Schussel's
DOWNSIZING JOURNAL

November 1992

HOST BUSTERS!
A Visit to Pacific IBM Employees Federal Credit Union

On my way to deliver the keynote at August's DOWNSIZING EXPO, I visited Pacific IBM Employees Federal Credit Union (PACIBM) in San Jose, California. This fascinating visit to PACIBM, a mid-sized, full-service bank, gave me the opportunity to examine a wildly successful example of downsizing that violated most of the rules offered by some of today's leading downsizing consultants. The rules that PACIBM had violated were:

☑ Don't start a downsizing project with your toughest application. Pick a simpler pilot for learning purposes.

Migrating DB2 Applications Part II

by Howard Fosdick

This is the second article in a two-part series on migrating and downsizing DB2 applications to PCs and PC LAN environments. In the first article, which was featured in the September 1992 issue of *Schussel's Downsizing Journal*, Fosdick discussed five popular approaches to downsizing: 1. moving from a mainframe to a PC, 2. moving from a mainframe to a PC LAN, 3. adding GUI front-ends to mainframe applications, 4. cross-platform application development, 5. architectural evolution where PCs are gradually added to the mainframe. In this month's

Downsizing DB2...

article, Fosdick discusses some of the compatibility problems that are encountered when trying to port legacy applications to new, downsized environments.

Software Compatibility Issues

Both software and SQL compatibility play large roles in several downsizing scenarios. In a total replacement situation in which the IS department is moving all applications to either a PC or a PC LAN, if the software isn't extremely compatible, you're going to end up rewriting all of your applications. The bottom line is this: the older your IS shop is and the more legacy systems you have, the more important compatibility issues between applications become. When downsizing applications from a mainframe to PCs, there are several items that need to be examined: program logic, screen handling techniques, and the database. These turn out to be three different forms of software that you have to deal with when downsizing.

SQL compatibility

When downsizing within IBM environments, there are several areas where software compatibility must be examined: SQL compatibility, operational differences, application development and support environments.
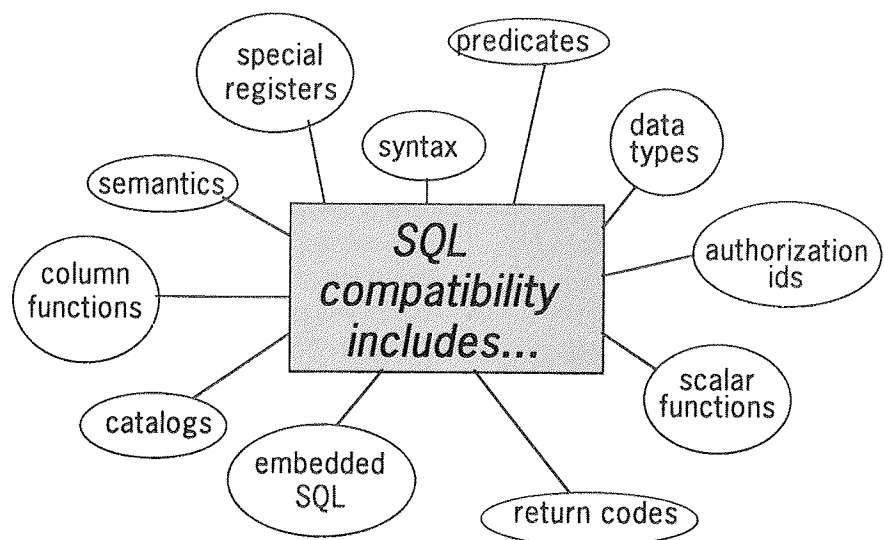
How has IBM attempted to create compatibility between their various environments? For starters, they have 15 SAA manuals in which they list every SQL statement for all of their operating environments. However, though compatibility between systems appears from these manuals to be excellent, when you actually starting working within these environments, you will quickly discover the truth. If you were to actually obtain manuals for each DBMS, and compare SQL statements, you would find that there are many omissions and inconsistencies between systems. One reason for these discrepancies is environmental differences: some statements that make sense in DB2 aren't applicable to OS/2 Database Manager. Similarly, there are keywords or phrases that make sense on a mainframe, but not in a PC architecture. This subject is far more complex than just asking

a vendor, 'do you support this or that SQL statement?' Insuring an acceptable level of compatibility can be a very complicated exercise for the user with whom the burden lies.

So, what is the bottom line concerning SQL compatibility? Comparing the different versions for DB2 and OS/2 Database Manager, the data manipulation language is primarily the same. You can get a lot of mileage out of this similarity. When you look at a DDL, you'll find that the logical object is the same, but that physical objects are treated very differently. On a mainframe and a PC, the method in which data maps onto the disk drive is very different.

Vendors tend to key on SQL because there is an ANSI standard. Therefore, products can be conformed to meet these standards. This is a fairly, easy thing to do. However, in this process,

special registers

predicates

semantics

syntax

data types

SQL compatibility includes...

column functions

authorization ids

catalogs

embedded SQL

scalar functions

return codes

operational and environmental differences between the products tend to be ignored. Just a few of the "details" that tend to be overlooked include: catalogs, limits, security, locking, isolation levels, concurrency, logging, utilities, and referential integrity. These are DBMS internals that become very important when downsizing applications. This is why most companies decide to downsize decision support, read-only applications first, and then progressively downsize more ambitious applications.

