The Newspaper of Information Systems Management
June 6, 1994, Vol. 28, No. 23, 164 Pages, $6/Copy, $48/Year

COMPUTERWORLD

Vendors to simplify client/server prices

By Rosemary Cafasso

DAVE MARSHALL

Succumbing to user pressure, two heavy hitters in the client/server application market will soon simplify their pricing structures, which these days can require a rocket scientist to decipher.

Officials at Dun & Bradstreet Software and SAP America, Inc. last week said they are evaluating options that would focus more on value-based pricing and less on hardware and other criteria. Both companies confirmed they will announce changes within a few months.

Industry observers said such changes are critical in an industry where pricing is all over the map and can make software licensing a nightmare.

While most client/server software providers have their own particular twists, they typically base pricing on combinations of number of users, servers and

Client/server, *page 12*

IBM overshoots

Parallel System/390 pricing much too high, users warn

By Craig Stedman

Initial pricing of IBM's parallel System/390 hardware is higher than expected — about $27,000 per MIPS. Users last week said that price is too steep to make the systems a compelling alternative to traditional mainframes.

Industry observers were expecting prices in the $20,000-per-MIPS range for the CMOS-based parallel systems, which will start replacing IBM's ES/9000 mainframes during the next two years. Customers have been counting on the parallel systems to reduce the cost of mainframe computing to more Unix-like levels.

But analysts briefed by IBM in the past two weeks said the computer giant is concerned that cutting prices too close to the bone may weaken demand for water-cooled ES/9000s. That, in turn, could force more rapid price reductions on those machines than IBM executives could stomach.

Users willing to wait

Yet customers at several large mainframe shops said the cost of the parallel machines will have to fall to $20,000 per MIPS before they will open their checkbooks. The difference between the two prices is significant, amounting to $700,000 for each 100 MIPS of processing power.

George Sekeley, president of CSX Technology, Inc., the informa-

IBM, *page 16*

Buyers' market

IBM is "going to have to get warmed up to the reality of what the market will pay," said Rich Evans, an analyst at Meta Group, Inc. in Westport, Conn. "Hopefully this is just the first part of the cocktail party because if not, [the parallel machines] aren't very interesting." He said he has "seen some really ugly prices" on the parallel systems.

Object engine to pump up AppWare

By Elisabeth Horwitt and Melinda-Carol Ballou

Novell, Inc. plans to announce by the end of this month an object-based strategy for providing information systems managers with tools to design distributed applications for multivendor client/server environments.

The strategy will center on two upcoming announcements. A distributed version of Novell's AppWare application environment, expected out by year's end, will enable users to implement pieces of an application on different client systems.

Further out, Novell plans to announce a "universal object request broker" that will reside on a NetWare server, and possibly a UnixWare server, thus allowing distributed pieces of an applica-

Novell, *page 10*

Borland slide accelerates

'94 loss hits $70M; customer faith wavers

By William Brandel

"It was a tough, tough quarter," said Borland Chairman Philippe Kahn. And a pretty bad year, judging by last week's release of Borland's oft-delayed quarterly and year-end results.

After three delays, a struggling Borland International, Inc. last week posted a loss of $70 million on $393.5 million in revenue for its fiscal year 1994. That represents a 15.2% decrease in revenue from fiscal 1993, when the company posted $464 million in revenue and a $49.2 million loss.

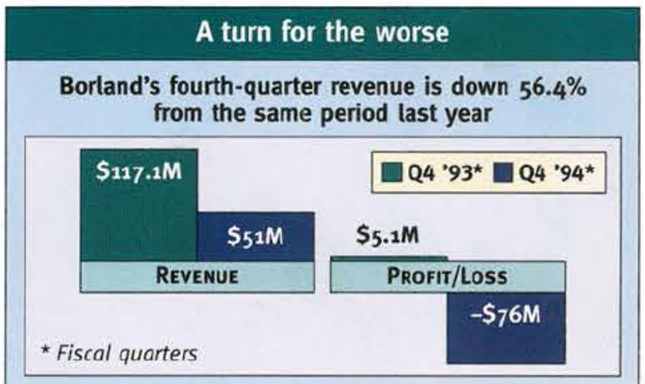
Profit long gone

The company also reported a loss of $76 million for the fourth quarter ended March 31, compared with a profit of $5.1 million for the same period last year.

