DB2 magazine

http://www.db2mag.com/db_area/archives/1996/q4/9601beu.shtml

Sysplex and DB2: Version 4 Data Sharing

By David Beulke

Winter 1996

This nimble new major enhancement for the MVS family lets you share data across multiple operating system environments.

As the representatives of the computer leasing company crate up the old water-cooled CPU, a smile comes to your face as you remember the day they brought it in. Nostalgically you recall the many nights and holidays you spent with it when normal people with lives were celebrating these occasions. Meanwhile, you and your coworkers were figuring out the problems with the latest system changes or reorganizing the large data warehouse that would only have a few periodic updates.

Generally your relationship with your CPU is changing. No longer are you keeping it for years; instead, you're replacing it when it's only in its "terrible twos." Just as dramatically as the PC desktop prices keep falling, mainframes' price/performance is now cheap and will continue to get cheaper. This price reduction is only one of the reasons why so many organizations are resurrecting their mainframes and are beginning to install DB2 Data Sharing technology.

As nostalgic as I was seeing our big old water-cooled workhorse go, I was even more amazed seeing its replacement: a new CMOS2 CPU. This nimble CMOS2 box -- the size of a big refrigerator and packing even more CPU punch than its older brother -- is air-cooled with smaller (25MIPS versus 60MIPS) CPU engines. (The New Maestro [CMOS3] MIPS are nearly twice that of CMOS2 -- almost 45 MIPS per engine.) The air-cooled feature saves money -- big dollars, in the hundreds of thousands for a typical data center. CMOS technology and DB2 Data Sharing also represent a major enhancement for the MVS family, from sharing-disk to sharing-data across multiple operating system environments.

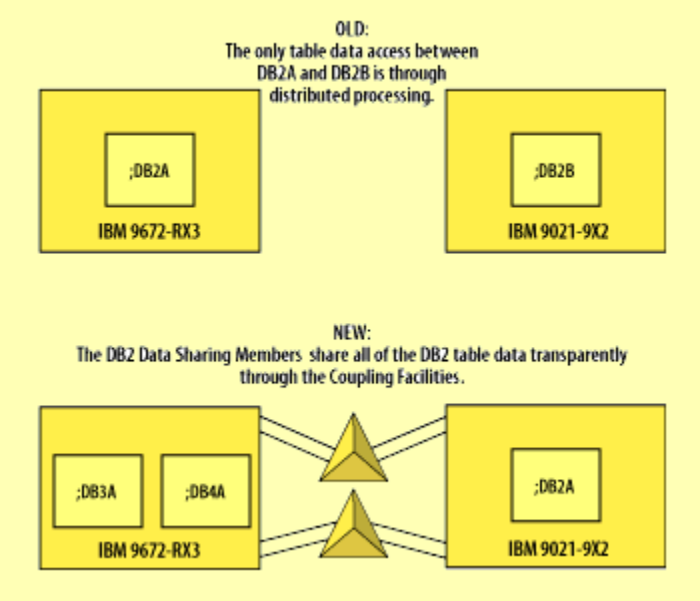
The shared-everything architecture opens operating systems to share data everywhere. Establishing an open operating system architecture previously required specialized hardware and software from a range of solution vendors. Now, MVS itself offers expandability and scalability of processing power by adding complete, cheap CMOS operating system complexes. Sharing resources and data across the Sysplex addresses large application system requirements, provides needed flexibility, and gives resources to application solutions and their priorities.

SHARING: NEW FLEXIBILITY

DB2 Data Sharing lets you update the same information from many different operating systems. This sharing is possible because of a new hardware component called the Coupling Facility

(CF). (See Figure 1.) The CF manages the locking, communications, and buffer pool activities of sharing data across the Sysplex DB2 configuration. The CF hardware and DB2 version 4 Data Sharing software are key partners in the new shared-data architecture.

Figure 1. Old and new DB2 environments.



The new CMOS price/performance technology, along with the suite of Sysplex OS/390 software, provides a powerful, centralized, standard, and secure platform to service any and all workloads. Various alternative operating system kernels, such as Unix and Windows NT, are able to work alongside MVS in the OS/390 CMOS configuration. These solutions are cheap whether they are intended for legacy or advanced development systems.