Physical objects

I would like to give you one, quick example of the discrepancies between physical objects: DB2 employs objects like table spaces, and index spaces while OS/2 Database Manager basically doesn't have physical objects. Instead, OS/2 Database Manager has something called a database that uses rules about mapping logical objects. To the extent that your application cares about these physical dependencies, you've got a major concern. If applications are transparent in terms of physical objects, you will have a better ability to downsize. SQL/400 looks at physical objects in an entirely different way then does SQL DS. So, even within the IBM family of products, you can find some major incompatibility problems.

Utilities

I also want to mention utility compatibility. Most people are familiar with DB2 utilities. What are their equivalents in a downsized environment? For example: how are backup and recovery handled in the LAN DBMS environment? One answer is that there exist very different types of utilities available within PC architectures. You could, for example, downsize a COBOL CICS application with certain products and have a very successful downsizing effort in terms of the code. But what about backup and recovery? To my knowledge, not one vendor can currently offer a package that will also downsize and port those DB2 utilities onto the PC. This is a big concern.

SAA benefits for the IS world to date

How far has the IS world moved in terms of the benefits provided by SAA? This world has improved greatly. SQL DML and logical DDL have high degrees of compatibility. In terms of programming languages, procedure languages, and application generators, there is a high software compatibility. The things to look out for are physical DDL, or DCL that still have differences between platforms. As I mentioned, when you get into operating system dependencies, issues such as security and teleprocessing interfaces make downsizing more difficult. When you look at an application in terms of portability or downsizability, in this narrow sense, you can achieve significant results with the appropriate downsizing tools. As you broaden your scope to include issue such as development, maintenance, and support environments, downsizing becomes much more of a burden.

SAA has given greater benefits to high-level users. If you have a similar

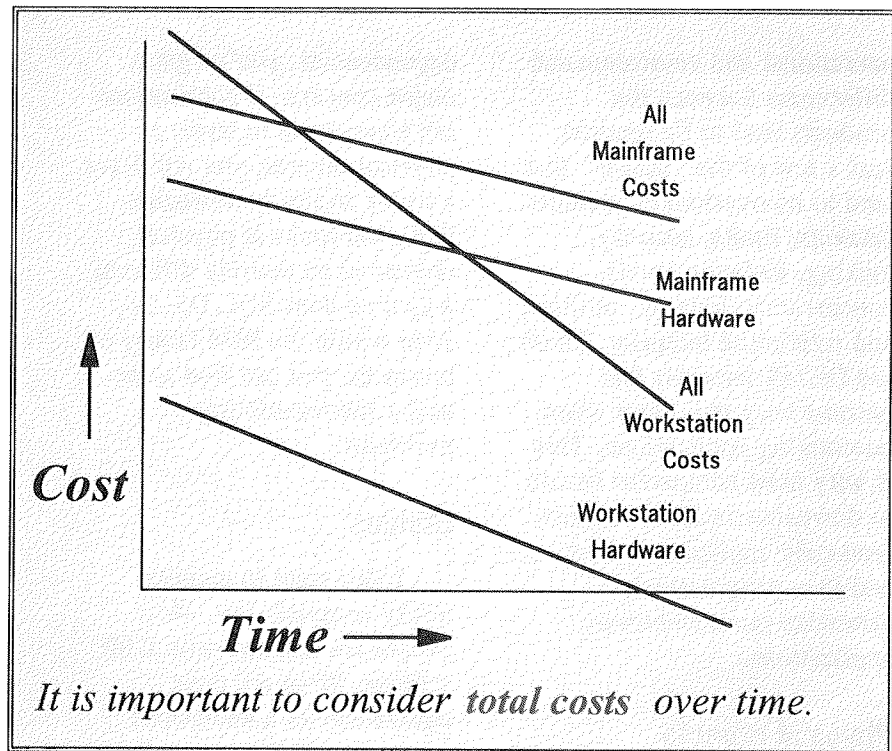|	High Compatibility between DB2 and downsized software	Low Compatibility between DB2 and downsized software
SQL	DML logical DDL	physical DDL DCL
Environment	SAA programming languages procedural languages application generators	TP monitors O/S control languages security
O/S		backup, recovery concurrency issues maintenance issues

Downsizing DB2...

(continued from previous page)

SQL on your LAN and mainframe for queries that programmers use (such as DB2) you will receive a great deal of benefit. But, as you progress into the lower levels of software that include operational issues, database management, or technical support, you will receive less advantage. This is where the cost of downsizing is currently increasing.

The complete costs of downsizing–knowing to look beyond hardware

If you don't downsize all of your DB2 applications concurrently, a situation will emerge where you must maintain new PC platforms in addition to the mainframe. The shops I've seen have relatively good success in downsizing, have taken incremental approaches. This means that very often, for short amounts of time, the shops end-up maintaining both systems with several applications running in parallel. Or, more commonly, there will be part of an application on the PC LAN with other parts still running on the mainframe. Operating in such a manner accounts for part of the cost of downsizing. This is a good example of why looking solely at hardware costs isn't an accurate measure of the total downsizing costs.

It is important to consider total costs over time.

While the cost of the hardware for your downsized systems may be inexpensive, you have to support both the old and new environments. The administrative and training costs alone can absorb a large portion of your savings.

