CICS or RISC microprocessor units that are capable of reaching a sustained processing capability of hundreds of MIPS. Don't be surprised to see a combina-

tion of these new hardware systems with software companies like Microsoft and Oracle providing computing technologies comparable to IBM's largest machines but at a tiny fraction of the price.

But client/server computing does not need to be relegated to the low end of the transaction processing spectrum. It is very reasonable to think of products like Oracle and Sybase in combination with middle-range or high-end superservers from companies such as Solbourne, Pyramid, Concurrent, Compaq, IBM, or DEC. This high-end super-server hardware is typically going to be built with parallel Intel 386, 486, and/or RISC chips from MIPS Computer or Sun Microsystems. By configuring a server with a multiprocessor design and an open operating system that supports it (e.g., Unix, Vines, OS/2, and LAN Manager), a vendor can build a machine with hundreds of MIPS of processing power and 250 gigabytes of disk data storage at a cost of well under \$500,000. Combining this technology with SCSI or other high-speed channels and a client/server DBMS allows a configuration of new technology hardware and database server to be considered as a replacement for a \$14 million IBM System 390 running DB2. With a potential savings of almost 95 percent, this would appear to be an offer well worth considering for many situations.

ADVANTAGES OF CLIENT/ SERVER COMPUTING

Besides saving money on the hardware/software platforms, client/server computing offers a whole series of important user benefits:

- ★ Developers can use PCs instead of time-share terminals as a primary development platform.
- ★ Even though the PC is used as a principal platform, security, integrity, and recovery capability comparable to minicomputers is the result.
- ★ The efficiency of query and transmission of the SQL language greatly reduces the network communication load.
- ★ Gateway technologies, which are an important component of client/server computing software, will allow PC users to gain access to data located in mainframe and minicomputer DBMS products such as DB2, IMS, and Rdb.
- ★ The client/server model isolates the data from the applications program in the design stage. This allows a greater amount of flexibility in managing and expanding the database and also in adding new programs at the application level.

- ★ The client/server model is very scalable because, as requirements for more processing come up, more servers can be added in the network, or servers can be traded up to the latest generation of microprocessor.
- ★ A lot of flexibility comes from a computing environment based upon SQL, because SQL has been almost universally adopted as a standard. Commitment to an SQL server engine will mean that most front-end 4GL, spreadsheet, word processing, and graphics tools will be interfaced to the SQL engine.

Client/server computing provides the robust security, integrity, and database capabilities of minicomputer or mainframe architectures while allowing companies to build and run their applications on PC and minicomputer networks. This hardware/software combination can cut 90 percent of the costs of the hardware/software environment for building "industrial-strength" applications.

For many typical business applications, the server can be a powerful PC or minicomputer running multiuser, multitasking server software built on top of operating systems such as Unix, VMS, and OS/2. The client is a smaller but still powerful PC that has the power of running applications. Although mainframe vendors discuss the concept of using a large mainframe

such as a VAX 9000 or ES/9000 as a database server to networks, it is just "slide ware" so far. For these larger machines to play a role in future networks, it is clear that they will have to acquire server functionality by acquiring and supporting emerging downsize standards such as Unix and LAN Manager.

The transaction capabilities of client/server software working with lower-end PC servers or superservers (minicomputer-style cabinets built with merchant microprocessors such as the 80486 or R4000) are quite astounding. For example, on the low end of the hardware scale, both Gupta's SQLBase and Microsoft's SQL Server can run on Intel's 80386-based PCs processing approximately 10 TPC-A transactions per second. PC hardware can support disks with 16-msec access time and 2M- to 3M-byte transfer rates. Such a machine can be configured with 300 megabytes of disk for under \$10,000. In case you're not familiar with the TPC-A benchmark, it should be pointed out that a rate of 10 transactions per second is adequate to support 250 automated teller machines on a single server.

CONCLUSION

Caveat emptor: The combination of client/server and downsizing sounds like a technology almost too good to be true. The careful buyer will know that there are many pitfalls on

this road. The following cautions are provided to assist in safely migrating to the new technology.

- ★ Remember that client/server computing across networks is new technology and along with this newness comes an open approach. This openness means that vendors are building product interfaces between networks, database management systems, and application development tools from different vendors. The progress in making all this technology work together is slow. As you begin to use it, be sure to take one step at a time and test the combinations of products you plan to use carefully.
- ★ Because software products to support client/server computing are new, there are not many finished applications that can be purchased. At this point, you're going to have to write your own applications.
- ★ You'll need the support of top management to succeed with a downsizing effort. Many groups are likely to oppose the movement to smaller platforms. First of all, a downsizing effort will have a drastic impact on IS management as the same jobs can be completed with radically



lower budgets. Second, the dominant hardware vendor at your site will quickly realize that downsizing means sharply lower income for hardware-related items. Your vendor is likely to offer many reasons as to why a downsizing approach won't work.

WHAT DO YOU DO NEXT?

One of the nicest things about downsizing with client/server computing is that it is a reasonably easy technology to migrate toward. It doesn't require you to throw away all your current investment in systems. I advise clients to look at the additional functionality in client/server computing that can extend and complement existing systems.

For example, as a first application you could choose to employ a client/server computing model for decision support. In this role, the decision support system would use read-only capability against data located in your minicomputer or mainframe database. The offloading of significant computing cycles for the decision support application will bring important savings. At a later date you can implement plans for true cooperative processing or transaction-based applications which would have otherwise gone on your mainframe.

Downsizing and client/server computing are technologies that will be with us throughout this decade. Companies that commit to them

now will enjoy many years of cost-benefit and performance advantages.

George Schussel, the guest editor of this issue, is president of Digital Consulting. He is the chairperson of the Database World Conference and the DCI Downsizing Expo. A 20-year career in consulting and teaching on database and PC issues has prepared him to be a leader in the trend toward downsizing computer systems.

He can be reached at Digital Consulting, 204 Andover St., Andover, MA 01810 (508/470-3870; fax 508/470-0526). ★