Worse still, the company said it expects an operating loss for the first quarter of its fiscal year 1995, which ends June 30.

For the second quarter, Borland bases a large part of its earnings expectations on sales of dBase for Windows and a new DOS version of the product that will be introduced at PC Expo in New York later

Borland, *page 14*



Mainframe dilemma

IS brain drain depletes ranks

By Julia King

Longtime computing professionals searching for career stability are leaving corporate information systems departments and embarking on second careers far from the fast-changing arena of information technology.

Many of these so-called IS refugees are taking with them valuable mainframe expertise, leaving in the lurch hundreds of companies still very much dependent on mainframe applications. They are also leaving a raft of unfilled mainframe jobs in their wake, some IS watchers say.

"There is definitely a brain drain going on in IS, and [as a result], organizational memory is walking out the door," said Stewart Stokes, president of the Boston chapter of

Brain drain, *page 29*

ANDY FREEBERG

Joanne Ward *is preparing to leave her 30-year IS career for the comforts of her own pie-baking business. Job uncertainties drove her decision.*

Newspaper

#BXBBJFT************* 5-DIGIT 95006
[redacted]

8 ESSENTIALS of DISTRIBUTED RDBMS

As distributed computing gains momentum, relational database management system vendors are keeping pace — at least in terms of marketing claims and announcements of strategic direction.

Today's commercial database management systems do indeed have some distributed database features. However, consultant David McGoveran warns that they still have a long way to go before they are truly "DDBMSs," or distributed DBMSs (see story next page). Nonetheless, distributed database features are useful and becoming more robust. McGoveran walks through eight key areas for distributed databases and how leading vendors plan to meet the technical challenges. In some cases, he notes, support for a single feature can be the difference between the success or failure of a particular application.

Also in this section, the Buyers' Satisfaction Scorecard examines four leading distributed RDBMS candidates through the eyes of their users. And Firing Line looks at mainframe DBMS vendor Cincom and its move to distributed computing with Supra Server.

1. TRANSPARENCY
Where the data lives shouldn't matter

2. DATA INTEGRITY
Distributed transaction or request support required

3. DISTRIBUTED PROCESSING
Location-independent computing

4. ASYNCHRONOUS REPLICATION
Updates distributed data with low overhead

5. PERFORMANCE FEATURES
Fragmentation and other tricks

6. ADMINISTRATION AND SECURITY
A weak link in today's chain

7. AVAILABILITY AND AUTONOMY
Local autonomy keeps the database up and running

8. INTEROPERABILITY
Public standards, private standards and gateways

8 Essentials of Distributed RDBMS

Distributed not yet delivered

But there are some gains evident in current vendor approaches

BY DAVID McGOVERAN

Database management systems, like many technologies at the core of information systems operations, are undergoing a sea change. Organizations are attempting to adapt rapidly to successive waves of client/server computing, re-engineering and user data-access demands.

IS managers are under the gun to place the databases containing vital company information closer to users. The goal is to increase response time and simplify access for those who need the data most.

The dilemma is that data management, which includes guaranteeing data security and integrity, is most easily done at a central location where IS can provide standard procedures and methods for getting the job done consistently.

Unfortunately, the waves are irresistible, and IS organizations and DBMS vendors are beginning to use advanced concepts and technologies, known as distributed DBMS, to meet user needs.

Ideally, a distributed DBMS should appear to the user as a single, nondistributed system — even though it is physically located at different sites. Accomplishing this with today's products is a data management and system design nightmare.

Although today's DBMS products are still far from achieving the goal of a truly distributed DBMS, many of the new features they offer can reduce the number of sleepless nights for data managers and IS executives.

One compromise between IS' need to manage distributed DBMSs and the requirement that they be placed in multiple locations is a concept called asynchronous replication. It requires that a change made in one database be relayed to a second database in a certain amount of time set by the data administrator. This allows the data to be updated in a time frame that conforms to IS capabilities.

TALE OF THE TAPE

Two approaches to distributed databases

METHODS	TWO-PHASE COMMIT	ASYNCHRONOUS REPLICATION
CHARACTERISTICS	Peer-to-peer servers	Master/Slave servers
	Real-time updates	Near-real-time updates
	Synchronous updates	Asynchronous updates
	New transactions blocked during update	Continuous server operation
BOTTOM LINE	**IS has limited control over updates**	**Lets IS control timing and network traffic**

Source: International Data Corp.