Conclusions: downsizing is an IS management challenge not a technical issue

Analysts right now are talking about utopia. They have been primarily looking at the hardware costs involved in downsizing. The result is the party line that downsizing will save you money instantly since workstations cost significantly less than mainframes, and because of advances in hardware, workstation costs are dropping at a faster rate–a differential that is only going to grow over time. But, being

an IS practitioner, I can tell you that there are other costs involved in downsizing. In fact, you may find that initial costs after starting a downsizing project, are higher than current expenditures. Additional costs will appears from all areas including systems running in parallel, software conversions, training, new software and hardware purchases, and organizational issues. In the short term, IS costs are likely to increase. However, don't be discouraged: in the long term, you are doing the right thing because your company will reap a multitude of benefits from the shift in an evolutionary manner to smaller, more cost-effective software platforms. In the long term, the operating costs will decrease.

What I want to stress is that many analysts leave the staff and software costs out of the downsizing picture. Personnel costs include re-training and re-organization, defining new standards and procedures, and designing and enacting complicated scenarios for transition. Most of these costs are management issues.

Currently, we are in the early evolutionary phase (see chart below) of downsizing. Some of the five major approaches discussed in the first part of this article have been well proven (see the September 1992 issue of *SDJ* for part one). However, other scenarios have yet to undergo extensive use. The complexity of the issues involved in downsizing is tremendous. T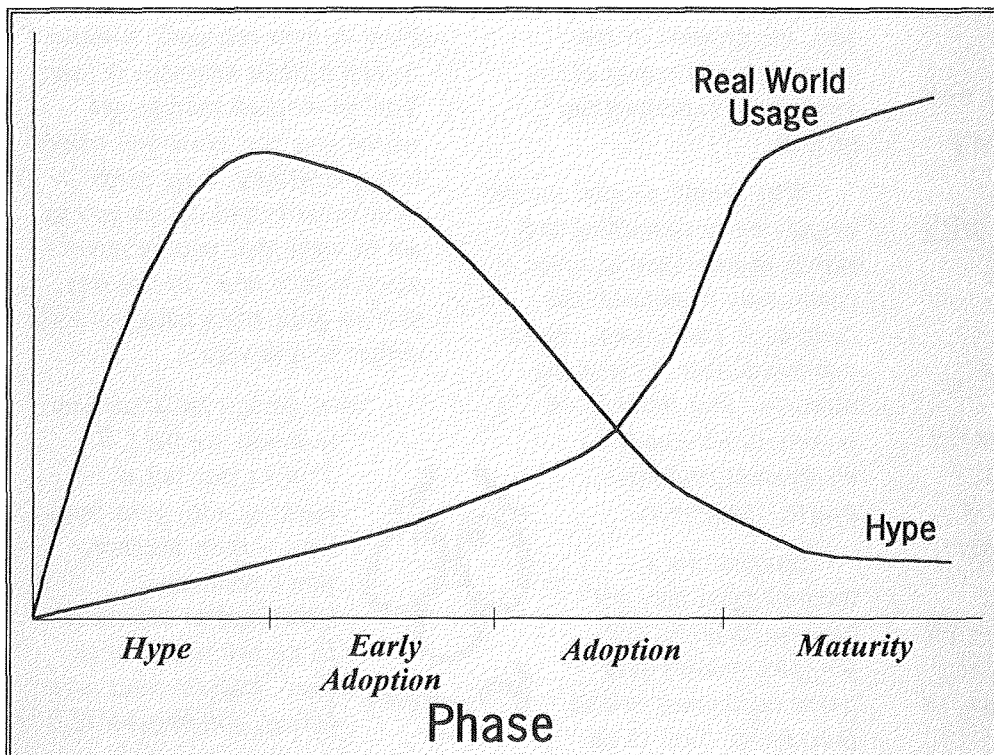he largest issue that anyone thinking of downsizing needs to address is whether or not you plan to keep existing applications. The retention of legacy systems will somewhat dictate the direction in which a downsizing project will proceed.

The common fallacy: PC hardware is so cheap, so why don't companies just change? The answer is that there is a sociology of computing that has been established over the past 30 years: people for the past three decades have interacted in the same manner with their hardware. We all agree in which direction we'd like to head, to smaller platforms, but we have to look at the entire picture in order to discover how to proceed efficiently and effectively to the new world.

The overall key to successful downsizing is to downsize IS culture. There is much more involved than just hardware. IS culture has evolved over a period of time to where we are currently in an evolutionary phase. To some degree, IS is unsettled due to the pace of change now in contrast to that of ten years ago. These are some of the issue we have to face. I believe that the most critical issues involved in downsizing are going to be faced by management. It is these managerial issues that will basically determine how successful corporations are in the downsizing process, rather than product selections or hardware costs. *HF*

Real World Usage / Hype

Phase: *Hype* | *Early Adoption* | *Adoption* | *Maturity*

Periscope

Downsizing: What's Really Going On...

"*T*ake care of your people, and they will take care of you."

Downsizing does not start with an inventory of software or hardware, it begins with an inventory of your employees. Human factors are the single most crucial item in downsizing. You can have the best equipment, programs, intentions, and plans, but if you don't care for the people, you will fail.

Failure might be the result of bad decisions, poor project management, incompetence, or even outright sabotage.

