Patricia Seybold Group

Editor-in-Chief
Michael A. Goulde

INSIDE

OPEN INFORMATION SYSTEMS

Guide to Unix and Other Open Systems

Vol. 8, No. 3 • ISSN: 1058-4161 • March 1993

Sybase System 10

Can It Manage Enterprise Data?

By Judith R. Davis

IN BRIEF: As distributed client/server computing gains momentum, vendors are competing furiously to own a piece of the action. Prime real estate includes the relational database management system (RDBMS), application development tool, distributed management system, and, ultimately, the client/server infrastructure of choice. Sybase Incorporated is targeting all of the above with its next generation of server technology, Sybase System 10. The company had a head start on some of its competitors by virtue of the fact that its RDBMS was designed from the beginning with a client/server architecture as well as with features to address major customer requirements: integrity, performance, and availability. The company subsequently published both its client and server APIs. The underlying philosophy is to allow the customer to integrate into the Sybase environment whatever client/server functionality is appropriate to meet its unique distributed computing needs. System 10 is geared to meet enterprise-wide client/server requirements, and we think Sybase has a shot at being successful. *Report begins on page 3.*

The Market Imperative

"Proprietary" Continues to Lose Meaning

DID UNIX BRING down IBM's CEO, John Akers? Has open systems "commoditized" the computer industry to the extent that price and service are now the only means of competing? Are open systems a cancer that is slowly killing the computer industry, or a plunger that is cleaning out the clogged sink? And will nothing short of a complete reinvention in a completely different mold save IBM?

IBM remains one of the few companies that has not abandoned its proprietary networking and operating system products and made a complete, across-the-board commitment to Unix. Even Digital Equipment is now beginning to talk about its 64-bit, OSF/1-based, "Unified Unix" as a cornerstone for its future development. Ironically, only IBM's former partner, Microsoft, continues to march to the "other than Unix" drummer. Novell dropped out of that parade with its intended acquisition of USL.

The economics of Unix-based systems says that proprietary technology sustains incremental value over commodity technology only if customers receive incremental benefits to which they assign value from the proprietary technology. In other words, why pay IBM a premium for a proprietary operating system if Unix, which is essentially a commodity operating system, can serve just as well?

This argument even holds when the proprietary operating system has been made compliant with open systems standards. Whether it is MVS or OS/400 from IBM, OpenVMS from Digital, or any of the other POSIX-compliant and XPG-branded operating systems, we have reached a point where most applications do not gain enough added value by running on a proprietary operating system rather than on Unix to warrant the added cost. Even when the price of third-party layered software products to make Unix functionally equivalent to MVS is added, the total cost is still less.

As a result, many in the industry have assumed that proprietary operating systems will go away, virtually overnight. However, there is an alternative available to the vendors of these standards-compliant, non-Unix operating systems that can both prolong their viability in the market and make them competitive. If the vendors accept the fact that they are competing with commodity products and price their products accordingly, then proprietary immediately stops being at a disadvantage vis à vis Unix. Since many downsizers are buying Unix simply to save money, IBM, Digital and others could retain customers, neutralize one of the principal arguments against proprietary, and compete more effectively against USL's Unix with a new pricing model.

The strategy could result in Digital extending the life of VMS as its cash cow and IBM repositioning MVS as guardian of legacy applications and data on a new generation of lower-cost (much lower!), microprocessor-based, ES9000 architecture systems. We bet IBM could do an eight-way RISC-based machine that could run MVS and its applications for about the same cost to customers as an eight-way Hewlett-Packard or Digital RISC-Unix box—an order of magnitude below the cost of current mainframes.

As long as everyone else is downsizing, why wouldn't IBM take a shot at surviving by downsizing the mainframe itself? After all, it already has a MicroChannel card that is the equivalent of a 4381 mainframe, and that is with gate array technology. With CMOS-based microprocessors, IBM could deliver a very competitive product.

With "proprietary" but standards-compliant operating systems priced directly against Unix, not only would one of the market's primary motivations behind the widespread migration to Unix be curbed, but, more importantly, the market would have to become a lot more precise about the definitions of *proprietary* and *open systems*.

OPEN INFORMATION SYSTEMS

Editor-in-Chief
Michael A. Goulde

MCI:
MGoulde
Internet:
mgoulde@mcimail.com

Publisher
PATRICIA B. SEYBOLD

Analysts and Editors
JUDITH R. DAVIS
STANLEY H. DOLBERG
DAVID S. MARSHAK
RONNI T. MARSHAK
JOHN R. RYMER
ANDREW D. WOLFE, JR.

News Editor
DAVID S. MARSHAK

Art Director
LAURINDA P. O'CONNOR

Sales Director
PHYLLIS GUILIANO

Circulation Manager
DEBORAH A. HAY

Customer Relations Manager
DONALD K. BAILLARGEON

Patricia Seybold Group
148 State Street, 7th Floor,
Boston, Massachusetts 02109

Telephone: (617) 742-5200 or
(800) 826-2424
Fax: (617) 742-1028
MCI: PSOCG
Internet: psocg@mcimail.com
TELEX: 6503122583

Open Information Systems (ISSN 1058-4161) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold Group, 148 State Street, 7th Floor, Boston, MA 02109. Second-class postage permit at Boston, MA and additional mailing offices.

POSTMASTER: Send address changes to *Open Information Systems*, 148 State Street, 7th Floor, Boston, MA 02109.

Sybase System 10

Can It Manage Enterprise Data?

Business and Technical Issues: The Sybase Perspective

Sybase Is a Client/Server Company

From Sybase's perspective, *client/server* means allowing the customer to use the most appropriate tool on the most appropriate platform to access any data source—essentially, the ability to glue pieces of any environment—homogeneous or heterogeneous—together as needed. This is one reason why client/server has been so successful and why there is so much emphasis on client/server applications today. Client/server applications benefit everyone if the architecture is implemented correctly. Sybase System 10 addresses an enterprise-wide, multidatabase, multiplatform environment.

Sybase has never been driven by technology, feature check-offs, or a compulsive need to adhere to industry standards in designing its products. The competition has often emphasized the fact that Sybase is behind in areas such as distributed database or full compliance with the SQL standard. But instead of rushing to implement features that come out in competitive products—e.g., an automatic, server-enforced two-phase commit protocol for distributed transactions—Sybase has kept a clear focus on identifying and solving real business problems for customers. In our opinion, Sybase has done a better job than any of its competitors in extending its products to address an entire *business* transaction, not just transactions at the *database* level. The introduction of the Sybase Open Server and Open Client APIs in 1989 was a major step in this direction. Sybase has since built its mainframe products and its non-Sybase data gateways using Open Client and Open Server. The System 10 components continue to move the company closer to its goal of supporting an enterprise-wide client/server environment. The extent to which Sybase has succeeded in its mission can be seen in Illustration 1.

According to some market researchers, the client/server software market will increase tenfold to $7.5 billion in 1996. Sybase states that a vast majority of its 6,500+ sites are already building industrial-strength client/server applications. The key motivations for adopting client/server computing are: from the departmental perspective, developing a competitive advantage; and from the central IS perspective, reducing the long-term costs of information systems by deploying cost-effective systems that leverage existing investments. Elements here include the fact that information has become a strategic asset; systems are evolving from supporting hundreds to thousands of users who are geographically dispersed and who need mainframe integration; databases are becoming increasingly heterogeneous and are increasing in size from hundreds of MBs to GBs.

Client/Server Is Evolving to an Enterprise Level

Sybase sees the client/server computing environment evolving from department-oriented, LAN-based systems, where client/server got its start, to complex, enterprise-wide systems that must accommodate a wide variety of hardware platforms, operating systems, and data sources. Large organizations are moving away from a centralized, host-based perspective toward an environment in which individual business units or departments create their own real-world, mission-critical applications and data. Yet there continues to be a dependence on legacy, mainframe-type production applications for basic business information. Users with PCs or workstations on the desktop often need to access mainframe production data, or extracts thereof, on a near-real-time basis for transaction processing or for ad hoc analysis. Thus, there is a need to integrate departmental "line of business" applications and central MIS production applications in a single environment. For some organizations, this means

continuing to run production applications on the mainframe while providing easy access to the data from the rest of the organization. For others, it means moving the applications and data off the mainframe to smaller, more easily managed platforms that are just as powerful as the mainframe. This is referred to as downsizing, rightsizing, or simply: "Let's do it effectively but not pay an arm and a leg to buy, operate, and manage it."

An Evolutionary Approach

Users want to move in this direction in an evolutionary way while providing instant access to information for the user who needs it. This ensures good business decisions in a context of orderly change—a difficult achievement under any set of circumstances. Sybase System 10 is not just a new release of a relational database engine; it is an overall architecture designed to manage this level of change with consistency. The five major requirements Sybase has identified for enterprise-wide client/server computing and the Sybase System 10 components designed to meet each requirement are:

- *Low risk.* Organizations are looking for proven products that can be extended to accommodate new technologies over time at low risk. Here, Sybase offers its flagship product, SQL Server Release 10, a high-performance, relational DBMS (RDBMS) that runs on open-systems hardware and offers portability of both applications and data across heterogeneous systems. (The current version of SQL Server is 4.9.1.)

- *Scalability.* Sybase's answer to the need for scalability is Navigation Server 10, a large-scale system capable of running mainframe-class applications without requiring a transaction monitor. Navigation Server initially runs on both massively parallel and SMP platforms from NCR.

- *Interoperability.* Support for interoperability among heterogeneous data environments is provided by OmniSQL Gateway. Since Sybase cannot expect to own the entire database environment, the user needs a way to transparently access data stored in any data source. Sybase will provide turnkey access to specific data sources; others can be added by the customer.

- *Reliability.* Replication Server 10 supports autonomy for business units/departments while ensuring that shared data are available in a near-real-time mode.

- *Control and management.* The Sybase Control Servers encompass a suite of tools to globally manage the components of a client/server network.

Sybase plans to introduce all of these products by the end of this year. Sybase's overall objective with System 10 is to support large databases, stable distributed computing environments, and heterogeneous, multidatabase applications. It is a large first step that will be expanded over the next year or two with standards-based enhancements.

In this article, we first describe the underlying architecture for System 10—the Sybase Open Client and Open Server APIs—and why these are so important to Sybase. Then we address the new functionality of each of the System 10 components followed by a brief summary of Sybase's tools strategy. While a detailed competitive analysis is beyond the scope of this article, we do highlight some key differences between Sybase's strategy and product plans and those of its primary competitors. Finally, we assess the fundamental question: How close is Sybase to effectively managing an enterprise-wide client/server environment?

Sybase at a Glance

Company	Sybase, Inc.
Corporate headquarters	6475 Christie Avenue Emeryville, CA 94608 (510) 596-3500
Founded	1984
Product first introduced	1987
Latest release	SQL Server Version 4.9 (introduced September, 1992)
Ownership	Public (NASDAQ)
Fiscal year	January 1 - December 31
Revenues 1992 1991 1990 1989 1988	$ 264.6 million 160.6 million (includes Gain Technology) 103 million 57 million 24 million
Net income 1992 1991 1990 1989 1988	$ 23.7 million 7.8 million (5.9 million) Loss 1.7 million (0.6 million) Loss
Geographic breakdown of revenues: Domestic International	 22 percent 78 percent
Breakdown of revenues by channel: Direct sales VARs OEMs	 90 percent 10 percent Not applicable
Breakdown of revenues—licenses vs. services Product (license) sales Services (support, consulting, education)	 70 percent 30 percent
Breakdown of license revenues by product type Database, networking, tools Applications Other	 100 percent (50% SQL Server, 35% tools, 15% connectivity) 0 percent 0 percent
Breakdown of license revenues by platform: Unix IBM mainframe Digital VAX Other proprietary midrange Desktop (DOS, OS/2, Novell, Macintosh)	 70 percent 5 percent 10 percent 1 percent 14 percent (DOS, OS/2, Novell)
Distribution channels U.S. sales offices International sales locations VARs	 27 5 subsidiaries, 26 distributors Sybase has over 175 software partners in its Synergy and VAR programs
Installed base: Number of customers Number of sites Number of licenses Number of users	4,300 6,525 16,950 (servers) Not available
Number of employees	Over 1,700 worldwide

Illustration 1. Sybase's ability to introduce products that focus on real customer needs has enabled the company to increase revenues at an impressive pace.