At the heart of the OS/390 shared-data environment is the Work Load Manager (WLM) component. This software component for systems and specific transactions provides the routing adjustments and tuning capability for "goal mode" operations. Giving a workload or transaction the specific priorities (or "goal") it warrants from an overall Sysplex perspective, the WLM will route the transactions to the lowest-utilized Sysplex CPU operating system image. This balancing act provides the flexibility to promote system integration. Sharing everything across the system through the CF and WLM guarantees performance and scalability of CPU resources for all processing priorities. This scalability -- to add hardware and clone software systems easily for cyclical or emergency workloads -- provides the capability and flexibility of incremental CPU resource growth. Instead of isolating workloads to guarantee system resources, the WLM provides the flexibility to tune the whole data center's processing CPU and I/O priorities.

SYSPLEX & CICS PLANNING

The road to Sysplex and data sharing is usually undertaken in the quest for added availability, reliability, or the flexibility of incremental capacity growth. These goals are extremely enticing for data centers with large or mission-critical applications. The new parallel Sysplex software licensing pricing is giving financial incentives to shops that move to this new environment. The availability, capacity features, and financial incentives of the new CMOS technology CPU's

price/performance are making Sysplex processing attractive for large-scale data centers.

Usually an application drives the migration to Sysplex processing via availability or capacity issues. Taking this application to the Sysplex environment requires careful analysis, planning, and implementation. Within the application's requirements and processing are the analysis decision points. These analysis points are, for example, the releases of all the software products used in the application -- everything from the IBM software to the neat little software routine someone picked up that solved an obscure problem. Knowing all the ins and outs of the software releases will help you determine their compatibility within a data sharing environment.

Planning and preparing for Sysplex processing requires installing or upgrading several pieces of software. The daunting task of upgrading MVS, CICS, IMS, RACF, DB2, and many other software pieces is being prepackaged and tested as an integrated solution. This package will save shops the months or weeks of integration, compatibility, and verification.

Before migrating, you must know how the application executes. Does the application have unique file, resource, or processing constraints or requirements? With Sysplex DB2 data sharing, the application should be able to run concurrently on several different operating system "clones." If one system has slow response time or fails, the Sysplex WLM software component will sense the difficulty and dynamically shift the workload to another system that will handle the processing. Having all the application components in all the system clones is vital for shifting workloads dynamically. Knowing all the processing points and configuring them to be shared or accessible from various subsystems is the core issue and consideration in Sysplex processing.

Evaluating the "shareability" of all your processing is no small task. You must analyze each online and batch process. Fortunately, some software tools can help you analyze online CICS regions and transactions. This analysis will identify the "affinities" within the CICS transactions. Affinities are unique points or processing dependencies that would prevent workloads from being routed to another operating resource, such as a CICS exit, temporary storage, or intersystem data passed among multiple transactions. Affinities can be related to a user or system with special settings or a physical device such as a terminal or printer.

These types of affinities can also have different durations. The durations can be associated with the specific CICS region's settings. The transaction's processing must take place in a specific CICS region, an Application Owning Region (AOR), to function correctly. An affinity duration can also be associated to the transaction's pseudo-conversational processing nature, building information during its processing lifetime, or to a user or terminal logged on to a particular CICS. This logon or user affinity is usually related to the features and/or user's security profile. Once the software affinity analysis has been completed, the job of resolving or satisfying the affinities is a time-consuming task.

PLANNING CONSIDERATIONS

The DB2 portion of the Sysplex picture requires a minimum release level of DB2 version 4.1. Version 4.1 is necessary because of the data sharing "hooks" into the CF and the new Type 2

index structures. Because the only migration path to version 4.1 is through version 3.1, any shops lagging behind on maintenance or releases should upgrade as soon as possible.

Naming conventions for all the components within MVS, CICS, IMS, DB2, and others used in the Sysplex are important for several reasons. These resources must be easily identifiable, cloneable, and manageable. These factors are extremely important because of the number of resources that might be moved or added to various MVS operating system images or CPUs. For example, having all the DB2 components named consistently makes it easier to know which resource is being used by which Sysplex subsystem. Having this naming convention planned in advance can make migrating to a DB2 Data Sharing Sysplex implementation easier. See Table 1 for a list of naming conventions for the MVS and CICS Sysplex components.