Asynchronous replication is flawed because it does not provide real-time updating such as that enforced by the more rigorous, but more resource-intensive, two-phase commit.

Despite its flaws, replication is becoming the predominant distributed DBMS strategy for many organizations and vendors.

Apart from this major trend, vendors are offering different approaches to the key requirements of distributed DBMS. The following are eight areas that need to be addressed to accomplish a truly distributed DBMS.

Knowing how the vendors address them and how important each area is to you and your users is the key to selecting the right DBMS.

1. TRANSPARENCY

The ability to locate data in different databases on servers closer to users is central to relational DBMSs and doubly important for distributed databases. The location of the data, its physical storage format and the methods used to access the data (for example, which index to use) should be invisible to the user.

Today's DBMSs do not even come close to providing distributed transparency. The vendors use different dialects of SQL, the key data-access language for today's relational DBMSs. These dialects mix the **logical constructs** of the DBMS, such as how a table is defined and managed, and the **physical construct**, which defines how the data is stored. This mixture of definitions is an unfortunate and disabling violation of the relational model. The result is that access to and management of distributed data is more complicated.

Worse, failure to provide transparency precludes taking advantage of additional physical resources without modifying application code.

Most of today's products do not support an important element of transparency: the distributed system catalog. Without it, developers must embed in their applications references to table locations or, at the very least, must explicitly open connections to servers involved in a distributed transaction, such as Digital Equipment Corp.'s Rdb/VMS.

Exceptions to this generally involve the creation of synonyms for distributed tables (Oracle Corp.'s Oracle 7) or connection to a special server (The ASK Group, Inc.'s OpenIngres).

2. DATA INTEGRITY

Enforcing data integrity in a distributed environment requires support for a distributed transaction or distributed request. This capability allows a transaction consisting of multiple SQL requests to be processed at multiple locations, locally or remotely.

This requirement is obvious if integrity constraints involve distributed objects. If a local table is updated, but there is a referential integrity constraint involving remote tables, an implicit distributed transaction is clearly required. Products such as Informix Corp.'s Informix-OnLine, Oracle 7 and OpenIngres/Star place restrictions on how integrity constraints can be expressed, effectively providing no support for distributed integrity constraints.

3. DISTRIBUTED PROCESSING

Few DBMSs support arbitrary multisite read, multisite write and multisite read/multisite write transactions. However, a true distributed DBMS must support not only **remote request, remote transaction** and **distributed transaction** but also **distributed request** because anything less implies location dependence.

This type of request allows tables from multiple locations to be accessed using a relational join or union operation. Both distributed transaction and distributed request require support for the infamous two-phase commit or its equivalent; otherwise, global database consistency cannot be enforced in a distributed environment.

At this point, products such as Cincom Systems, Inc.'s Supra Server (see Firing Line, page 117), Informix-OnLine, Oracle 7 and OpenIngres/Star provide some support for remote request, remote transactions, distributed transactions and distributed request, while products such as Sybase, Inc.'s System 10 support only remote request and remote transaction.

4. ASYNCHRONOUS REPLICATION

In the last two years, so-called asynchronous replication has become increasingly important. Asynchronous replication provides the ability to propagate updates without the overhead of two-phase commit.

This involves two assumptions. First, the updates need not be propagated immediately, so all copies need not be simultaneously consistent. Second, the nature of the application is such that any violations of global database consistency will not be known immediately.

The validity of these assumptions can be extremely difficult to prove and depend on subtle details such as transaction mix, database design and application functionality. This feature cannot replace the need for distributed transaction and distributed request, although sometimes it reduces their inappropriate use.

Products suites such as Oracle 7, System 10, Informix-OnLine and OpenIngres offer replication capabilities, but these differ in many ways. Most apparent is the mixture of data sources and targets permitted and the granularity of the replications such as table, view and database.

For example, only OpenIngres Replicator was designed for peer-to-peer as well as master/slave configurations, a capability to be added to Oracle in Version 7.1.

Informix-OnLine supports full database replication, whereas Oracle 7, OpenIngres and System 10 support

RDBMS, *page 114*

Oracle rides reliability to the top

RDBMS vendor outscores rivals in security and integrity

COMPUTERWORLD BUYERS' SATISFACTION SCORECARD

By Derek Slater

"Open" relational database vendors are playing a high-stakes game of leapfrog, adding loads of features to each new release. In fact, the upgrades are coming out so quickly, you can't tell the players without a Buyers' Scorecard.