Sybase Open Client/Open Server Architecture

Making the Leap to the Next Generation

Sybase System 10 represents Sybase's next generation of server products, all of which will be delivered in 1993. System 10 is a modular family of products that address an enterprise-wide, multidatabase, multiplatform environment.

Externalizing the SQL Server APIs

One of the most important developments in the Sybase product line was the introduction of the Open Client and Open Server APIs. These are the APIs used within SQL Server itself; Sybase externalized them to allow extensions to the SQL Server architecture by both Sybase and third parties. These APIs have become the underlying foundation for all Sybase products, as well as separate products in their own right. All of the System 10 servers are built on a new version of the Open Server Toolkit, and Sybase is releasing a new version of Open Client with SQL Server Release 10. Sybase's strategy here is fundamentally different from that undertaken by its competitors, although Oracle Corporation (Redwood Shores, California) is beginning to emulate Sybase's approach to some extent with its Oracle Glue and Open Gateway directions.

The Ability to Integrate Non-Sybase Clients and Servers

With Open Client and Open Server, developers and customers are able to integrate non-Sybase front ends and servers into the Sybase environment. These APIs provide a single interface for integrating any component into the Sybase framework, or infrastructure, giving Sybase users access to heterogeneous products. (See Illustration 2.) Thus, customers do not have to wait for Sybase to provide turnkey access to tools, servers, or data.

Open Client Is Sybase's Front-End API

Sybase has always had a client API, known as Database Library, or DB-Library. Today, Open Client consists of DB-Library and Net-Library. DB-Library is a call-level interface (CLI) that provides a set of function calls for an application to send requests to either the Sybase SQL Server or Open Server, and to receive the corresponding data or messages in response. Requests can be in the form of Database Remote Procedure Calls (database RPCs), SQL text, or unformatted command strings in any language. Net-Library supports multiple network protocols and provides protocol independence. Sybase makes the Net-Library API available to some business partners for development of new Net-Libraries, and plans to publish it in the future.

Open Client 10 Will Support Other APIs

With System 10, Open Client will become an umbrella name covering a *family* of interfaces to the Sybase server APIs: DB-Library, the Sybase embedded SQL precompilers, Microsoft's Open Database Connection (ODBC), and Client-Library (CT-Library). CT-Library is the new version of DB-Library that includes, in addition to DB-Library, support for all the other interfaces.

Sybase extended DB-Library in the form of CT-Library to take advantage of cursors, new ANSI SQL syntax (all added to SQL Server Release 10), and the ability for a client to initiate multiple actions on a single connection, as well as other additional APIs. Embedded SQL and ODBC resolve themselves down to CT-Library calls; DB-Library calls are simply passed through. Sybase plans to implement an ODBC driver by the end of this year. (Note: Microsoft's ODBC emulates cursors in the client; Sybase's version will take advantage of native cursors in SQL Server for better efficiency.) This architecture gives Sybase tremendous plug-and-play functionality and makes it a bona fide client/server framework vendor. Currently, Open Client only supports Sybase's own Tabular Data Stream (TDS) messaging protocol, but support for other formats and protocols (FAPs), such as RDA and SQL Access Group's FAP, are planned.

We expect Sybase to add support for IBM's Distributed Relational Database Architecture (DRDA) and Borland's Integrated Database API (IDAPI—see *Open Information Systems*, Vol. 7, No. 12) to Open Client in the future. Sybase is already committed to DRDA, and

IDAPI will give Sybase a connection to the millions of Borland desktops and Paradox/dBase databases out there.

Open Server Can Integrate Nonrelational Data Sources

The most significant aspect of Open Server is the fact that it can interface to any server and is not limited to SQL-based servers. Sybase views this as essential to its OLTP customers since online applications are very demanding, and one size does not "fit all." Open Server can be used to connect a gateway, a DBMS, or any other server application to Open Client applications. The developer uses a consistent set of routines to access a wide range of servers and data environments. Open Server masks the complexity of the server, the network, and the data. Sybase was the first database vendor to open up its server interface in terms of licensing the source code and allowing the interface to be used with any type of server. Open Server is also a call-level interface that can accept requests from client programs running Open Client. Open Server cannot issue requests to other servers except through Open Client.

Open Server 2.0 included event alerters and a multithreaded architecture. System 10 adds internationalization, cursor support, full integration with Net-Library (support for multiple, simultaneous Net-Libraries), and text and image data support.

Sybase Open Client/Open Server Architecture

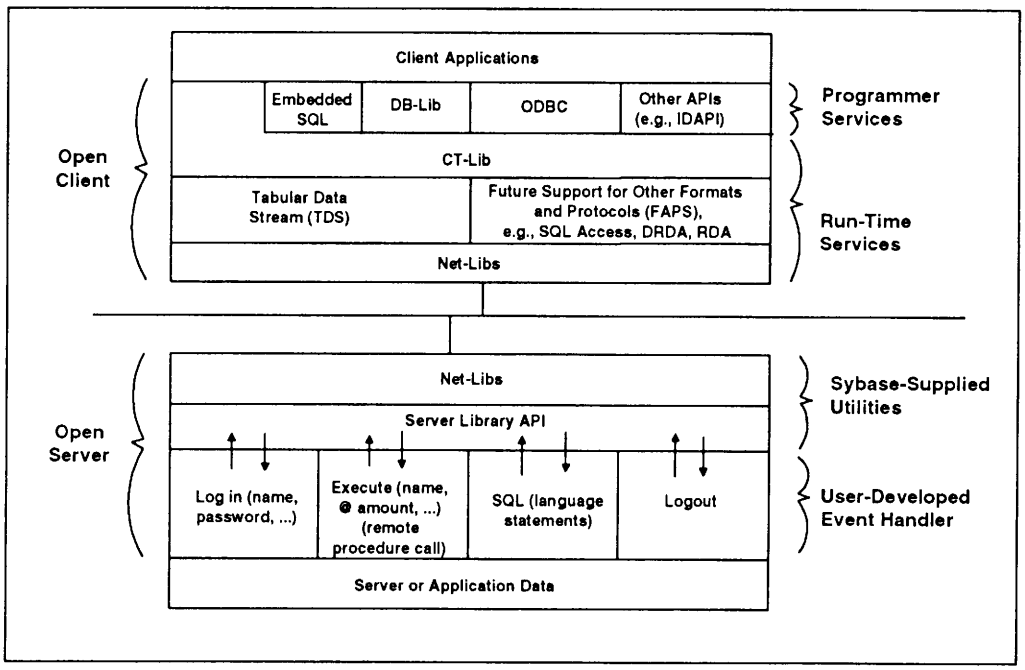

Illustration 2. The Sybase Open Client and Open Server provide a network interface and API libraries. On the server side, the user must write an event handler so the server can understand requests coming from the client and the client can understand data coming back from the server. Open Server is the foundation on which all of the System 10 servers are built. With Open Client 10, Sybase is introducing an architecture that will support a variety of front-end APIs, both Sybase proprietary (e.g., DB-Library) and standards-based (e.g., ODBC) APIs. This gives the customer a choice of applications and tools that can fit into the Sybase client/server framework.

Broad platform support for the Sybase APIs is crucial because these are a basic requirement for porting System 10 components. Open Client already runs on a wide variety of Unix platforms plus DOS, Windows, OS/2, Macintosh, VAX/OpenVMS, HP MPE, and IBM MVS/CICS. Digital Alpha/OpenVMS is in beta test and Novell NetWare is in development.

Sybase Open Client/Open Server Architecture

Open Server runs on many Unix platforms (DG AViiON, Digital RISC/Ultrix, HP 9000, IBM RS/6000, NCR System 3000, Pyramid, Sequent, SunOS, Sun Solaris), VAX/VMS, and IBM MVS/CICS. Alpha/OpenVMS and OS/2 are in beta test; Novell NetWare and Silicon Graphics are in development.

Oracle and Sybase Both Want the Client/Server Infrastructure Market

Oracle is following Sybase's lead on the client API front with the recently announced Oracle Glue API. Our take is that Oracle Glue is comparable to Open Client's CT-Library layer. An application is written to the Oracle Glue API, and Oracle then does the translation to the underlying API/transport: ODBC, Oracle's SQL*Net, etc. The essential difference between the two architectures is that Oracle Glue rides on top of all other APIs, whereas, in Open Client Release 10, the APIs ride on top of CT-Library. An application can be written to a non-Sybase API, such as ODBC, but can take advantage of the Sybase client/server infrastructure. Open Client no longer forces the application to write to the Sybase DB-Library API. It will accommodate other APIs as well. (Either Sybase or a third party will have to support the translation of non-Sybase interfaces to CT-Library.)

Customers Buy in at Different Levels

On the front end, Oracle is taking a different approach from Sybase. Oracle Glue is a "universal interface" intended to appeal to large MIS shops that don't want to worry about which API to embrace. Oracle's message is "write to Glue and leave the rest to us." The application developer is forced to write to Oracle's API in order to use Oracle's client/server framework. While Sybase is in the same position today, Release 10 of Open Client will open the Sybase architecture to other APIs. The basic difference is the level at which the developer buys into the framework. Oracle wants to capture customers at a higher level, which makes the customer more dependent on Oracle's support for multiple APIs. Oracle, for example, has stated that it will provide support for the Sybase Open Client underneath Oracle Glue, but not until the second release. Sybase, on the other hand, will provide access to Oracle data via any of the supported APIs and its OmniSQL Gateway (turnkey) or Open Server (customized).

Oracle Is Still Missing an Open Server Product

Another major difference is that Oracle does not yet have an equivalent to Open Server on the back end, so the customer is limited to accessing data sources/servers supported by Oracle Glue. The first release of Glue, due out in April, will access Oracle Versions 6 and 7, all data sources supported by Oracle SQL*Connect gateways (DB2, SQL/DS, Tandem's NonStop SQL, IMS, RMS, and HP's TurboImage), Oracle*Mail, and Sharp Wizards. Oracle promises its Open Gateway products (Procedural Gateway and Transparent Gateway Developer's Kit) by this summer. The second release of Glue will add access to ODBC, non-Oracle mail servers (via VIM, MAPI, and MHS), and Sybase. Oracle Glue is only available on Windows 3.x initially, which limits its flexibility. Oracle does plan to port Oracle Glue to other platforms. The Macintosh will follow Windows by six months or so, followed by other platforms, such as Unix, Windows NT, Pen Windows, and OS/2 (but not DOS).