Success is a result of the proper attitude from the top. It is absolutely vital that you have the complete commitment of top management. "Top management" in this context means as high in the company as you can get. If you can receive a total commitment from your Executive Board of Directors, that helps. You'll need commitment from top management because of the length of time that downsizing takes, and because you will encounter more than a few surprises. It may even be that some of your key people desert you, the project, or the company. Some may even work hard to *destroy* the project.
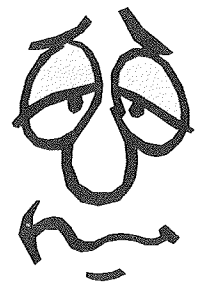
Why would people react so negatively to something that stands to help (and in some cases, may even save) the company? The answer lies in one word: fear. Fear is the ultimate *"four letter word."* It is usually accompanied by its multi-syllabic parent, *ignorance*.

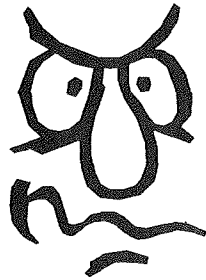Imagine for a moment (you may not have to imagine) that you have twenty-five years experience in data processing. You're an expert on mainframe programming and database design. You've played with the PC a little bit: helping your daughter learn a word processing application on the IBM AT at home. But, if the truth be known, your teenage daughter knows more about PCs than you do. Now you've been asked to head up a key committee that will guide the downsizing of applications at your company. How do you respond? A fairly typical human response is fear. You've worked hard to get where you are. Now you could lose everything. Fear grips your innards and drives you to act in ways that you've never acted before now. In one day, you've gone from being an IS expert to a novice.

Now, image for a moment that you are the CIO. You've just left a meeting with your boss. One way or another, you have been instructed to downsize IS by the end of next year. Back in your own office, you wonder if

it's possible. It's taken years of work to pull together a flowing operation that supports thousands of people every day. Now you are looking at starting over–undoing and redoing everything you've worked so hard to establish. Where do you begin? How do you learn when you don't even know what you don't know?

Although people react differently, you can expect that everyone whose job has been somewhat dependent on the mainframe will have a reaction to downsizing. The faces of your staff will seldom reveal the inner turmoil in their hearts. You'll never know the content of the late night discussions with spouses and friends. And you will only get very limited glimpses of their personal travail.

But be assured that at every meeting and on every project, there will be several people who are silently desperate. They are asking themselves constantly: Can I do it? Will I be laid off? Deep within the heart of every working person lurks that little voice that every now and then whispers, "One day, you will be found out." The decision to downsize changes this whisper into a shout.

You need a strategy to deal with such fear and ignorance. Ignorance will be the easier menace to dispel. This is done by assuming unfamiliarity: make the assumption that the people in your organization do not know what you wish that they did, and then make plans to deal with that lack of knowledge. Here are some practical suggestions:

- Order subscriptions to several magazines for each person. Some recommendations: *Data Based*

...With planning, luck, initiative, fortitude, and compassion, downsizing can be a positive experience for both your company and your employees...

Advisor, Info World, Byte, Schussel's Downsizing Journal, and *PC Magazine*. Have the magazines sent to their homes. Encourage people to copy and distribute key articles.

- Replace dumb terminals with PCs on every programmer's desk as rapidly as possible. Network the PCs together. Give your employees a budget for software. It is vital to let your people learn by doing. When it comes to PC-based networks, there is no

substitute for hands-on experience.

- Engage a good consultant. The acquisition of knowledge can be gained two ways: through your own experience and mistakes, or through someone else's. A good consultant will save you many times his/her fee by guiding you through the experiences of others and aiding you in avoiding the pitfalls that might delay or destroy your efforts. Get the best consultant you can afford. You are purchasing vital experience and wisdom. After your first few projects, you may not need as much assistance since you will have gained a great deal through your own experiences.

Be wary, though. Just as every lawyer can write a will or perform a real estate closing, every consultant claims to be able to downsize a company. In fact, relatively few have actually downsized anything. Get references for the individuals that you will be using and check them. Don't judge merely by a company's size or longevity in business. One savvy individual at a smaller company can save
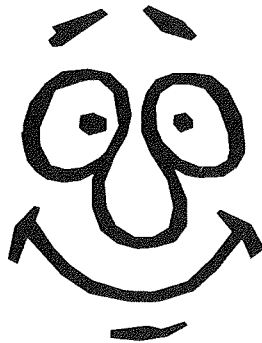
Periscope...

(continued from previous page)

you millions of dollars and years of wasted effort.

The consultant's personality is just as important has his/her knowledge. The consultant is going to have to walk a fine line. Many of your people will be doubly threatened: first by the downsizing then by outside help.

● Finally, get as much good training for your people as is possible. William Connor, Corporate Vice President and Director of Information Technology for Motorola in Phoenix, Arizona, offers the following principles on training as being key to their successful downsizing experience (the entire Motorola story will be featured in the December 1992 issue of *Schussel's Downsizing Journal*):

1. Start with a few people at a time.
2. Set up the new environment before you start training.
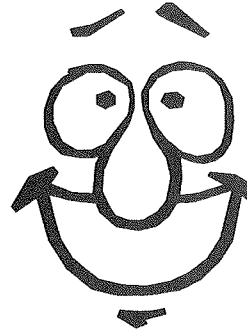3. Train people just before they are moved into the new environment.

4. Don't train everyone in everything, but do some cross-training over time.
5. Concentrate heavily on user involvement and commitment.