Oracle Corp.'s namesake relational database management system topped *Computerworld*'s user satisfaction poll with an overall rating of 72, ahead of cutthroat rivals Sybase, Inc.'s Sybase and Informix Corp.'s Informix-OnLine. The ASK Group, Inc.'s Ingres DBMS, soon to be acquired by Computer Associates International, Inc., finished fourth.

Key to Oracle's success was its lead in overall reliability, according to ratings from its installed user base. Reliability categories, including data integrity, backup and recovery functions and data security features, were users' No. 1 concern in evaluating RDBMSs, and Oracle's respondents gave the product a comfortable margin in each of those crucial areas.

Oracle also outscored competitors, notably in operation in a heterogeneous environment and support for complex tables. Oracle's weakness was in the area of technical service and support, where it tied for lowest user satisfaction.

Sybase's System 10 includes several notable improvements, including Control Servers for better backup and recovery and database tuning. Sybase's Replication Server is also a new addition to the product line. In the survey, Sybase stood out in ease of installation in client/server environments but lagged behind in quality of application development tools. Sybase's new Gain Momentum product is one effort to improve that.

Informix distinguished itself in technical support and documentation. It also earned the best satisfaction ratings in on-line transaction processing (OLTP) performance, probably because of Informix 6.0's multithreaded architecture, which lets the DBMS take advantage of multiple processor servers. However, Informix had the lowest score in performance in decision-support applications.

Ingres did not take the top score in any area, but it finished second in the important area of security features. Users also indicated that Ingres' application development tools are roughly on par with its competitors' tools. Ingres' overall score was particularly hurt by low scores in value for the dollar and cost of acquisition and maintenance. Ingres 6.4 is being replaced by OpenIngres, released too recently to make up the bulk of the respondents. OpenIngres adds object-oriented functionality and replication capabilities.

The chart at right shows the relative weight users place on each evaluation area. At this comparatively early stage of distributed computing, users said they are far more concerned with reliability and data integrity features than with performance issues.

Ratings in the charts below comprised several areas: OLTP performance, decision-support performance and support for complex tables. Users also rated products' interoperability, which comprised three subcategories: standards support, operation in a homogeneous distributed environment and operation in a heterogeneous distributed environment. ■

Slater is an assistant editor.

IMPORTANCE RATINGS

AREAS MOST CRITICAL TO USERS

8.8	RELIABILITY
	DATA INTEGRITY
	BACKUP AND RESTORE FEATURES
	SECURITY
8.2	VENDOR SUPPORT
	DOCUMENTATION
	VENDOR SERVICE AND SUPPORT
7.9	COST
	VALUE FOR THE DOLLAR
	ACQUISITION/MAINTENANCE COST
7.7	EASE OF USE
	EASE OF QUERYING
	EASE OF PROGRAMMING
	APPLICATION DEVELOPMENT TOOLS
	EASE OF INSTALLATION
	END-USER TOOLS

IMPORTANCE RATINGS ARE BASED ON A 1-TO-10 SCALE, WHERE 10 IS THE MOST IMPORTANT

ORACLE CORP. ORACLE

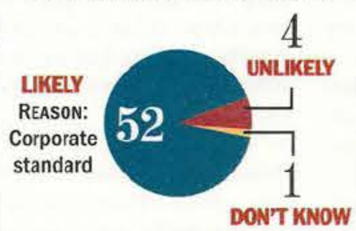
TOTAL SCORE 72

RESPONSE BASE: 57 USERS

7.9	RELIABILITY
6.7	SUPPORT
6.8	COST
7.0	EASE OF USE
7.2	INTEROPERABILITY
7.0	PERFORMANCE

RATINGS ARE BASED ON A 1-TO-10 SCALE, WHERE 10 IS BEST.

Would you buy this product again?