SQL Server Release 10: Adding Needed Enhancements

Sybase SQL Server continues to use a Virtual Server Architecture (VSA), a multiserver architecture that can take advantage of multiple processors on SMP machines. Some of the features of VSA include dynamic load-balancing among servers, the ability to define the number of processors that Sybase can use (other CPUs are free to service other applications), and the fact that a server does not have to wait if a task is interrupted (the task is put back on the queue and the server takes another; Informix Software Incorporated (Menlo Park, California) will support this in OnLine 6.0; Oracle7 servers wait if a task is interrupted).

With Release 10, Sybase is adding some much-needed enhancements to its flagship SQL Server DBMS engine. These new features will make SQL Server competitive in areas that

major rivals Oracle and Informix are stressing in the feature/function wars. Here are some highlights. (See Illustration 10 for a comprehensive features chart.)

SQL Server Will Comply with SQL92 Entry Standard

ANSI SQL COMPLIANCE. Sybase has been behind in compliance with the ANSI SQL89 standard. SQL Server's inability to support database cursors has been perhaps the most obvious missing feature. Although Sybase has supported cursors in its front-end tools, lack of support in the server has made life difficult for Sybase application developers and put Sybase behind the curve in releasing embedded SQL precompilers. The existing precompilers allowed the creation and manipulation of cursors, but cursors actually have been simulated in the client and not supported natively in the server.

This will all change with Release 10. SQL Server will be fully compliant with Entry Level SQL92, including native cursor support, declarative integrity constraints, new data types, and additional syntax. (See *Unix in the Office*, Vol. 6, No. 11, for more details on the SQL standard.) SQL92 Entry Level encompasses the SQL89 standard, the integrity enhancement, and a few additional features such as SQLSTATE. Pre-release versions of SQL Server 10 and the SQL precompiler for C have passed the NIST test suite for compliance with ANSI SQL89, the integrity enhancements, FIPS flagger, and sizer. Sybase expects to pass the new tests for SQL92 when they are available.

Support (Finally) for Database Cursors

Database Cursors. Release 10 will support all fundamental cursor operations defined in the SQL89 standard (declare, open, fetch, positioned update/delete, and close) as well as some significant extensions: the ability to declare a cursor as updatable or read-only, an array interface, and the option to keep cursors open across transactions. Cursors will be supported in the server, stored procedures, SQL precompilers, and CT-Library.

Integrity Constraints Can Now Be Declarative

Declarative Integrity. Sybase has always supported the implementation of integrity constraints using triggers and stored procedures (for enforcement of any type of integrity, such as referential integrity or business rules, using *procedural* code), rules (for field-level validation), defaults (to specify field default values), and domains (extensions to base data types, which Sybase unfortunately calls "user-defined" data types). With Release 10, SQL Server will also offer *declarative* integrity constraints, including primary and foreign keys, unique column constraints, check constraints, and user-defined constraint violation messages. Each constraint can be named, and violation messages can be stored in multiple languages for international portability.

Other SQL Enhancements. SQL Server will now support other SQL syntax as well. Some syntax is new and some simply implements existing Sybase functionality using standard language.

The New Backup Server Is Bundled with SQL Server 10

SQL Server Release 10 includes several system management enhancements. The most important of these is probably the new Backup Server, which will be bundled with SQL Server 10. Database backup-and-load is one area in which Sybase has had performance problems. Sybase depends on its own dump and load utilities, which lack speed and may hamper the server's ability to manage very large databases. With the Backup Server, Sybase will be able to do high-speed backups of multi-gigabyte databases, adding capacity to the local SQL Server. Instead of the SQL Server handling backups and data loads as ordinary tasks to be interleaved with online user requests, the Backup Server offloads much of these tasks from the SQL Server.

Other important benefits of Backup Server include its ability to centrally manage backup for multiple servers across the network, to schedule and perform any backup tasks without operator intervention, and to back up data automatically based on configurable thresholds. Backup Server represents Sybase's first steps toward a "lights out" operational mode and an extension of its ability to manage a distributed system.

Backup Server will support multiple backups stored on one volume of media, a single "striped" backup spanning multiple volumes, a wider range of backup devices (virtually any supported backup device/media on a platform), the ability to write and restore multiple striped backups simultaneously (e.g., the server can read multiple database disks and write to multiple backup devices in parallel), elimination of the need to initialize each data page when loading data, and backing up and restoring data across the network between heterogeneous systems (one benefit is the ability for multiple SQL Servers to share one high-performance backup device).

Log Thresholds Will Provide Helpful DBA Triggers

Another feature that DBAs will welcome is the Log Space Monitor. This will enable the DBA to specify thresholds in the transaction log at which a stored procedure will automatically execute. The stored procedure might trigger a log backup before the log gets too full and halts database processing. The stored procedure could also execute a remote procedure call that notifies a beeper. Sybase has also included a "last chance" threshold to prevent the log from becoming so full that there is insufficient space even to log that the log is being dumped.

Other Significant Features

SECURITY. SQL Server 10 is also designed for compliance with the C2 standard for secure databases. A B1 level of security will be optional, and availability will follow C2 by six to nine months.

AUDITING. Release 10 includes a flexible auditing function that can be dynamically configured to record server-level events, access to database objects, or actions of individual users.

Replication Server

Replication vs. Two-Phase Commit

Sybase has long maintained that a two-phase commit (2PC) protocol for managing distributed (multisite) updates solves only part of the business problem customers face today. The bigger issue is the need to share the same data among multiple sites in an organization while ensuring the integrity and consistency of the data at every location. Two-phase commit is a yes/no, all-or-nothing approach that requires all components of the distributed network to be available. A transaction is usually rolled back if one site is unable to commit. From Sybase's perspective, 2PC presents performance problems with the network traffic required *during* the transaction (which affects transaction throughput), and makes an unrealistic assumption about network reliability.

The Replication Server is Sybase's strategic answer to this problem. The company has spent over two-and-a-half years developing Replication Server in conjunction with specific customers, many of whom have spent time and energy building their own customized replication services. Sybase sees replication as a real business problem.

A Replication Example

Let's look at an example where replication would be an appropriate solution. A company has two major locations, one in New York (eastern region) and one in Los Angeles (western region). Each office is responsible for maintaining its own customer records, but both offices need access to all customer records for reports and inquiries. The New York office would replicate its customer records to Los Angeles, and Los Angeles would send copies of its records to New York. Sybase Replication Server would make this transparent, automatic, and near-real-time.

Another example would be an organization that currently makes copies of a central database on multiple remote sites overnight. With Replication Server, the copies could be updated during the day as transactions to the central site occur, providing more up-to-date information in the remote sites and eliminating the need to coordinate overnight data dumps.

Replication Server

Sybase Replicates Transactions, Not Whole Tables

Replication Server replicates *transactions*, not *tables* across nodes in the network. Only the rows affected by a transaction at the primary site are sent to remote sites by the Replication Server. (See Illustration 3.) Whenever an update occurs to the primary copy of a replicated table, the Log Transfer Manager (LTM), a program that monitors log activity at the primary site, passes the changed records to the local (primary) Replication Server. If the rows are in tables that are replicated, the Replication Server stores the rows and passes them on to the appropriate distributed Replication Servers. At the remote sites, these rows are then applied to the replicated copies. The Replication Server responsible for each remote site makes sure the transactions are executed in the right order on the copy and applies the appropriate transaction management.

Updates from Remote Sites with Asynchronous Stored Procedures

Sybase also now supports an asychronous stored procedure that can be used by remote sites to update data indirectly. The user at the remote site issues an update via a stored procedure. The stored procedure is passed to the local Replication Server by the LTM and then to the primary Replication Server, which executes the stored procedure at the primary data site. The change is then replicated back to the copy and the remote user. An example here would be a travel agent in a local office changing the address of a client. The travel agent generates the transaction from a copy of the client record, and the update is propagated through the primary data site and back to the copy. Depending on the network configuration, the change could be reflected in the copy in as little as a few seconds. If a component of the network is down, the stored procedure would be queued and then processed after the failed component is restored. The travel agent, in the meantime, can go on and do other tasks due to the asynchronous nature of the stored procedure. This is a new mechanism for Sybase, and it allows Replication Server to perform functions that many real-world customers need and have had to build their own systems to accomplish.

Replication Can Be Near-Real-Time

One important point here is that the replication process is initiated by a database event—a transaction—at the primary data site. It is not scheduled and done at a particular interval for the database as a whole, as in other replication schemes. Thus, Replication Server can be used to replicate data on a near-real-time basis.

Access Levels Are User-Defined

It is also important to understand that there is nothing inherent in Replication Server that makes replicated data read-only. The user is responsible for establishing security levels over replicated data. The plus side is that this gives customers flexibility in designing sophisticated, customized replication schemes. In the example we used above, all of the customer records could be in a single table at both sites; the difference would be that, in New York, the eastern region customer records would be primary data and the western region customers replicated. In Los Angeles, the reverse would be true. The DBA would create the necessary permissions to coordinate updating replications in each location.

Sybase provides an LTM for SQL Server with the Replication Server. Sybase will make LTM design and interface requirements available to customers and third parties for access to non-Sybase data. Replication Server doesn't assume SQL and can be used with non-SQL data sources. It would certainly make sense for Sybase to make turnkey LTMs available for products like Oracle and DB2 so these could function as primary data sources.

Replication Server Has Significant Benefits

HIGH AVAILABILITY. Replication Server has a store-and-forward capability to accommodate the reality that network connections and network components do fail. Replication ensures that you are only updating the data in one place at one time; there is no "all or nothing" requirement across nodes. One Replication Server can manage multiple data nodes, but the more Replication Servers you have (up to one per data source or node), the more available the data will be. Asynchronous stored procedures also contribute to availability.

In addition, there is a switchover capability. If the primary data server or Replication Server goes down, another site can become the primary, regardless of whether the remote data server is an SQL Server or not. This provides a level of fault tolerance and allows updates to

continue to the primary data even if the primary data source fails. Replication Server also has the intelligence to automatically synchronize its data after failure.

PERFORMANCE BENEFITS. Sybase's asynchronous store-and-forward replication architecture has performance benefits over 2PC. It also eliminates the need to first extract the updated records from the primary site database before sending them as in other replication schemes. This means better performance at the primary site. Replication also ensures that reads are performed quickly since all data are local.

SITE AUTONOMY. Another major benefit is site autonomy. Updates are always made to the primary data source first and then replicated to copies. Therefore, each primary data source maintains control over its own data.