Table 1. Example data sharing naming conventions

DB2 Group Name	DSNDB0G
Catalog Name	DSNDB0G
Group Attach Name	DB0G
Member Names	DB#G
Member command prefix	-DB#G
Member subsystem name	DB#G
Member BSDS	DSNDB0G.DB#G.BSDS01 DSNDB0G.DB#G.BSDS02
Member active log prefix	DSNDB0G.DB#G.LOGCOPY1 DSNDB0G.DB#G.LOGCOPY2
Member archive log prefix	DSNDB0G.DB#G.ARCLG1 DSNDB0G.DB#G.ARCLG2
Member work file database	WRKDB#G
Member procedure names	DB#GMSTR DB#GDBM1 DB#GDIST DB#GSPAS
Member subsystem parameters load module	DSNZP0#G
IRLM group name	DDXRDB0G
IRLM member subsystem name	DJ#G
IRLM procedure name	DB#GIRLM
IRLM member ID	# (NUMBER)

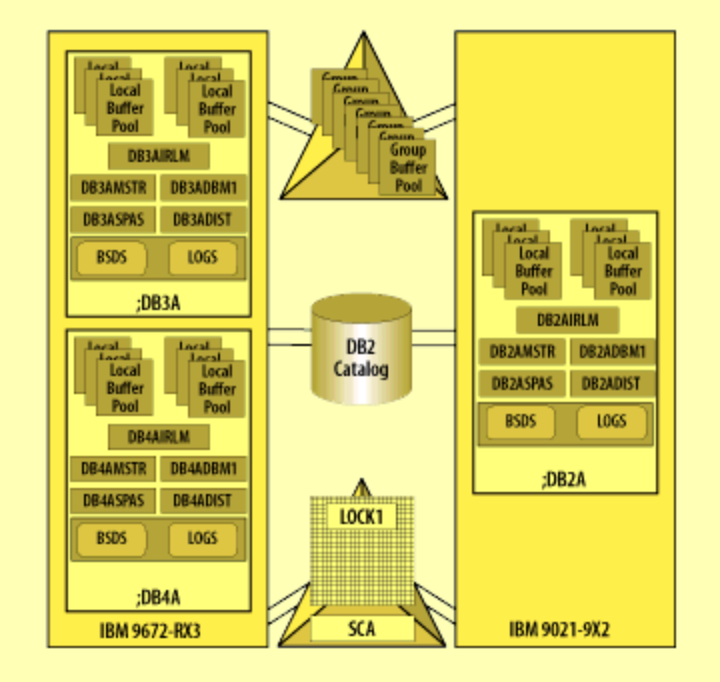
THE COUPLING FACILITY

A vital piece of the Sysplex data sharing environment is the coupling facility (CF). The CF control program functions can be implemented through either a separate Integrated Coupling Migration Facility (ICMF) logical partition or a new standalone CF hardware configuration.

The new standalone CF is a CPU, memory, and communications configuration that enables communications between the various Sysplex operating systems and DB2 data sharing group members. The CF uses fiber optic, non-ESCON connections and its special control program to perform high-speed caching, locking, and list functions. These CF functions enable the Sysplex environment to share and communicate with all its resources. Because the CF is vital to the sharing environment, having dual CFs with dual channels to each sharing environment is a standard for being fault-tolerant in a high-availability environment.

DB2 Data Sharing uses the CF for its Shared Communication Area (SCA), Lock, and Group Buffer Pools (GBPs) structures. (See Figure 2.) These structures are defined into the CF and managed under a CF Resource Management (CFRM) Policy. The CFRM Policy handles and manages the structures in the event of any failure and can dynamically recreate structures. To avoid a single point of failure, dual CFs are recommended so the structures can be spread across both CFs. For example, the SCA and Lock are defined on one CF, and the GBPs are defined on another CF. This structure configuration (with appropriate CFRM Policy failure statements) will dynamically rebuild the structures on the remaining functional CF.

Figure 2. Example of DB2 data sharing environments.



Sizing the DB2 Data Sharing structures is also very important. The SCA sizing has four guidelines: small, medium, large, and extra large. This structure size, which can be defined in 4k increments, is based on the number of databases and tables in the DB2 Data Sharing Group. Analysis recommends the settings shown in Table 2. Sizing the GBP caching structures requires detailed analysis. First, total the memory of all the local buffer pools and hiper pools involved in the data sharing environment that need a supporting cache structure. This total storage requirement is your beginning point. The next step is determining the storage needed for caching intersystem sharing interests. Perform analysis of the purposed sharing configuration, and evaluate the number of DB2 data sharing subsystems and the amount of sharing and update activity. Considering this analysis, multiply your local buffer/hiper pool storage totals by the

intersystem interest factor: five percent for light intersystem interest, or up to 50 percent for heavy intersystem interest. Having enough structure storage is vital because DB2 will add intersystem interest pages that cannot be cached to the GBP. Instead, the pages will be added to the Logical Page List (LPL). The LPL indicates that a logical error has occurred, and it leaves the pages inaccessible until they are recovered.