REASON GIVEN IS THE MOST FREQUENTLY STATED ANSWER

INFORMIX CORP. INFORMIX-ONLINE

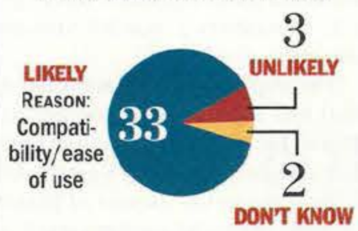
TOTAL SCORE 69

RESPONSE BASE: 38 USERS

7.3	RELIABILITY
7.1	SUPPORT
6.7	COST
6.8	EASE OF USE
6.8	INTEROPERABILITY
6.5	PERFORMANCE

RATINGS ARE BASED ON A 1-TO-10 SCALE, WHERE 10 IS BEST.

Would you buy this product again?



REASON GIVEN IS THE MOST FREQUENTLY STATED ANSWER

SYBASE, INC. SYBASE

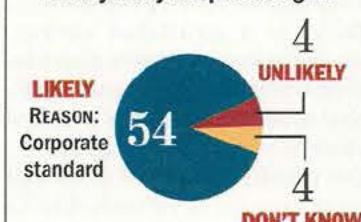
TOTAL SCORE 69

RESPONSE BASE: 62 USERS

7.3	RELIABILITY
6.8	SUPPORT
6.5	COST
6.7	EASE OF USE
7.0	INTEROPERABILITY
6.6	PERFORMANCE

RATINGS ARE BASED ON A 1-TO-10 SCALE, WHERE 10 IS BEST.

Would you buy this product again?



REASON GIVEN IS THE MOST FREQUENTLY STATED ANSWER

THE ASK GROUP, INC. INGRES

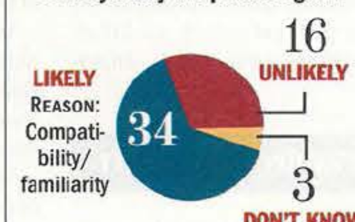
TOTAL SCORE 67

RESPONSE BASE: 53 USERS

7.2	RELIABILITY
6.6	SUPPORT
5.9	COST
6.5	EASE OF USE
6.8	INTEROPERABILITY
6.5	PERFORMANCE

RATINGS ARE BASED ON A 1-TO-10 SCALE, WHERE 10 IS BEST.

Would you buy this product again?



REASON GIVEN IS THE MOST FREQUENTLY STATED ANSWER

8 ESSENTIALS of DISTRIBUTED RDBMS

CONTINUED FROM PAGE 112

higher levels of granularity.

Details of design flexibility, installation and management facilities, error detection and recovery, data integrity enforcement and performance overhead all differ among these products.

5. PERFORMANCE FEATURES

There are several ways in which a DBMS might improve performance. Among them are fragmentation, replication and distributed query optimization.

Fragmentation is the ability to partition a table into subsets of rows or subsets of columns and place these subsets in different physical locations.

While some products support manual fragmentation at a single site, such as Informix-OnLine and Tandem Computers, Inc.'s NonStop SQL, distributed and automatic fragmentation is not supported today.

Replication, not to be confused with asynchronous replication, is the ability to maintain a set of distinct physical copies of a table that are automatically kept in synchrony by the DBMS, regardless of physical location.

Placement of fragments and replicas could be either manual or automatic, the goals being to minimize disk I/O and network traffic and increase parallelism.

Replication can be simulated in some products by creating triggers, although this may leave integrity holes if updates are not done via two-phase commit.

Distributed query optimization and execution takes best advantage of physical resources such as multiple CPUs, data location and network loads. When a DBMS supports multisite read, multisite write and multisite read/multisite write statements, optimization is much more complex than it is for single site versions.

Most products such as Oracle 7 simply divide a distributed query into single-site subqueries that are then individually optimized. This simply does not qualify as distributed query optimization. A higher degree of distributed query optimization is provided by products such as Informix-OnLine, OpenIngres and Supra Server.

6. ADMINISTRATION AND SECURITY

Systems administration and security facilities are particularly weak among today's DBMS products. While some support for remote start and stop is usually provided, coordinated start and stop of multiple sites usually is not.

Similarly, coordinated backup, restore and recovery are frequently weak. In general, these are needed to ensure global database consistency, as well as to ease systems administration.

Vendors sometimes appeal to **local autonomy** as the reason that current products are weak in this area. This simply evades the issue of how difficult it is to manage a distributed database environment when systems administration and security facilities are incapable of treating a distributed database as a single logical entity.

Failed distributed transactions are particularly problematic in today's products. While two-phase commit can cover most failure modes, all implementations have at least one failure mode that results in either halting the system or a lack of global consistency.