Replication Server

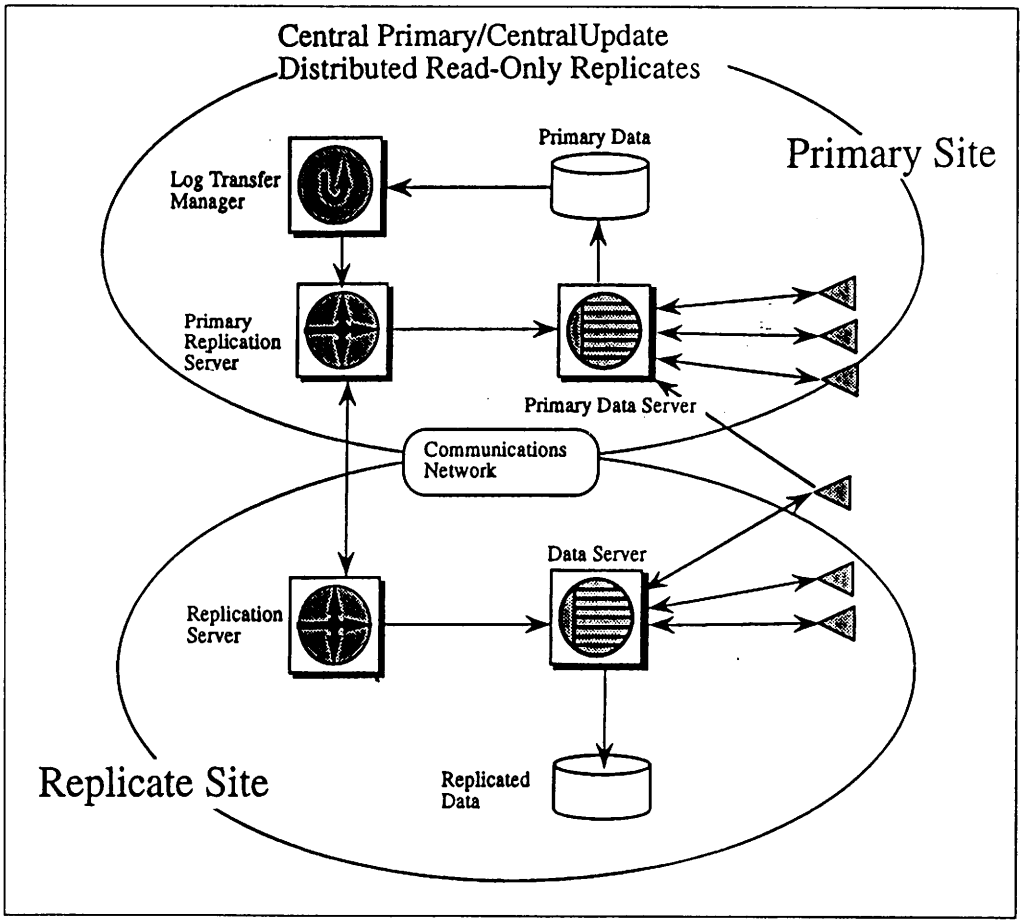

Illustration 3. The Log Transfer Manager passes changes made to the primary data site to the primary Replication Server. The changes are then propagated to remote sites via remote Replication Servers. If any part of the network is down, Replication Server uses store-and-forward capabilities to communicate changes after the failed node comes back online.

Automatic 2PC Is Still Missing for Sybase

Sybase is still missing server-enforced 2PC. This is required in situations where a single transaction must update data in two locations across the network (e.g., inventory and orders are on different systems). Replication cannot solve this problem unless the two tables can be merged at a single location. Sybase does support a two-phase commit protocol, but it is a client-side 2PC that must be programmed into the application. Multisite updates cannot be managed automatically by SQL Server. All of Sybase's major competitors—Oracle, Infor-

mix, Ingres, Progress, and InterBase—have implemented an automatic, server-enforced 2PC protocol as part of their distributed database strategy. Sybase is committed to implementing an automatic 2PC but has not published a time frame. Sybase thinks many more customers are interested in replication than in automatic 2PC. And, judging from the Sybase users we have talked to, Sybase is definitely on target. There is tremendous customer interest in Replication Server.

If the Replication Server approach catches on, in addition to heating up the discussion of 2PC versus replication for distributed data environments, it raises questions about the need for transaction monitors. Sybase's philosophy is that transaction monitors are not required in the client/server environment. However, Sybase is also committed to XA compliance in the future.

Replication Server is in alpha customer sites with general availability targeted for September on SunOS and IBM RS/6000. Other platforms will follow.

Oracle's Replication Is More Limited

Oracle7 is the only competitor that has any replication capabilities. Oracle7 has table replication in the form of "snapshots." (See *Open Information Systems*, Vol. 7, No. 9, for in-depth coverage of Oracle7.) The major differences are the fact that Oracle's replication is not database-event-driven (replication can be scheduled but is not based on a database transaction), does not cover non-Oracle data sources, is not customizable (programmable), and doesn't have the ability to switch if the master table is unavailable. Oracle essentially provides an automated extract capability with its table replication.

There is an important limitation in Oracle's replication. When maintaining a snapshot log in order to send only changes from the master table to its copies, Oracle automatically uses the after-row trigger on the master table to create the snapshot log. This means that a user-defined after-row trigger cannot be used on this table, even though the after-row trigger would be the logical place in Oracle7 to program row-level integrity/business rules. So Oracle makes you choose between a snapshot log and after-row business rules. While we understand why Oracle has had to do this (how else could Oracle7 automatically add or delete the snapshot log procedure if it were contained in user-written trigger code?), it may not be an acceptable choice for some customers.

Informix OnLine Replication Will Address Availability

None of the other major Unix RDBMS vendors currently offers replication services. In OnLine 6.0, Informix will offer the ability to replicate one secondary copy of a database. The secondary copy will be read-only, and, if the primary database fails, the secondary site can take over. The main purpose of Informix's replication is hot-site backup for high availability in case of primary site failure, not replication of data in a distributed sense.

Navigation Server

Sybase's Strategy for Parallel Transactions and Queries

The Navigation Server is designed to provide parallel processing of both transactions and queries in a very large database environment. It is actually a collection of specialized servers—Data, DBA, Schema, Control, and Split Servers—that work with SQL Server. Navigation Server will support SQL Server 4.9.1 in its first release.

Two fundamental features of the Navigation Server are the ability to make multiple SQL Servers look like a single server to the user and the ability to partition data across these servers. In general, the Navigation Server moves the processing to the data rather than moving the data around. Thus, a single transaction or query can run in parallel on multiple servers, optimizing performance through transaction-scaling and the ability to parallelize queries across large databases. Sybase expects to attain OLTP transaction rates in excess of 1,000 tps/A and to decrease response time on queries by 50 to 90+ percent. Sybase is con-

vinced that, with the appropriate database functionality and configuration, the customer doesn't need a TP monitor to achieve an impressive number of transactions per second.

Joint Development with NCR

Navigation Server was jointly developed with NCR's Large Systems Division. NCR's (Dayton, Ohio) primary interest is to offer a high-performance parallel processing system for OLTP and mixed applications (both OLTP and decision support), a platform that has scalability, speed, and the ability to manage both queries and transactions. Large customers are moving off the mainframe, and NCR's strategy is to lure those customers from the IBM fold. The Teradata system, also part of the NCR portfolio, is not a solution here. While it supports parallel query processing quite well, it is strictly a decision-support system and runs on specialized hardware (NCR DBC 1012 or a customized version of the 3600). It does not provide good transaction performance since it cannot parallelize transactions; it serializes everything. We would expect the Navigation Server, designed to handle both queries and transactions, to make the Teradata system obsolete over time.

Architecture

The Navigation Server is a collection of servers allowing database tables to be horizontally partitioned across multiple loosely-coupled processors. (See Illustration 4.) It is a shared-nothing, message-based system, and it supports three data-partitioning methods, all of which are applied at the table level:

- Range partitioning, by which rows are located based on the value of one or more fields in the row. For example, all employees with names beginning with A through E are on one server, F through I are on another, etc.
- Hash partitioning, by which Navigation Server calculates a hash value for each row, and the hash value determines the server on which the data are stored.
- Schema partitioning, by which individual tables are located on different servers in the network.

The details of how the Navigation Server works are too complex for this article, but here is a summary of the basic components and their functions. All of these components use Open Server and Open Client and support Sybase's TDS messaging protocol.

Navigation Server: A Collection of Servers

DATA SERVER. The Data Server is the smallest executable unit of parallelization within the Navigation Server. It usually includes an SQL Server, a Control Server, and a Split Server. The Control Server communicates with the user, receiving requests, controlling execution, and sending data back to the user. Therefore, only one Control Server is active for any one SQL request. A Control Server can reside on a node separate from the SQL Server, and extra Control Servers might be necessary as front-end processors to handle high volumes of concurrent users. The Split Server, primarily used for join requests, has two main functions: to redistribute and to merge split tables. The SQL Server, of course, accesses the local database. The Data Server is the runtime component of the Navigation Server.

DBA SERVER. There is one replicated DBA Server in each Navigation Server configuration. The DBA Server is essentially a Control Server with added functionality for compiling Data Manipulation Language (DML) statements and creating the parallel SQL (PSQL) language necessary to execute a statement across nodes, and a Schema Server. It also executes data definition statements, checks security statements, and synchronizes the Global Directory and SQL Server system tables. A second DBA Server always runs as a hot standby in case the primary one fails.

The Schema Server component of the DBA Server manages the Global Directory, which is the data dictionary used by the Navigation Server to resolve user requests. This directory includes system tables and other tables defined by the Navigation Server to keep track of how data are partitioned. All Schema Server data are mirrored and accessible from two

nodes to guarantee high availability. In addition, selected global schema information is cached in Control and Split Server as appropriate. A Recovery Manager keeps cached data consistent across all servers.

Navigation Server Architecture

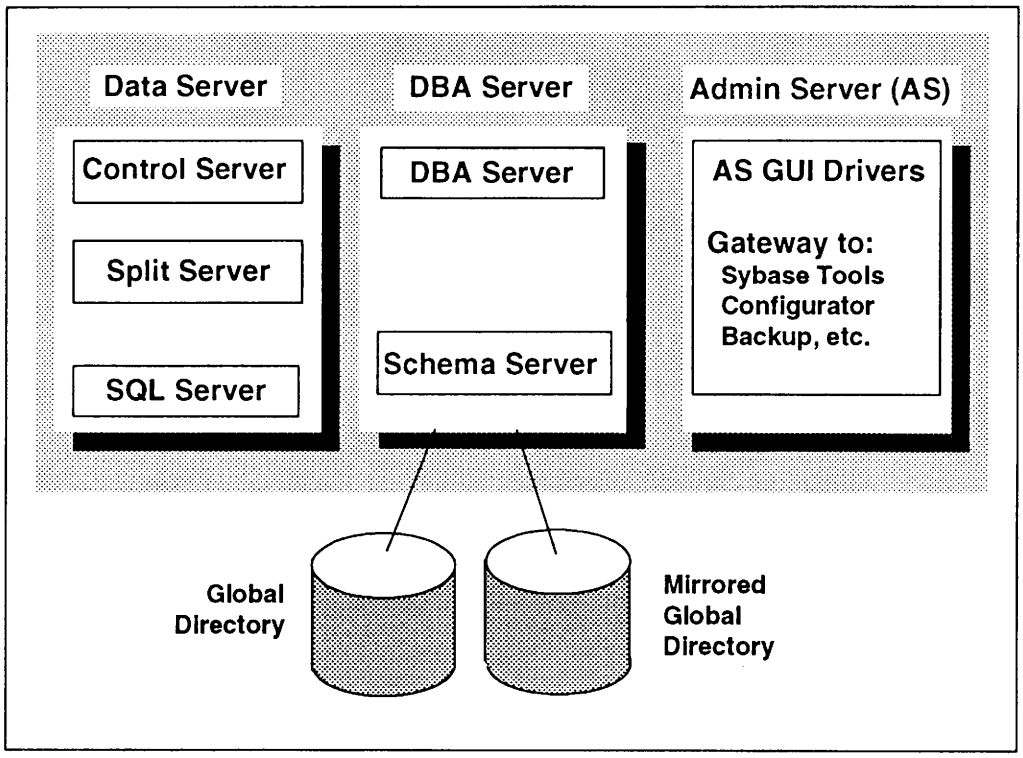

Illustration 4. Navigation Server is a collection of servers that work together to process an SQL statement in parallel across partitioned data in a loosely-coupled server environment.