Table 2. Recommended Shared Communications Area (SCA) sizing.

Site Size	Databases	Tables	SCA Structure Size
Small	50	500	8MB
Medium	200	2000	16MB
Large	400	4000	32MB
Extra Large	600	6000	48MB

Because the Lock structure manages all the intersystem interest and contention for resources, determining its proper storage allocation is vital. The Lock structure is used by all the DB2 members' internal resource lock managers (IRLMs) to communicate and reflect all the resource locks used in the data sharing configuration. Totaling the storage from all the IRLMs gives a sum of the complete locking interest in the system. Next, divide the sum by 2, multiply it by 1.1, and round to a power of two. For example, if 10 DB2 and IRLM pairs are in the DB2 Data Sharing group, with a maximum Extended Common Storage Area (ECSA) of 6MB for each, the calculation would be:

((6MB x 10) / 2) x 1.1 = 33MB
(Rounded to 32MB)

Sizing the Lock structure correctly is vital, because if you do not have enough storage available for locks, DB2 could fail or terminate threads.

After migrating to your DB2 version 4 naming convention and defining your structures in the CF, DB2 is ready to be enabled for data sharing. The first phase is enabling the system for one-way data sharing. Enabling one-way data sharing is similar to a DB2 version migration. The system installation panels generate system jobs that will enable your DB2 system. Starting your enabled one-way data sharing DB2 system or the originating member will allocate the SCA and the Lock1 structures within the CF. The GPBs will not be allocated until the additional DB2 members of the data sharing group are defined, started, and used to cache intersystem interest. Once the data sharing originating member is enabled, going back through the installation panels and generating the jobs to define the additional members is simple. At this point, your naming convention plans start to pay off, and you can think about the next testing and performance phase.

ANALYSIS AND TESTING

When you began down the road to Sysplex and DB2 data sharing processing, an application drove you to this decision. Having that application perform perfectly is the goal of the analysis and testing phase.

To begin your analysis, determine the optimum setup for your DB2 data sharing environment. Determine the number of data sharing members, CICS regions, and everything else you will need to process the workload or provide the targeted availability. Start with your current configuration and modify it for your beginning Sysplex DB2 data sharing configuration. Overlay your naming conventions for your systems and see if they need any fine-tuning. The first step is to validate all the test environments. The next step is to begin testing transactions and workloads in the data sharing environments. Determine and validate the setup of all the systems and special cases that are implemented to resolve CICS affinities, resource problems, or software product constraints. Validation of software licensing agreements can be a critical factor for spreading a workload across multiple CPUs. Validate your software contracts to see which machines are available for the workload. Establish whether your new configuration brings any new problems or system constraints by evaluating the workload response time. Closely monitor the impact of the smaller CMOS technology CPU engines.

Testing workloads on the smaller CMOS technology CPU engines can highlight the difference between a 60MIPS CPU engine and the new 45MIPS CPU engine. This difference can become apparent if your workload is CPU-intensive. DB2 utilities and specialized processing may require the larger CPU engines to make their processing windows. These hardware requirements are known as hardware affinities. Analysis shows that most workloads -- typical CICS or DB2 workloads -- can be spread effectively across the smaller CPUs without any response time or availability problems.

Next, determine your typical batch and online job flow and its impact within your Sysplex and DB2 data sharing environments. Are there critical times when your CICS transactions and batch jobs should not be spread across multiple Sysplex complexes? Your environment may need to change or evolve as the day's workload complexion changes. With the WLM and other priority adjustments available within your Sysplex environment, utilizing dynamic configurations is possible and ideal for different workload profiles.

Analyzing the batch environment is important to find the CPU- and I/O-intensive programs or processes. You must consider these processes in your Sysplex DB2 data sharing environment. The CPU-intensive processes will need to run on the larger 60MIPS CPU engines, whereas the I/O intensive programs will acquire a large number of locks across the data sharing environment. If a process is performing updates against a large, frequently accessed table, the number of locks propagated into the CF structures could be enormous. Running that process on the same Sysplex subsystems with the other processes that share that table will help minimize the locks and pages propagated to the CF.