The means for detecting and recovering failed transactions are still primitive, often requiring detailed manual analysis of transaction and error log entries.

Deadlocks in a distributed environment can cause severe system degradation. Provisions must also be made for either global deadlock avoidance or global deadlock detection and recovery.

Vendors usually implement a time-out mechanism, incorrectly referring to it as a means of deadlock avoidance; it is not because it only ends a deadlock if one happens to exist, and a time-out cannot differentiate between long-running queries and a true deadlock condition.

Few, if any, DBMS products handle distributed deadlock avoidance, detection or recovery.

7. AVAILABILITY AND AUTONOMY

Local autonomy does not mean the local owner cannot temporarily delegate management to a remote site. Instead this delegation is an efficiency measure that should not prevent the local database from continuing operation in case it becomes disconnected from remote sites.

It should always be possible for local operations to be performed and controlled locally, although they may be performed and controlled remotely for efficiency reasons.

True support for local autonomy would permit a high degree of system availability because it implies the system is less vulnerable to a single point of failure and bottlenecks.

In order to avoid reliance on a particular remote site, the DBMS must support distribution of most functions including dictionary management, query processing, concurrency control and recovery control.

The implementation must not assume that any logical reference has any particular locality, a requirement that is routinely violated in today's products.

By limiting the impact of physical implementation and configuration, a DBMS administrator can curtail the need for planned system shutdowns.

Adding, changing (upgrading software versions or hardware, downsizing hardware and so on) or removing a site, node or database should not cause a disruption to the rest of the system, nor should creating or destroying fragments and replicas or destroying other database objects such as tables and indexes.

8. INTEROPERABILITY

At the platform level, interoperability involves hardware independence, operating system independence and network independence.

Regardless of the mix of hardware, operating systems or networks selected in a distributed environment, the DBMS software should work the same way. Similarly, if these are changed at any time, other nodes and sites should not be affected.

Most DBMSs have some behavioral dependence on the platforms, often in terms of the number of locks and files available and therefore the complexity of transactions that can be processed. In addition, certain functional features may not be available on all platforms; SQL syntax may differ slightly and storage management may cause differences in which errors can occur under a given circumstance.

DBMS interoperability is intended to provide uniform access to multiple DBMS products in a single distributed environment.

Interoperability has been addressed through a combination of public standards such as ANSI SQL, X/Open Co.'s Relational Database Architecture and the Open Software Foundation's Distributed Computing Environment; proprietary standards such as Microsoft Corp.'s Open Database Connectivity, Oracle's Glue, Borland International, Inc.'s Integrated Database Application Programming Interface and IBM's Distributed Relational Database Architecture; and gateway technology.

Almost all DBMS products support standards in an attempt to provide a common set of syntax or formats and protocols, but these usually do not address behavior.

Also, they tend not to address product-specific extensions, providing only a least-common-denominator approach to interoperation.

By contrast, gateway technology, such as that of ASK, Oracle and Sybase, attempts to support interoperation by translating syntax, formats and protocols among DBMS products.

However, even gateway products typically restrict the degree of interoperation, often failing to support all SQL constructs and data types or not addressing differences in error and recovery management. ■

DISTRIBUTED DBMS TERMINOLOGY PRIMER

It's even more complicated than you think

Local autonomy means that local database data (and by implication, data definitions, authorizations, integrity constraints, etc.) is locally owned and managed.

Physical construct is the location of the data, its physical storage format or the methods used to access the data.

Logical construct is the definition of a table, column or integrity constraint. The way such a definition is implemented is a physical construct.

Remote request support allows a single SQL request to be processed at a single remote location.

Remote transaction capability allows a transaction consisting of multiple SQL requests to be processed at a single remote location.

Distributed transaction support allows a transaction consisting of multiple SQL requests to be processed at multiple locations (local or remote). Each SQL request can be processed only at a single location, but different requests within the same transaction can be processed at different locations.

Distributed request allows a transaction consisting of multiple SQL requests to be processed at multiple locations (local or remote). Each request can be processed at multiple locations. Of these last four items, a distributed request is the most difficult to implement.

McGoveran is president of Alternative Technologies, a Boulder Creek, Calif., relational database and client/server consulting firm founded in 1976. He is publisher of the Database Product Evaluation Report Series.