At compile time, the Control Server sends the SQL statement to the DBA Server. The Schema Server accesses the Global Directory for information about how the data involved are partitioned. All of this is then given to the PSQL Compiler, which resolves the statement, optimizes it, and generates PSQL statements. The last steps are to create the appropriate SQL statements, in the form of stored procedures, for each Data Server and control modules for the Control Server. The DBA Server then passes the control modules back to the Control Server and the stored procedures to each participating Data Server. At runtime, the Control Server communicates directly with the SQL Servers to execute, in parallel, any previously compiled and optimized SQL (i.e., stored procedures). (See Illustration 5.) Ad hoc queries are parallelized at runtime; Sybase maintains that this will not significantly impact performance for the user.

In the case of inserts, updates, and deletes that modify data in more than one partition, the Control Server coordinates the distributed transaction using Sybase's client-side 2PC protocol to ensure that the transaction is committed or rolled back at all nodes.

ADMINISTRATION. The Administrative Server handles a variety of tasks necessary to administer the complex Navigation Server environment. It provides gateways to Sybase tools, the Backup Server (for backup and restore), and the Configurator (for new and modified configurations). It has a graphical user interface for access to management functions and performance monitoring data. It also handles log management and recovery. (Navigation

Server uses a hierarchical monitoring scheme to handle software failures; for example, the Control Server within each Data Server monitors and restarts the servers in its unit.)

SUMMARY. Navigation Server will be a hardware- and network-independent way to approach the performance of a Teradata machine, not on a specialized database machine, but on standard components. The concept is to allow the customer to start with one node and expand the configuration as the database grows—a low entry point with scalability. An important point here is that Navigation Server is not an *add-on* to an SQL Server environment; it *is* the SQL Server environment. Since data are partitioned, all requests must first go through the Navigation Server front end; you cannot communicate directly with the SQL Servers that are part of the installation. The Navigation Server takes care of installing the SQL Servers; you cannot use existing SQL Servers as part of the Navigation Server.

Navigation Server Runtime Environment

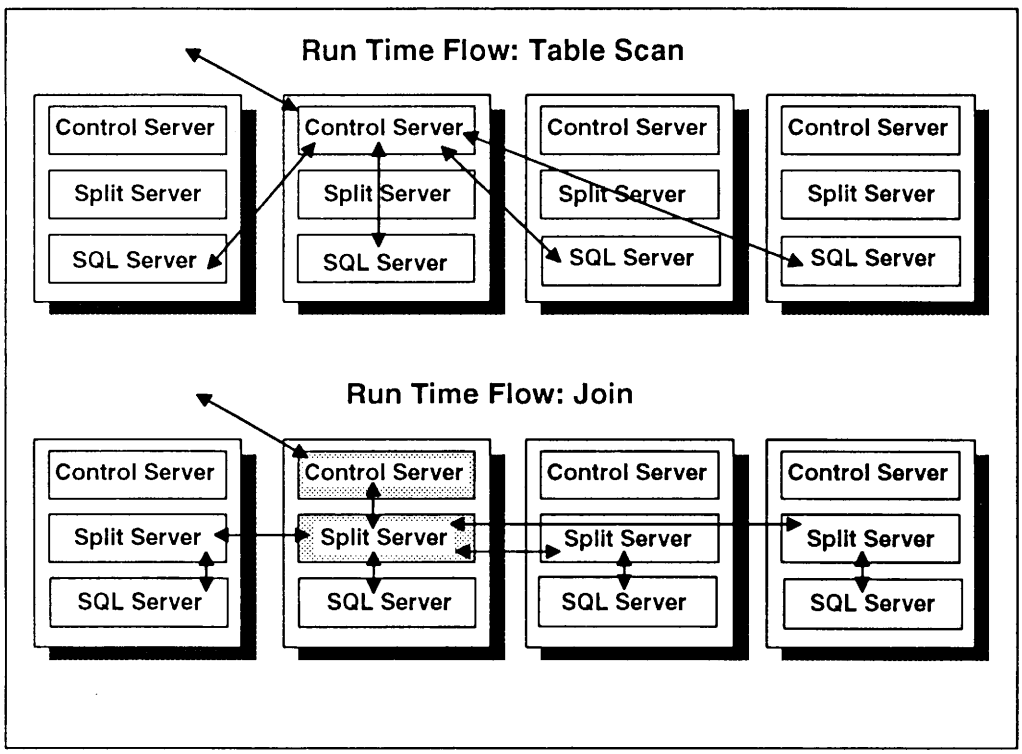

Illustration 5. The top diagram illustrates the runtime flow among Navigation Server components for table scans, inserts, updates, deletes, and in-place joins. The bottom diagram shows the runtime flow if the request involves a join where the columns in the join do not match the columns on which the tables are partitioned—for example, customer records are on one server, orders and line items on another. Here, the Control Server involves the Split Servers for redistribution of data as temporary copies of partitions so an "in-place" join can be executed on one or more servers.

NCR Is the Initial Platform

Sybase states that four partitions are required before any speedup of table scans, joins, or sorts can be expected. And, if the Navigation Server must move or replicate partitions, the network overhead will reduce the performance gain from parallel processing.

Initially, Navigation Server will run on the NCR 3600/3700 (massively parallel systems) and will be certified on the 3450 and 3550 (both are SMP machines). Navigation Server will be generally available on NCR by the end of 1993. Sybase plans to move Navigation Server to other platforms in the future and is in the process of determining specific platforms and timing. The nodes in the Navigation Server are connected by memory, fast interconnect, or

fast LAN, and each node has its own data storage subsystem. The first release will require homogeneous nodes that share the same type of interconnect.

The Configurator Determines Partitioning and Monitors Performance

The partitioned data appear as a single logical database to the application developer and to the user. The Configurator, one of the System 10 Control Servers, is a tool for upfront planning and ongoing management of a Navigation Server configuration. It recommends how data should be partitioned based on database design, planned transactions, and performance and capacity requirements specified by the DBA or systems programmer responsible for installing Navigation Server. Since location of data is key to performance, Sybase thinks its Configurator will be a differentiator in the industry. It combines knowledge about entity-relationship and flow modeling, capacity planning, and workload simulation to analyze customer requirements and draw diagrams of the type of machine and data layout needed to optimize performance. After the Navigation Server is implemented, Configurator can then monitor actual performance and recommend changes if appropriate. There are also facilities to manage the placement, mirroring, and moving of data among systems for those who need to know how data are physically located.

Initially, the Configurator only works with the Navigation Server, providing configuration support at the high end. Sybase plans to migrate the Configurator to work with other System 10 components—SQL Server, Replication Server, the OmniSQL Gateway, etc.—as a general capacity planner, and to work at the low end as well.

Limitations

There are three primary limitations to Navigation Server in its initial release. First, it runs only on NCR, although Sybase plans to generalize it for other platforms as well. Second, Sybase has not built parallellization into the SQL Server itself. To partition data and parallelize a query across disks *within* one SMP machine to take advantage of multiprocessors on a single platform, Navigation Server has to install multiple SQL Servers on the SMP machine. So Navigation Server is designed for a loosely-coupled/massively parallel environment. Navigation Server does, however, provide management tools that make the entire parallel installation look like a single SQL Server. Third, Release 1 of Navigation Server is based on SQL Server 4.9.1, so it will not support the new features in SQL Server 10. The second release of Navigation Server will use the then-current version of SQL Server. We should also note that, in Navigation Server, triggers can only affect data controlled locally by each SQL Server. Navigation Server will not allow triggers to use RPCs to access remote data (since placement of data is controlled by Navigation Server).

Oracle and Informix Are Addressing Different Parallel Architectures

Oracle's Parallel Server option is a different animal from Sybase's Navigation Server. It allows multiple Oracle server installations to share a common database in a loosely-coupled or massively parallel environment using distributed lock and cache managers. It does not do any parallellizing of data or SQL statements across nodes or processors. The first pieces of Informix's parallel architecture, coming in OnLine 6.0 later this year, are designed to support an SMP environment. The ability to partition data and parallelize SQL statements will be built into the multithreaded server itself, not in a layer of software around the server as done by Sybase. OnLine 6.0 will parallelize operations like building indexes and sorting records. Later releases will introduce parallel queries and joins, and the ability to parallelize data and processing across nodes in a network with the requisite administration capabilities.

OmniSQL Gateway

A Turnkey Gateway to Heterogeneous Data

OmniSQL Gateway combines Sybase's existing gateway products into one environment with some impressive enhancements. It provides transparent read/write access to all supported heterogeneous data sources as if they were SQL Servers. This means applications can use a single database access language—Transact-SQL—to access any available data source. Previously, with the Sybase gateways for Oracle, Informix, Ingres, Rdb, and DB2, one used native SQL. All the gateway provided was translation of the data coming back. The new

OmniSQL Gateway

OmniSQL Gateway will also offer stored procedure transparency (you can use Sybase stored procedures with non-Sybase data since OmniSQL Gateway includes the Transact-SQL parser and compiler) and, therefore, a single debugging environment for multiple databases.

Heterogeneous Distributed Joins Are Supported

The gateway also provides location transparency and heterogeneous distributed join capability. Joins are executed on the OmniSQL platform after data from each data source have been qualified (via subqueries generated by the Gateway's distributed optimizer), brought to the Gateway platform, and translated into SQL Server format. Tables and rows are never moved from one node to another for a remote join, even if this would be more efficient in terms of network traffic than bringing qualified rows to the Gateway for the join. We would like to see this option available for joins involving all-Sybase data. (The Informix and Oracle7 distributed query optimizers, for example, can do this.) The Gateway keeps statistics on the number of rows in each table for optimization purposes but does not maintain any distribution of data statistics. Sybase is currently conducting performance tests on OmniSQL Gateway.

Targeted for Decision Support

Sybase is positioning OmniSQL Gateway primarily as a decision-support tool for multidatabase queries and joins. It can also be used for "lightweight" transaction processing (low-volume, single-site transactions), but it has no two-phase commit protocol for distributed transactions. Limited trigger support is available in that triggers on Sybase tables can cause the gateway to access non-Sybase data sources; triggers cannot, however, be defined on non-Sybase databases through the Gateway.

The OmniSQL Gateway is built on Open Server components and uses the appropriate access routines for heterogeneous data. The gateway to Oracle, for example, translates Open Client requests into Oracle's Oracle Call Interface (OCI). Any Open Client can connect to the gateway. The DBA runs a utility to synchronize the data dictionaries between OmniSQL Gateway and the underlying DBMSs. Sybase is considering an automated facility for this.

Access to Oracle, DB2, RMS, and C-ISAM Initially

In the first release, due out in May, OmniSQL Gateway supports Oracle, DB2, RMS, and C-ISAM. The base OmniSQL Gateway (called the OmniSQL Server) includes access to SQL Server and either RMS (on VMS) or C-ISAM (on Unix). The Oracle and DB2 access modules are optional. The OmniSQL Server, Oracle and DB2 access modules are each 40 percent of the SQL Server license fee. OmniSQL Gateway will first be available on VAX/VMS, IBM RS/6000, HP 9000, and SunOS.

In the next release, Sybase plans to add support for Informix and Ingres, enhance its administration tools, add NCR as a supported platform, and add generic gateway support for external gateways (an API to incorporate additional data sources other than those provided by Sybase).

SA Companion

Managing a Distributed Client/Server Environment

A major goal for Sybase is providing comprehensive tools—the System 10 Control Servers—for managing a distributed client/server network environment. As we would expect, the company is not limiting its focus to the database environment but is taking a broader view of this whole issue. The recently announced agreement with Tivoli to jointly build DME-compliant management tools that cover both the operating system and the DBMS is a strategic part of Sybase's direction. With the initial release of System 10, we will begin to see Sybase's plan take shape as it expands the scope of existing system administration products and introduces new ones. While these tools will not be integrated in the first go-round, the intent is to pull them all together under a single interface—the Tivoli framework—in the future. (Backup Server and Configurator are also considered Control Servers; these are covered in the SQL Server and Navigation Server sections, respectively.)