How you communicate change to people can become even more important than the details of what you do. Paul Flanagan, Vice President of Information Services at the Christian Broadcasting Network, Virginia Beach, Virginia, took an interesting and very effective approach to allaying employees' fears: "First, we restructured and established self-managing teams. During the nine months before we began downsizing, we intentionally introduced changes. Our goal was for employees to love change. When we started downsizing, it was accepted as just another change."

Flanagan took another innovative approach: at the onset of the project, he announced that in two years, the MIS staff would be reduced by fifty percent. But, he also explained that everyone would be trained in the new technology, whether or not they would survive the final cut. In other words, the Christian Broadcasting Network was commit-

ting to train people who they knew would be leaving the company so as to provide them with skills that would enable them to succeed at another firm. The result was that *no one left*. A happy group of employees is currently finishing ahead of schedule having completed a hugely successful downsizing project.

Dick Schell, Director of MIS at Turner Construction in New York City adds to this prescription from his experience. "The leadership must change first. If leaders can't get it all together and adapt, neither will anyone else."

With planning, luck, initiative, fortitude, and compassion, downsizing can be a positive experience for both your company and your employees. You may even be surprised to find that the benefits of downsizing are better than you had anticipated and accrue more rapidly. *RP*

Ron Peri is the President and Founder of Computer Support of North America, a firm which provides downsizing services from initial consultations through software conversion, hardware installation, and outsourced support. Mr. Peri can be reached at Computer Support of North America, Basking Ridge, NJ, 908-766-9200.

Distributed & Client/Server DBMS: Underpinning for Downsizing Part III of IV

This is the third in a four-article series by Dr. George Schussel on client/server and distributed DBMS architectures. This month's article examines some of the more advanced capabilities of both styles.

More advanced capabilities for distributed (or client/server) DBMS

Gateways – Many distributed database and client/server DBMS products have optional gateways that allow access to data stored in other DBMS. Lower levels of functionality provide for read-only access, while higher levels of function allow write access also. This higher level should always be accompanied by a two-phase commit capability across the different systems (general availability of this capability is still in the future).

Distributed access is a technology that is closely related to distributed DBMS. Distributed access is about the building of gateways that allow one DBMS to access data stored in another DBMS. This can properly be thought of as a subset of the technologies being delivered by vendors selling distributed DBMS or client/server DBMS technologies.

The demand for distributed access, of course, is the greatest in popular mainframe file and database environments such as IBM's IMS, DB2, VSAM, and DEC's Rdb. This is because local DBMS capability is not a requirement for distributed access. Instead, most vendors provide a piece of software known as a requester to run on the client side of the RDA environment. Some of the products in this market are not finished gateways, but tool kits so that users can build custom gateways.

Relational Integrity – An important server function that supports increased productivity in application development is relational integrity. This can include features such as referential integrity, or the ability to state business rules directly into the database using stored procedures or program triggers.

Triggers – Triggers are small SQL programs, written in SQL extended language, that are stored in the DBMS catalog. Each trigger is asso-

ciated with a particular table and an SQL update function (e.g., update, delete, and insert). They are automatically executed whenever a transaction updates the table. You can write triggers to enforce any database validation rule, including referential integrity.

Since triggers are stored in the catalog and are automatically executed, they promote consistent integrity constraints across all transactions. Triggers are easy to maintain because they are stored in only one place. They result in rules that are enforced for any application that accesses the database, such as spreadsheet programs.

Multi-Threaded Architectures – For the best performance, distributed (or client/server) DBMS should implement a multi-threaded, single-server architecture. Multi-threaded servers perform most of their work and scheduling without interacting with the operating system. Instead of creating user processes, multi-threaded servers create a thread for each new user. Threads are more efficient than processes – they use less memory and CPU resources. A multi-threaded server DBMS can service 10 to 40 users simultaneously on a machine as small as a 33 MHz 80386 PC with 10 MB of RAM.

Symmetric Multi-processing – Another advantage of

Distributed and Client/Server DBMS...

(continued from previous page)

DBMS servers is direct support of multi-processor hardware architectures in a symmetric multi-processing (SMP) mode. Most operating systems either currently (UNIX, Windows NT, VMS) or soon (OS/2) will offer support for this functionality. Therefore, there needs to be an effective integration between the DBMS and the operating system for gaining the advantages of the potentially improved throughput.

Direct support for SMP means that the DBMS can take advantage of several parallel processors under the same skin (with an appropriate operating system). These processors can be either tightly or loosely coupled. As of mid-1992, Borland's Interbase has been able to take advantage of VAX Clusters, which (at this time) neither Sybase nor Oracle can use to the maximum advantage.

Cursors – A cursor stores the results of a SQL query and allows a program to move forward through the data one record at a time. Sometimes, programmers are also able to move backwards within a cursor. Without a cursor, it becomes more difficult to program transactions to browse through data.

Text, Image, Date, and other extended data types –

Support for different types of data can make any DBMS useful in a wider variety of applications. To store a picture, it would be useful to have something like IMAGE data types of binary data. Another useful item is TEXT data types, which are printable character strings.

Remote procedure calls (RPC) – RPCs allow an application on one server (or client) to execute a stored procedure located on another server. Stored procedures enhance computing performance since all of the commands can be executed with one call from within the application program.