IMPLEMENTATION ISSUES

The key to achieving the best performance in a DB2 data sharing environment is to minimize as many things as possible. Convert your old indexes to the new Type 2 indexes, which do not take locks and therefore are not registered into the CF structures. Minimize the amount of data locked by having frequent Commit points to release locks during a program's execution. Define enough storage within the CF lock, SCA, and GBP structures to minimize the number of signals communicated between the data sharing members and the CF structures.

Why are Type 2 indexes important to the Sysplex DB2 data sharing environment? Type 2 indexes do not take locks, eliminating index contention for the data being used. Because Type 2 indexes do not take locks, the only locks being communicated to the CF structures are the data page locks. Also, the new Type 2 indexes allow the table to be altered for data row-level locking. Use row-level locking with extreme care because even though there are no Type 2 index locks, the number of row (data) locks propagated to the CF could be enormous. Convert all database tables targeted for the data sharing environment with old Type 1 indexes to Type 2 indexes to eliminate the extra locking and communication to the CF.

Implementing Type 2 indexes can have a dynamic effect because of the extra CPU and communications that were necessary for the old Type 1 index locks. Performance improvements will be directly proportional to the number of Type 1 index locks used in the application. The old Type 1 indexes can be very detrimental to a data sharing environment because of the extra CPU and communications necessary to manage their index locks. A further negative impact of old Type 1 indexes is the lock information held in the CF and communicated to the multiple data sharing members. You can achieve performance improvements by implementing the new DB2 version 4 Type 2 indexes, because they do not communicate locks in the data sharing environment. When going from a one-way to a two-way data sharing environment using old Type 1 indexes, IBM benchmarks indicate the performance impact to be 25.75 percent. Benchmarking the same workload with the new Type 2 indexes indicate that the impact is only 13.29 percent. Carrying the old Type 1 lock index information, the costs were almost double. For this reason, Type 2 indexes are vital for DB2 data sharing performance.

The Bind parameters can affect the number of locks retained by the DB2 program or plan. The first factor that determines the number of locks retained is the Isolation Level. The Isolation Level defines the scope of how long DB2 retains the locks. The four Isolation Level settings are: repeatable read (RR), cursor stability (CS), read stability (RS), and uncommitted read (UR).

The RR setting holds the lock on all the data accessed until the run unit issues a Commit. The CS setting holds locks on only the data being accessed currently. The UR setting, sometimes known as the "dirty read," offers query applications the opportunity to read uncommitted data. This UR takes no parent or child locks and permits the data to be read without possible contention.

The other Bind parameters affecting the number of locks are the Release and Currentdata parameters. The Release parameter, with options of Commit or Deallocate, determines when DB2 will release any acquired parent (tablespace, partition) locks. The Commit option may release the locks more frequently than the Deallocate option during a program's execution. Using Deallocate is recommended to cut down on the number of communications and locks released and re-acquired for particular data resources. This process can be particularly costly for data resources required in the CF because of the interest by other DB2 data sharing members. The Currentdata parameter tells DB2 whether the database information and the data captured through the definition and opening of an SQL cursor should be kept identical until the next Commit point. Even though this parameter is only applicable for Isolation Level CS read-only cursors, it can create additional page or row locks that can increase deadlocks and CF contention. Using the Bind parameters properly can save 40 to 50 percent of the number of

locks retained and propagated to the CF. Monitoring these parameters closely can have a dramatic impact on overall performance.

DB2 buffer pools and their tuning considerations are an involved, complicated, site-specific subject. The separation and various threshold adjustment issues can be debated endlessly. To use DB2 Data Sharing effectively, the buffer pools and their matching GBP caching structures are a critical piece of the overall Sysplex's performance.

The first issue of buffer pool separation is vital for setting up an efficient Sysplex data sharing environment. By separating the DB2 catalog, DB2 sort area (DSNDB07/32), user tables, and indexes into multiple buffer pools, you can isolate and properly tune read-write efficiency to improve overall performance. By grouping the tables and indexes with similar I/O characteristics, you can easily adjust various thresholds to improve buffer pool efficiency. All the tables to be shared across the Sysplex must have their buffer pools defined and supported by GBP structures. Limit the number of tables using those finely tuned buffer pools in order to focus critical memory resources. Separating the data into different buffer pools also provides operational options in starting and working with the data sharing members. If your environment has only one buffer pool, BP0, separating the tables into several different buffers should be a priority.