SA Companion

SA Companion

A new release of the SA Companion product for SQL Server system administration is coming by mid-year. The primary enhancement is support for System 10 components, the first of which will be Backup Server and the OmniSQL Gateway in addition to SQL Server. Managing OmniSQL Gateway will be an initial step toward management of heterogeneous databases.

SA COMPANION FOR SQL SERVER SYSTEM ADMINISTRATION. SA Companion is an APT-Workbench 5.0 client application developed by SQL Solutions before it became part of Sybase. It gives the DBA menu access to SQL Server system administration tasks. An important feature of SA Companion is its ability to centrally manage multiple servers in a distributed network from a single client workstation. The DBA can view a list of servers in the network and easily connect to whichever one is appropriate. (See Illustration 6.) With SA Companion, the DBA can view and manage servers, devices, databases, and users, and it has access to an SQL editor and reports. Every system configuration parameter is available through SA Companion.

A CONSISTENT METHODOLOGY. Most of what can be done in SA Companion is also available from the command line and/or from system-stored procedures that Sybase distributes with SQL Server. However, in addition to a friendlier interface, SA Companion also offers a single environment and a consistent methodology for doing all of these tasks. SA Companion provides a layer of transparency for the DBA to mask differences among heterogeneous hardware and operating systems. The administrator uses a single interface and the same language to manage the system regardless of the underlying platform. Installing a device, such as a new tape drive, is done the same way in SA Companion for any hardware and operating platform. The DBA doesn't have to remember the details of how to do this on a particular platform or what arguments a stored procedure requires. This not only masks differences between underlying platforms but also ensures that anyone performing DBA tasks will do them right and in the same way. This is particularly helpful for activities that are complex or done infrequently, like deleting or adding a disk, and for environments where developers also act as DBAs. As Sybase expands the scope of SA Companion, this underlying design philosophy will become even more important.

"LIGHTS OUT" MANAGEMENT IS THE GOAL. In Release 10, SA Companion manages the OmniSQL Gateway and the Backup Server in addition to SQL Server, with access to the new features of all three. For example, the DBA will have "lights out" backup capability (the ability to schedule backup tasks for execution without operator intervention), with other unattended capabilities coming in the future (such as recording sequential snapshots of the SQL Server with the ability to record only the differences between snapshots).

SA COMPANION WILL MANAGE ALL SYSTEM 10 COMPONENTS. Over time, SA Companion will centralize management of other System 10 servers—Replication, Navigation, and Configurator—plus any other open server application in the network. It will eventually be subsumed by the Sybase/Tivoli DME-based effort as a part of an overall client/server management toolset. SA Companion currently has a character-based interface (it is built in APT-Workbench) although it does support drop-down menus and a mouse for point-and-click; a full graphical interface will come with the move to the Tivoli environment.

SQL Monitor: A New Performance Monitoring Tool

Currently, Sybase doesn't offer any performance monitoring tools that provide access to what's happening inside SQL Server or information about resource usage. A new product for this function, SQL Monitor, will debut with the first wave of System 10 components in mid-1993.

SQL Monitor: A New Performance Monitoring Tool

A Client/Server Architecture

As we would expect, SQL Monitor is a client/server application built on Open Client/Open Server. On the server side, an Open-Server-based Monitor Server will track a wide variety of operational statistics on a real-time basis in shared memory in the SQL Server. On the client side, SQL Monitor will then query the Monitor Server on a configurable interval and display the desired statistics on the screen. Thus, SQL Monitor is "unobtrusive"; it does not query SQL Server directly and does not impact server performance. (See Illustration 7.) This is an important difference between SQL Monitor and other products, like Oracle's SQL*DBA, which do query the server. SQL Monitor statistics will include database-locking status by process and server-wide data and procedure cache utilization, I/O volume and average completion time by device, memory allocation, network traffic, CPU utilization, and transaction rates, among others.

SA Companion

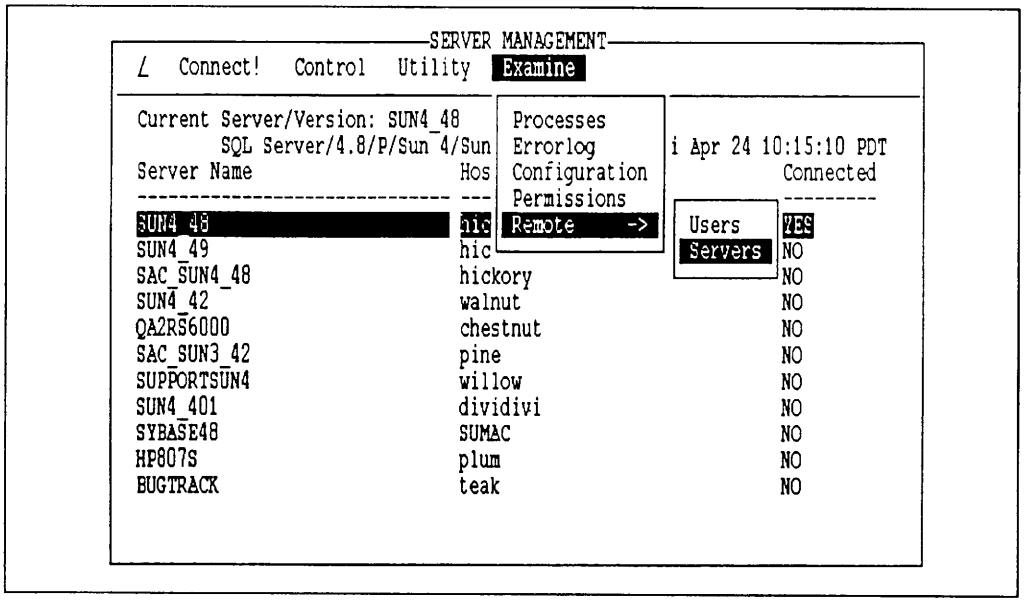

Illustration 6. SA Companion allows the DBA to view all servers in the network and remotely manage them.

With a Graphical Interface

SQL Monitor has a GUI interface (initially Motif), presenting database server statistics graphically in multiple windows. You can select the statistics you want and the level of detail, the type of graph for each, whether the legend is displayed or not, colors, and many other options. SQL Monitor can also highlight user-defined exceptions based on thresholds (e.g., CPU utilization rises above a certain percentage). The first release of SQL Monitor will not include any reports, but screen displays can be paused and a print command used to capture the statistics. Reports and support for Windows will be in a future release.

SQL Debug Handles a Single Client Connection Today

SQL Debug is a source-level debugger for Transact-SQL code. Introduced in early 1992, it is currently packaged with SQL Advantage, an SQL editor, as the Sybase Testing Toolset. (Sybase plans to combine the two products in the future.) SQL Debug lets the developer interactively monitor and debug any Transact-SQL code—code generated by an application and/or stored procedure code—as it runs against SQL Server. (See Illustration 8.) The product runs on SunOS, HP 9000, IBM RS/6000, and VAX/VMS and has a Motif interface. Sun Solaris and NCR 3000 are coming by the end of 1993.

Moving to a Multiclient Code Debugger...

The Release 10 version, due by mid-year, will roll out two significant enhancements. First, it will enable multi-client debugging of code. Currently, SQL Debug runs as a single client application connected to the server. In the next release, SQL Debug will have a console from which multiple Open Client connections can be viewed and debugged simultaneously

(see Illustration 9); these connections can be from multiple clients or multiple connections from one client, or a combination of both. SQL Debug can intercept *all* language events against the SQL Server and control processing done by multiple client applications. Thus, the developer can debug problems caused by the *interaction* among clients as well as errors within individual programs. The developer can view up to five monitored connections at any one time on the console screen (although an unlimited number can be monitored and controlled), and can focus the Inspector (the current SQL Debug environment) on any individual connection for more detailed debugging.

...That Can Debug Any Open Client Application

Second, SQL Debug will debug what Sybase calls *naive clients*. These are clients that don't know they are being debugged (intercepted) as they send Transact-SQL to the server. Many third-party tools on the market generate SQL and/or a stored procedure on the client application's behalf, but the application developer either doesn't know this or can't get access to it within the tool. In this case, SQL Debug not only can identify what the tool is generating for the client, but it can aslo allow the developer to see how multiple clients interact. While the developer cannot change the language a naive application generates from within SQL Debug (since the developer, in this case, doesn't have direct access to the source code), the developer will be able to modify the application with the original tool to solve the problem. The developer could, however, modify any stored procedures in the server.

SQL Monitor Architecture

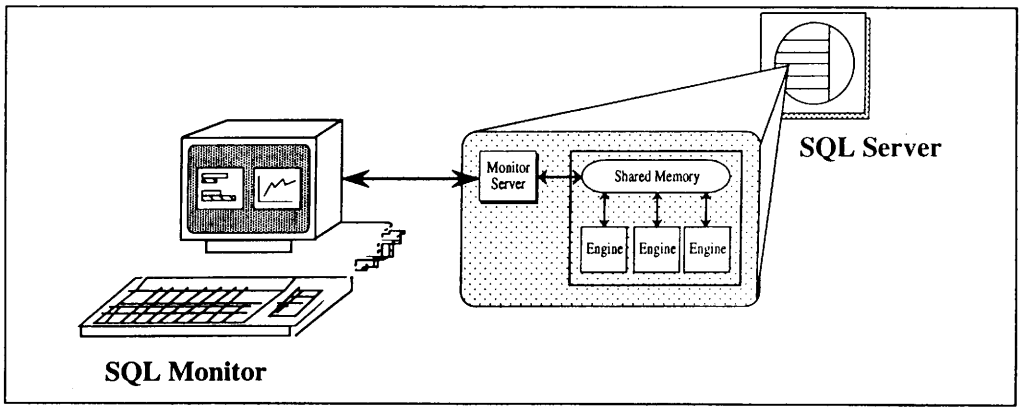

Illustration 7. SQL Monitor will provide real-time access to information about SQL Server activities and workload using a graphical (Motif) interface. SQL Monitor accesses shared memory for performance information; it does not query the server and, therefore, doesn't impact performance itself in a production environment.

Another Step in Managing a Heterogeneous Client/Server World

Both of these features are increasingly important as third-party application development tools become more popular and heterogeneous clients may be accessing the same database. Sybase will offer a single debugging environment for client/server interaction, regardless of what tool was used to write the application. We have not seen any other products that can do this for either Sybase or any other DBMS product.

Extending Sybase's Management Scope

Tivoli Agreement

Sybase recently announced an agreement with Tivoli Systems (Austin, Texas), the vendor whose object-oriented system administration technology has been adopted by both the Open Software Foundation (OSF) and Unix International (UI) as base technology for their respective distributed computing management environments—OSF's Distributed Management Environment (DME) and UI's Atlas-DM. Sybase is licensing Tivoli's Advanced Development Environment (Tivoli/ADE) and is already working with Tivoli to

integrate Tivoli's operating system-level administration capabilities with Sybase's distributed database administration efforts.

Tivoli brings a more general level of system and network management expertise—operating system control, network transport for RPCs, the ability to monitor traffic across the network—that complements Sybase's ability to manage the RDBMS environment. Tivoli will provide tools to monitor traffic across the network, and Sybase will provide tools that can look within the database components. The goal is to provide an integrated, object-oriented environment within which to manage an entire distributed network. Sybase plans to incorporate all of its evolving management tools into the Tivoli effort—SA Companion, SQL Monitor, Configurator, etc.