Multi-platform implementations – Another primary advantage of a robust DBMS is multi-platform portability and networking. If your software runs on many different vendors' hardware, then you have that much more flexibility. For example, Oracle was built with an approach that has outdone all other DBMS products as far as the variety of supported hardware.

Disk Mirroring – For companies wanting the reliability of mainframe environments on the PC LAN, a disk or server mirroring capability is necessary. Mirroring implies that dual operations are executed for each computing step with error reports whenever there is any difference between the results. Mirroring also allows

the system to continual operate essentially at full speed, even after one of the processors or disks has failed. Disk mirroring is supported through the process called "shadowing." This is a very useful facility for applications that require extremely low amounts of down-time – if one disk fails, the system will automatically divert and use the other disk without interrupting user operations.

BLOB data types – A BLOB (binary large object bin) data type has no size limit and can include unstructured, non-relational types of data such as text, images, graphics, and digitized voice. It is possible to handle BLOBs as a single field in a record, similar to a name, date, or floating point number. It can then be governed by concurrency and transaction control.

The ability to create "database macros," which can be executed by the database engine should be supported within the DBMS. These macros would be implemented as centrally-stored, user-written procedures that tell the database system how to translate BLOB data to other formats. Because they are stored in one place and managed by the database, BLOB macros are simpler to create and maintain than similar code found in an application.

Host Busters! A Visit to PACIBM...

(continued from front page)

☑ Downsize decision support and development functions first. Leave the OLTP applications until after more experience is gained.

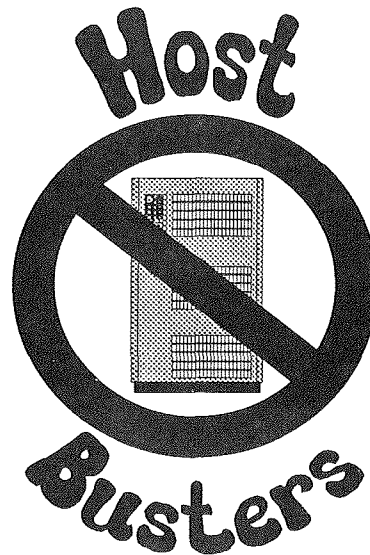☑ Don't downsize complex, performance-sensitive OLTP applications at all.

Prologic Computer Corp. of Richmond, British Columbia arranged for my visit to PACIBM. Prologic's PROBE database management system (DBMS) and 4GL were the essential enabling technologies used at the bank. My host was Daryl Tanner, President and CEO of PACIBM. Tanner was unlike any bank president I had ever met. He was just as versed and comfortable in talking about bus architectures and improved disk caching schemes as he was discussing new ways to provide banking services to PACIBM's customers.

The bank's history

A credit union is a bank owned by its depositors, the members of the credit union. At PACIBM, in seven of the western United States, any IBM employee (and/or relative) may become a member. In total, PACIBM has 49,000 members with a total of 54,000 savings accounts and 20,000 checking accounts. The bank has issued 20,000 ATM cards and runs eight branches throughout California and Arizona. In addition, PACIBM manages approximately 25,000 loans, 14,000 VISA card accounts, and handles $350 million in assets.

The final result of PACIBM's downsizing has been a PC LAN client/server system that processes

100,000 production and decision support database transactions every day. The bank is totally dependent on the non-stop operation of this system. The IBM 4381 this system replaced was still in the PACIBM computer room during my visit, but its sole function was to support two older ATM machines. The plug on the 4381 was scheduled to be pulled sometime this fall. The difference in size between the IBM PS/2 Model 95 with its inboard SCSI disk was hilarious when compared to the multiple large cabinets of the 4381 and its 3880 disk subsystems. The tape backup system for the Model 95 was also only about 1/100 the size of the tape drives supporting the 4381.

The story

In 1990, Tanner, with the approval of the PACIBM board of directors, prepared a plan for replacing and upgrading their existing computer technology base. PACIBM's systems were running on an IBM 4381 mainframe with 16 GB of 3880 disk supporting 151 terminals. Since most of the PACIBM board consists of IBM employees from advanced IBM research and development labs such as Almaden and Santa Theresa, the board echoed Tanner's desire to move their data processing out of the 1970s and into the modern era.

The bank's systems had been written in COBOL and RPG, and were difficult to maintain. As is typical in systems with older software, the MIS department was only able to respond to maintenance requests slowly. The 4381 systems didn't support new types of applications such as those using ATM software.

Host Busters! A Visit to PACIBM...

After surveying available packaged and development software, a final solution was presented for approval to the PACIBM board. As Tanner explained, "I told the board that I wanted to recommend a rather radical proposal for the new system. The general reaction was serious apprehension as the member's imaginations ran away with different ideas of what I might be proposing. When the idea of developing our own systems using an SQL-based DBMS and 4GL running on PCs and LANs was presented, the look of relief from the board was very visible. One of the members commented 'We were afraid you were going to recommend a DEC system!'"

The configuration of the new system consisted of using the IBM Model 95 as the base for various servers such as database, file, and print (reports). IBM Model 55s were to be used as clients. The software would be a custom application built with the PROBE software system from Prologic Computer Systems. PROBE, while not generally well-known in banking circles, has

received a "Most Outstanding Product of the Year" award from the MicroBanker Association. More importantly for PACIBM was the fact that PROBE had been successfully used to completely automate the Richmond Credit Union of Canada, an operation approximately three times the size of PACIBM.