The second issue with GBPs is the local DB2 member's buffer pool performance and efficiency. Without having good local buffer pool performance and efficiency, the GBP effectiveness will suffer. If the local buffer pool performance is poor, such as having a large number of sync reads or page thrashing, it will reread data pages to pass to the GBP, doing twice the I/Os for data with intersystem interest.

Sometimes the thresholds associated with buffer pools can be modified to improve efficiency. By resetting the thresholds to reflect the workload characteristics, more updated pages can be retained in the buffer pool. This modification will effectively avoid the extra I/Os of reading inter-interest pages twice.

The DB2 system checkpoint, deadlock cycle, distributed process time-out, and the CF Castout settings are very important in the data sharing environment. The system checkpoint parameter, which determines the frequency at which DB2 registers its status into the BSDS, is very important. Because the BSDSs are local to the data sharing members, they should be tuned based on their workload. The DB2 member's unit of recovery is the last BSDS checkpoint. Keeping these settings similar with all the data sharing group members will minimize the potential for losing log records in the event of a disaster.

Deadlock and distributed processing timeout parameters are important because of the potential to hold extra locks in the local DB2 system and the CF. Having the deadlock and distributed time-out parameters set to a minimum will help flush deadlocked run units out of the Sysplex quickly, freeing both local and CF resources and locks. The Castout parameter is important because it tells DB2 when to "cast out" changed pages in the GBPs to DASD. Without proper attention to this parameter, the GBP will become saturated with intersystem interest pages. Setting the Castout parameter to trigger often will drive up the CPU and I/O resource consumption; however, by not triggering Castout often enough you risk having too little space

available in the GBPs. In extreme cases, too little room in the GBPs could cause write failures; DB2 will mark the intersystem interest pagesets for recovery after five failed attempts.

Using the IBM Relational Warehousing Workload with Type 2 indexes, and all the other various parameters correctly, only increases the processing overhead by a maximum of 17 percent. These IBM workload tests for up to 16 DB2 data sharing members were achieved through effective use of the CF lock and caching structures. The 17 percent overhead is caused by the incremental overhead of the CF communications and data sharing locking mechanisms. This figure is based on IBM's Relational Warehousing Workload studies of having data sharing among two to 16 data sharing group members. Because this workload is typical for the majority of data centers, your data sharing overhead will probably vary based on your workload's lock and caching requirements.

The overall impact of the CMOS CPU with 45MIPS engines versus the 60MIPS engines will depend on whether your shop's workload is I/O- or CPU-constrained. If your shop's workload is 100 percent CPU-bound, then your processing will be degraded. Most shops only have a portion of their environment CPU-constrained. Typical transaction workloads for the Sysplex environment will be minimally affected. Consider the components of a transaction: screen handling by CICS, communications through the network, file access, and DB2. Because CICS, VTAM, and file accesses are not usually CPU-constrained, CMOS CPU engine size might have its biggest effect on DB2. A typical CICS DB2 transaction only references multiple rows of information through the database using a minimum of CPU. Judging from these types of processing characteristics, the processing impact of the new CMOS CPUs will be minimal based on the CPU constraints of the components.

SHARING IS THE FUTURE

Sysplex, CMOS CPUs, and data sharing are the trends of the future. The opportunity to share MVS, Unix, and Windows NT workloads and data across the computing complex solves issues of availability and ever-growing capacity. Allowing incremental growth, global resources, and DB2 data integrity provides the growth and capacity for solving any business processing issue. Sysplex processing and DB2 data sharing are positioned to solve tomorrow's problems today.

Next time you see PC prices fall, think of the price/performance improvement for its big brother, the mainframe. Look fondly on that big water-cooled CPU with its bipolar technology: It got us here. Thanks to the mainframe, we can appreciate the new frontier of sharing everything at a much cheaper cost.

David Beulke has been building database environments for over 13 years and is currently the database administration manager for DB2 at the Spiegel Group Inc. He is an international speaker at technical conferences and has published several technical articles. He is also vice president and editor for the Midwest Database Users Group and chairman of the International DB2 Users Group 1997 Conference, to be held in Chicago on May 11-16.