SQL Debug Inspector

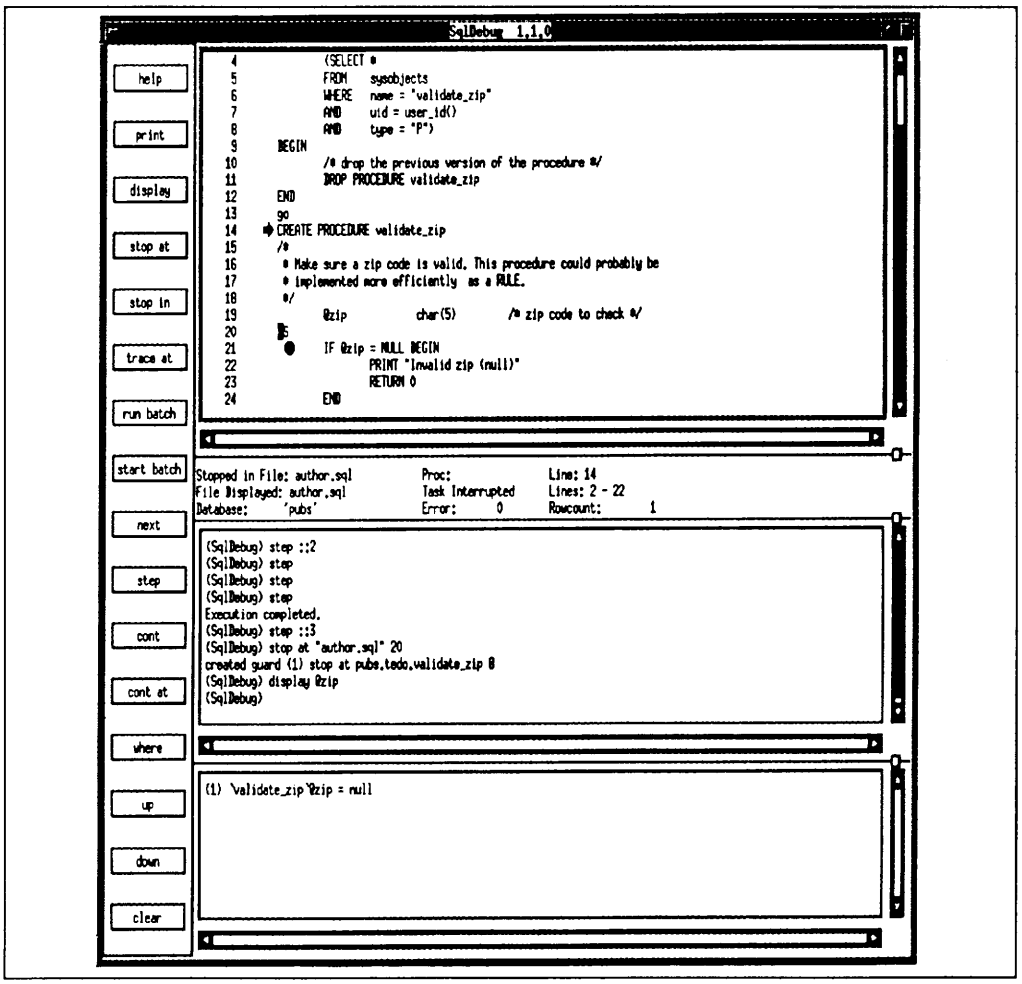

Illustration 8. Currently, SQL Debug has three windows for interactively debugging code. The top window displays the Transact-SQL code as it runs, graphically identifying with icons the current step, procedure calls, and debugging points (stop guards, trace points, etc.). In the middle window, the developer enters SQL Debug or Unix commands. The bottom window displays the current value of any variables the developer wants to see. On the left is the customizable button panel for frequently-used commands. The windows are all configurable as well. This screen for debugging a single client connection will be incorporated in the next version of SQL Debug as the Inspector component.

Debugging Multiple Clients

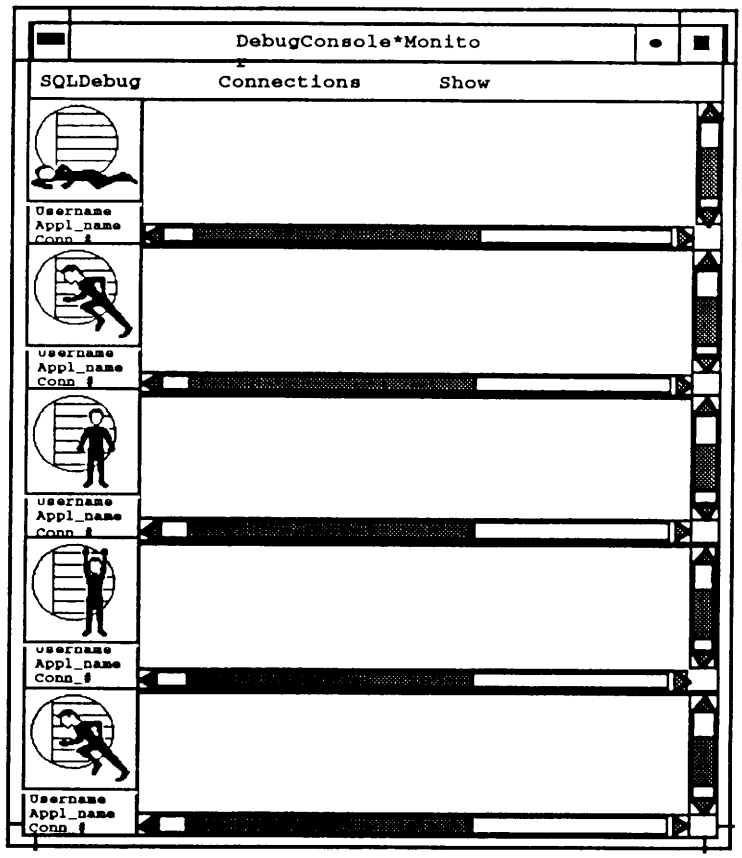

Illustration 9. Release 10 of SQL Debug will enable the developer to debug multiple client applications running simultaneously against SQL Server. The developer will have commands to operate on all applications collectively (e.g., stop all when you get to this point in this application or stop when this variable is a certain value), and the ability to invoke the Inspector (see Illustration 8) for any one of the applications. Icons indicate whether each application is sleeping, running, or stopped.

Tivoli currently offers its Tivoli Management Environment (TME) software for managing Unix systems. TME's graphical, object-oriented, drag-and-drop interface makes administration of Unix users, groups, devices, remote systems, security, network-wide distribution of software, etc. much easier and more intuitive than working at the Unix command line. While Tivoli understands Unix systems, Sybase understands how to manage database users and servers. The two companies are working to integrate these two environments for the user. Sybase will also take advantage of specific Tivoli features, such as the software distribution module, the Sentry component for administrative event monitoring, and the object-oriented development infrastructure.

We think Sybase is out in front of the rest of the industry in this area. Managing database objects with Tivoli's TME is closer to reality than most people think, and while it doesn't give Sybase a proprietary advantage, it does mean that the company is pretty far along in its DME-compliant implementation effort. This fits well with and lends credibility to Sybase's claim that it will, indeed, be able to manage an enterprise-wide client/server network.

Extending Sybase's Management Scope

Future Directions for Management

Control and administration of the overall system is a major area of future development for Sybase. The company recognizes the need to integrate all its management tools into a single control center (a *global viewer*) to monitor the entire Sybase environment, to conform to industry standards (e.g., DME), and to provide interoperability by enabling its management tools to work with non-Sybase components. These capabilities will be built on the existing System 10 components over time. The agreement with Tivoli is a large part of this overall control and administration effort.

The Sybase Mainframe Connection

The Mainframe Will Continue to Be an Important Component

The Sybase mainframe products are an important aspect of assessing the company's ability to effectively manage enterprise-wide data. Customers who are moving off the mainframe are not going to do so in one fell swoop; they need the ability to migrate mainframe applications to other systems in an evolutionary way. And, while many organizations will continue to maintain data on the mainframe, they want the mainframe to participate effectively in the overall network. Sybase has introduced several IBM MVS mainframe products to incorporate the glasshouse into the Sybase client/server environment. To date, the competition has addressed this issue only with proprietary gateways to specific data sources—DB2, SQL/DS, IMS, VSAM, etc. Sybase offers the Open Server and Open Client for CICS, the Open Gateway for DB2, and the Net-Gateway as its approach.

Open Server for CICS: Access to Mainframe Data

Open Server for CICS is a set of system services for CICS transactions. Through CICS transactions, client applications can access any data source on MVS, including VSAM, DL/1, sequential files, static and dynamic access to DB2, and non-IBM data managers. This gives non-mainframe users access to data on the mainframe.

Open Client for CICS: Access to Data on the Client/Server Network

The Open Client for CICS is Sybase's latest mainframe product. This allows a CICS user on a 3270 terminal to access any Open Server application on the network. As an example, suppose some mainframe data have been moved to Unix and are accessible on a LAN. The CICS user could access these data as if they were still on the mainframe. This is a very powerful concept that recognizes that there are not only data on the mainframe but users as well. Both data and the users can be migrated to a client/server environment using an organized, step-by-step approach.

Sybase Provides Both Dynamic and Static SQL Access to DB2

Sybase's turnkey gateway for DB2 provides read/write access to DB2 via dynamic SQL with automatic error-mapping and data-type conversion. The client uses native DB2 SQL to communicate with DB2. The DB2 gateway also includes all of the functionality of Open Server for CICS (on which it is built) and, therefore, access to DB2 via static SQL as well. This is very important in production environments because static SQL is preparsed, preoptimized, and precompiled. Static SQL offers significant performance enhancements over dynamic SQL, which is the only option provided by competing DB2 gateways. Unlike using dynamic SQL through the DB2 gateway, using static SQL through the Open Server for CICS does require development by the customer. (Sybase offers more transparent but more limited access to DB2 data in its OmniSQL Gateway.)

Net-Gateway Provides Network Protocol Conversion and Security

Net-Gateway provides the connection and protocol conversion between the LAN (client/server network) and the IBM SNA/LU6.2 network. To the client application, Net-Gateway looks like an SQL Server. It maps database RPCs issued by the client (which can be another SQL Server or Open Server as well as any Open Client) to CICS transactions, or to the appropriate server in the LAN if the client is CICS. The routing of the request is transparent to the client. When accessing the mainframe, the client can attach to multiple mainframes, to multiple CICS regions within a single mainframe, and to multiple transactions within a CICS region. Net-Gateway enforces mainframe security and can interface to mainframe security packages, such as RACF. It currently runs on the RS/6000, SunOS, and OS/2.

Tools Overview

System 10 Includes the Tools Strategy

Application development tools have traditionally been a weak area for Sybase. This is one reason the company has been aggressive in developing partnerships with third-party tools vendors. In fact, Sybase sells Uniface and Unify directly in addition to its own SQL Toolset (Data Workbench and APT-Workbench). Sybase has also bolstered its tools offerings through acquisition. SQL Solutions brought in life-cycle, report-writing, and administration tools, Deft brought CASE, and Gain Technology (Palo Alto, California) brought GUI builder and multimedia capabilities. We have already described some of Sybase's tools and plans on the system management side. Sybase wants very much to be as successful in the tools business as it is in the server business. Here is a brief overview of what we expect to see from Sybase in development and end-user tools over the next several months.