Prologic initially bid on the PACIBM job using the Richmond Credit Union application as the basis for customization; the intention was that Prologic would alter the Canadian package to meet US banking regulations. However, once the development team looked closely at the requirements, it was decided that writing a new application would the easiest solution. As Tanner explained to me, North American banking regulations are very different (as well as more complex and numerous) than Canadian regulations. For

example, US laws require the reporting of any cash deposit over $10,000 within a 24 hour period. If that requirement were for a single deposit, then it would be an easy task to implement. The regulation, however, must be applied to *all* deposits at *all* branches during a 24 hour period, and those 24 hours must be interpreted as a *rolling, ongoing* 24 hours, not just any single day.

The final contract approach was to award the software development to a Prologic VAR and the systems integration and hardware contracts to IBM (of course). Efforts began in January of 1991 and the system went on-line February 29, 1992.

The goals

I asked Tanner what the goals for the new system had been as a prelude to asking him how well the new system had met those goals. He responded with the following list:

1. The movement to the new system must be seamless. Any trauma of movement would need to be minimized for tellers and other system users.

2. Improvement of the quality of support and

system services was necessary. This would include features such as better and more timely access to information and improved financial controls.

3. Improved application software maintenance was considered to be essential. In particular, the ability to maintain the system must be enormously easier than implementing change under the legacy COBOL/RPG environment.

4. The resulting application should be easily and modularly expandable, for both hardware and software. This is to anticipate the accommodation of new, added services, and/or for additional members or business.

5. Finally, the system must be seen as an "empowering" of all employees.

Since the motivation for so many downsizing projects is the desire for cost savings, I asked Tanner about this point. Cost savings had not been a principal goal of the conversion project, but PACIBM had, in fact, achieved important cost savings for normal operations. Tanner's estimate was that savings over the first year were probably on the order of 20%. However, at the time this article went to press, Tanner did not know the total value of cost savings for 1992. However, PACIBM places a higher value on the added, easily implementable functionality. As an example, I was showed the micro-based voice response system that PACIBM has installed. This system, which interacts closely with the PROBE database, was built on PCs with DOS technology boards acquired on costs that typically run in the $100s of dollars range. Comparable mini-computer-based technology typically runs in the $10,000s of dollars.

The Good

We then talked about both the surprises and the anticipated results achieved with the new system. In particular, I was interested in what results were better (or worse) than expected. Tanner explained that the good news was four-fold:

Customized systems – The fact that PACIBM's systems were custom-built meant that automated support was available for the way the staff wanted to work; it wasn't necessary to change established procedures to meet the demands of a new software package. The superior development environment provided by PCs and 4GLs greatly improved PACIBM's ability to implement system changes. Tanner informed me that business conditions currently result in an average of three to four production system changes each day. Such flexibility was totally unachievable on the mainframe system.

Integration with the desktop – The use of PCs for operational systems allows for tight integration with office automation packages such as word processors and presentation graphics. Using the Windows clipboard means that PACIBM staff can move data and text between data processing applications and office applications. In addition, a single 386 PC acts as both the office automation support tool and the data processing terminal.

Better access to data – The old system and its rigid file structure had limited access to data through predetermined and programmed paths. Evolving business practices, however, continually mean that a bank needs improved access to data. As an example, Tanner pointed to a new functionality in the system that hadn't been previously available – background referencing. In this new application, a teller can call

Host Busters! A Visit to PACIBM...

(continued from previous page)

up credit history and loan information for on-line review while a customer is waiting to have a check cashed. With this new option, the bank staff has more and better data to support a decision to cash, for example, a large check that otherwise might not be cashable until additional funds had cleared.

Open systems – While not wanting to get into a debate about what constitutes "open systems," Tanner clearly enjoys the ability to acquire new capabilities for PACIBM's system by scanning catalog pages in journals such as *PC World*. Illustrating another advantage of PACIBM's new environment was the clone computer on Tanner's desk. Even though IBM was the chosen hardware supplier, IBM's backordered position at certain times in the project's evolution meant that PACIBM needed to order clone equipment to fill gaps in delivery schedules. This, of course, proved to be no problem!

The bad

After reviewing the pleasures of the new system, I asked Tanner what he

believed to be the worst problems. His response was that their new downsized system wasn't in any way disappointing, but that the biggest implementation *surprise* PACIBM encountered could have been handled better. The Novell NetWare 3.11 LAN O/S supporting the database server had been sourced from and installed by IBM. When PACIBM's new system first went on-line, the system's response-time was slow. This problem was immediately ascribed to NetWare performance. However, the IBM engineers on the PACIBM support team were not familiar enough with NetWare to determine how to improve operations. Eventually, IBM produced network analysis equipment and a qualified engineer to help the PACIBM staff tune NetWare to a satisfactory performance. In addition, a combination of telephone advice and manual reading was used to develop the necessary solution techniques.

The future

As I asked Tanner about future directions for PACIBM's systems, again his technical background proved strong.

First of all, we spoke about database

performance. Even though the PROBE relational database is highly optimized to take advantage of both DOS and NetWare environments, Tanner wanted even better performance. Since the Model 95 server with its 486/33 processor is a top performer for single processor boxes, Tanner had some ideas about improving performance. He mentioned that PACIBM was already using Model 95s for non-database services such as file and report management and generation. He wanted to achieve further added functionality by using multi-servers against a single database.

Prologic will supply this capability with its NetWare redirector file sharing capability which is due out this fall. As Greg Hope, Prologic's Founder and Vice President of Development explained, this new architecture will replace the proprietary DOS-based Profile file system with a NetWare redirector from multiple servers to a single NetWare-based file system containing the database. In other words, multiple servers will run the PROBE DBMS against a single NetWare file server housing the data.

Tanner further described more powerful hardware as a method of increasing PACIBM's customer service. He indicated his

The Bad News

desire for the new Model 290 (the Parallan designed machine) that IBM will be manufacturing this fall. Tanner indicated that this new model should solve what he felt were limitations in the current IBM Model 95 bus architecture and disk caching.

Improvements in disk subsystems were also high on Tanner's wish list. After looking at RAID technology, he was not satisfied with the tradeoffs that currently need to be made between security, mirroring, striping, and performance.

In the software arena, Tanner's two main wishes were for operating systems

and imaging. For operating systems, he is looking at the possibility of using more OS/2. Since running under OS/2 isn't one of Prologic's current objectives, however, it may not be reachable in the short-term. We talked about Windows NT as an alternative since PROBE is currently being tested in that environment. Given the end-of-year due date for NT's delivery, however, any migration in that direction is still a while in the future.

As soon as the capability for implementing imaging software and hardware is available, PACIBM is likely to travel down that road. Tanner talked about the

service and storage advantages of allowing the retrieval of items such as mortgage and loan documents.

The conclusion

When I asked Tanner if he would undertake the project all over again, his reply was, "Absolutely, but if we could, we would do several things differently." Tanner commented on the fact that PACIBM could have used better support for NetWare. When the bank needed NetWare assistance, IBM had only recently

Distributed and Client/Server...

Application Specific Functions – This capability allows a user to easily extend the range of database commands by adding new functions, coded in C, to the DBMS kernel. This facility is helpful in the manipulation of BLOB data.

Event Alerters – An event alerter is a signal sent by the database to waiting programs that indicates that a database change has been committed. Event alerters work remotely and can span across multi-vendor networks. Although it would seem to be simple to

add event alerters to a system that supported the concept of triggers, implementation of the technology is made difficult by the need to support an asynchronous, heterogeneous environment.

Event alerters offer the following benefits:

✪ The waiting programs consume neither network traffic nor CPU cycles.

✪ Notification is effectively instantaneous rather than dependent on some polling interval.

✪ Event notification works remotely, even across different platforms. The notification mechanism is managed by the DBMS.

✪ Unlike a trigger, an event alerter can affect programs running outside the database.

Multi-dimensional Arrays – In scientific processing or time-series commercial types of applications, array support for the database is useful. Arrays are stored as single fields in records so that retrieval is expedited. Arrays are widely used in scientific processing and are very expensive to normalize for a relational DBMS – normalization typically means creating redundant data to generate separate records for information that is really only different at the field level. *GS*

Host Busters! A Visit to PACIBM...

acquired distribution rights to the product so that IBM engineers at PACIBM were not adequately knowledgeable.

Another area that should receive additional attention during development is the "stress test." Just testing the software for current operation is not adequate in a database application. Tanner would have liked to have a reasonable way to test the system performance at full load, so that actual performance could have been closely simulated. That capability wasn't available and it took a live system to really apply the stress.

My visit to PACIBM was convincing evidence that downsized solutions are a good idea today for almost all development projects. If you still don't believe that mission-critical, high up-time, OLTP applications can be run on PCs, then you should visit Pacific IBM Employees Federal Credit Union. **GS**

UPCOMING **downsizing** Events...

DATABASE WORLD, one of DCI's premier events will be in Chicago, this December 8-10, 1992. Being held in conjunction with **DATABASE WORLD** is **CLIENT/SERVER WORLD** which features four conferences on Managing the Client/Server Environment, Building Client/Server Applications, Client/Server Databases, and Client/Server Networking. Dr. George Schussel is the chairman of these events. He will be joined by keynote speakers Dr. Robert Epstein of Sybase, John Landry of Lotus Development Corp., Charles Stevens of Microsoft, Shaku Atre of Atre, Inc., Larry DeBoever of Tucker/DeBoever Technologies, and Chris Date, Independent Consultant.

Implementing Client/Server Applications and Distributing Data, one of the seminars in DCI's Downsizing Seminar Series, is being held in Orlando, November 5-6, 1992. In this two day course, Instructor Herb Edelstein will cover the following topics: Distributed Systems, RDBMS, Networks, Distributing Data, Database Servers, Distributed and Federated Databases, Distributed Queries, and Transaction Management and Concurrency.

Another seminar from the Downsizing Seminar Series, **Finkelstein's Practical Guide to Client/Server DBMS Computing**, is being held in Toronto, November 23-24, 1992. The seven topics to be covered include: Client/Server Environments, Choosing a Database Server – Feature Comparison, Database Servers: Highlights and Conclusions, Understanding Database Benchmarks, Choosing Front-end Tools, Case Tools for Client/Server Development, and Client/Server Case Studies.

For more information on any of these classes, call DCI at (508) 470-3880.

For your enjoyment and ease of reference, like a chameleon, **SDJ** will be changing its colors monthly. Our changing color scheme will make it easy to spot each new issue at a glance, and quick to reference past issues.