Three Sets of Tools Focus on Different Users and Application Styles

Sybase will address development tools in three general categories. The first is enhancement of the APT-Workbench toolset. This will continue to provide a forms-based development environment with both character and GUI support through GUI style-mapping. The key here is deployment across a broad range of user interface styles, including character based. The second category of tools, currently under development, will be strictly GUI based but will be targeted as a "small footprint" alternative to full-blown multimedia resource requirements. It will add graphics to forms in presenting data to the user. Both these categories are aimed at the business user.

The third category is the Gain/Momentum tool for high-end workstations (a single development license costs $20,000) and a multimedia, object-oriented development environment. The resulting applications would be targeted at what Sybase calls "volunteer users," users who do not have to use the computer to get the job done, but will if it is easy and intuitive enough without training or documentation. An example here is someone renting a car and faced with the choice of waiting in line or using a graphical application on a touch screen.

A Future Repository Will Connect the Toolsets

Sybase will also introduce a repository (the Meta Server) for storing both data and application objects. This will be the unifying force among the different toolsets and applications, and it will also be available to third-party tools.

Can Sybase Catch Up?

It is early to evaluate Sybase's tools strategy and plans since the company has not divulged much in the way of details. The key for Sybase in tools is implementing a product line that covers all the bases while providing a migration path for users. We have concerns about the company's ability to both catch up on the tools front and, at the same time, effectively integrate all of its acquisitions on a timely basis.

Availability

System 10 Is Not That Far Off

One of the most interesting aspects of System 10 is the fact that it will begin to roll out sooner than most expect. Sybase was careful at its November introduction to indicate that this was a "high-level" announcement of direction and that specific products would not be available until sometime in 1993—the implication was late 1993 at best. However, we will see the first set of System 10 components generally available by the end of June, and the rest, with the exception of the Tivoli system management environment, by the end of the year.

All of these products will also run with SQL Server 4.9, the current release, and do not require an upgrade to SQL Server Release 10. In general, Sybase plans to stick with user-based pricing for its System 10 products. The major issue will be what platforms each component will run on initially and when other platforms will be available. Navigation Server, as we mentioned, will only be available on the NCR 3600 platform at first. Sybase needs to

Availability

roll out additional platforms quickly to maintain a competitive advantage, especially for the Replication Server.

What about Microsoft SQL Server?

Microsoft Corporation (Redmond, Washington) is porting SQL Server to Windows NT and will be solely responsible for selling it on this platform and on OS/2 1.x. Sybase is porting SQL Server to OS/2 2.0 (the 32-bit version) and will sell this, since Microsoft has declined any commitment to supporting OS/2 other than its existing OS/2 1.0 SQL Server. We expect that both platforms will upgrade to Release 10 and that, over time, other components of System 10 will be ported to these workgroup-oriented servers. Prime candidates would include Replication Server, the system administration tools, and the OmniSQL Gateway.

Conclusion

The Strategy: To Own the Client/Server Infrastructure

Sybase's strategy is extraordinarily clear. This is a client/server company intent on claiming ownership of the client/server infrastructure, or framework, in a distributed computing environment. While Sybase also provides front-end tools and back-end servers, the customer is not limited to using Sybase products. Sybase has already established its architectural underpinnings in Open Client and Open Server. Now the challenge is to complete the picture with the necessary distributed management, additional distributed functionality, and platform coverage. If Sybase does this, we think the company can, in fact, be a successful manager of an enterprise-wide data environment.

With System 10, Sybase has all the necessary client platforms and a majority of the server platforms; a good RDBMS engine; IBM mainframe integration; a strategy and initial products for parallel processing of data in a massively parallel/SMP environment, sophisticated data replication, and data access across heterogeneous data sources; and a well-thought-out approach to management tools for distributed client/server applications. (See Illustration 10.)

From an enterprise perspective, what's missing is server-enforced automatic 2PC and support for transaction monitors through XA (Sybase is committed here but with no firm time frame, and there *are* applications that require these capabilities), a connection to data in the vast IBM AS/400 environment (Sybase will probably do this through DRDA), full distributed management capabilities (these are under development leading to full DME compliance), and state-of-the-art tools for applications development and end-user data access (although many are available from third parties, and better solutions are coming from Sybase). We also think Sybase needs to expand the platforms for Replication Server as fast as possible. This will be one of the most popular components of System 10; many customers don't want or don't need to incur the overhead and problems inherent in using 2PC for distributed transactions. (We should note that Replication Server can manage data sources on other platforms; it does not have to run on the same platform as either the primary or replicate data.) In addition, Sybase should consider implementing parallelization capabilities *within* the SQL Server. This will offer the SMP customer who wants to parallellize data on a single machine a simpler alternative to Navigation Server.

Sybase has built its success on addressing real business problems ahead of the competition. Oracle is the closest to Sybase in overall strategy, but we see Sybase ahead of Oracle in several areas: a proven, available, and more open client/server architecture; life-cycle and distributed management tools; partitioned databases and the ability to parallelize transactions. Informix is closest to Sybase in its future server architecture and product plans (and has a much stronger story to tell on the tools side), but it doesn't have as broad a perspective as Sybase; it is more of a database/tools company than an enterprise-wide client/server company. Sybase is no longer the new kid on the block, but it continues to be an innovative leader in both the client/server and RDBMS markets. It keeps the competition honest while helping users focus on the real issues.

Sybase, Incorporated
6475 Christie Avenue
Emeryville, CA 94608
(510) 596-3500

Conclusion

Sybase Features Chart

DBMS PRODUCT	Sybase SQL Server Version 4.9/Release 10
COMPUTING ENVIRONMENT	
SERVER PLATFORMS	
Unix	Yes
Proprietary	Digital VAX/VMS
OS/2	Yes (Version 4.2)
DOS	No
PC LANs	Yes; Novell NetWare NLM
CLIENT PLATFORMS	
Unix	Yes
Proprietary	Digital VAX/VMS
OS/2	Yes
DOS	Yes
DOS Windows	Yes
Macintosh	Yes
NETWORK SUPPORT	DECnet, TCP/IP, LAN Manager (Named Pipes), PC LANs; integral part of the product
DATABASE ENGINE FEATURES AND FUNCTIONS	
ARCHITECTURE	
Client/server	Yes
Multiserver	Yes
Configurable number of servers	Yes
Maximum number of servers/system	Limited only by operating system
Maximum number of users/server	Limited only by operating system
Maximum number of users/system	Limited only by operating system
Multithreaded server	Yes
Support for transaction monitors	No; coming in the future
Support for multiprocessors	
Symmetric	Yes
Loosely coupled (e.g., clusters, massively parallel)	No; coming with Navigation Server on NCR by end of 1993
Underlying file structure	Native file system or raw partitions
Support for raw input/output	Yes; optional
Database can span multiple physical devices (disks)	Yes
Support for partitioned tables	
Horizontal partitioning	No; coming with Navigation Server on NCR by end of 1993
Vertical partitioning	No

Sybase Features Chart (continued)

DATABASE PARAMETERS	
Database size	Limited only by disk space
Databases/server	32,767
Tables/database	2 billion
Rows/database	Limited only by disk space
Row size	2K (not including text and image datatypes)
Columns/row	250
Indexes/database	251/table; 1 clustered, 250 non-clustered
Databases connected to a client	32K
Maximum number of tables referenced in a single query	16
Maximum number of databases referenced in a single query	8
Applications/database	No limit
DATA TYPES	
Standard data types:	
Character--fixed length	Yes; 255 character maximum
Character--variable length	Yes; varchar; 255 character maximum
Long character/text	Yes; text data type; 2 billion character maximum
Integer	Yes; int, smallint, tinyint
Decimal	No; coming in Release 10
Float	Yes; 4-byte real and 8-byte
Logical	Bit data type, which may have values 0 or 1
Date/time	Yes; 4-byte and 8-byte
Binary	Yes; binary for fixed length and varbinary for variable length; 255 character maximum
Serial	No; coming in Release 10
Other	4-byte short money and 8-byte money
Ability to extend base data types	Yes (data type and length)
Support for nulls	Yes
Support for complex data:	
Blob/image	Yes; image data type; 2GB maximum
Manipulation facilities	Pattern matching on a text data type; can read/write portions of text or image data type (specify offset and length)
Arrays	No
Number of dimensions	n/a
Manipulation facilities	n/a
Ability to define new data types	No
Ability to define functions/operators for new data types	No

STORED PROCEDURES	Yes
Precompiled	Yes
Preoptimized	Yes
Can call another procedure	Yes
Recursive	Yes
Support for cursors	No; coming in Release 10
Parameter passing	
Data types supported	All data types
Maximum number of parameters	255
Support for complex parameters (e.g., arrays)	No
Ability to return status	Yes
Ability to pass parameters by reference in addition to values	Yes
Ability to return multiple results sets in a single invocation of a stored procedure	Yes
Under transaction management	Yes, provided actions of the stored procedure are within the same server or programmatic 2PC is used
TRIGGERS	
Scope	No limit; triggers can call stored procedures
Database operations supported	Insert, update, delete
Precompiled	Yes
Preoptimized	Yes
Number per table	3 triggers/table (one for insert, update, and delete); trigger can call an unlimited number of stored procedures
Can specify execution order	n/a
Cascading	Yes, up to 16 levels
Recursive	No; yes in Release 10 (optional)
Can be deactivated	No
Under transaction management	Yes, provided actions of the trigger are within the same server
EVENT ALERTERS	Available with Open Client 4.6 and Open Server 2.0
Under transaction management	No
B-TREE INDEXING	
Maximum number of indexes	251/table
Maximum number of columns/index	16
Maximum size of index key	256 characters
Order options	Ascending only
Unique index	Yes
Clustered index	Yes (one/table)
Other file access methods (hash, etc.)	None

Sybase Features Chart (continued)

SQL	Support for ANSI SQL92 Entry Level coming in Release 10
Standard SQL statements	
Data definition language (DDL)	Yes
Under transaction management	No; DDL statements automatically commit
Data manipulation language (DML)	Yes
Extensions to SQL	TRANSACT-SQL
Execute operating system commands	No (yes in APT-SQL or via Open Server and remote procedure calls)
Load/unload data to/from ASCII file	Yes
Additional data definition statements	Yes
Control-of-flow logic	Yes
Outer join	Yes
Can create new table with query results	Yes
Stored queries	Yes
Case-insensitive (e.g., column names)	All database identifiers are case-sensitive; case sensitivity on searches/indexes can be configured on a server-wide basis
How create SQL queries/statements	Interactive SQL editor
Query optimizer	Yes
Syntax-independent performance	Yes
Uses table statistics	Yes
Minimum/maximum value	Yes
Average count per value	Yes
Distribution of values	Yes
Explain capability	Yes
INTERFACE TO DBMS	
SQL precompilers (embedded SQL)	Yes; C, Cobol, Fortran, Ada
Support for dynamic SQL	Yes
Module language interface	No
Call level interface	Yes; Open Client APIs; ODBC driver coming by end of 1993
INTEGRITY	
Referential integrity in data dictionary	Yes, with triggers; declarative integrity coming in Release 10
Column validation in data dictionary	Yes; rules/stored procedures; SQL92 defaults and check constraints coming in Release 10
Support for business rules in data dictionary	Yes, with triggers and stored procedures
CONCURRENCY CONTROL	
Locking levels:	
Database	Option to dynamically lock database as read only, single user, or database owner only
Table	Only through lock escalation
Row	No
Page	Yes; default locking level
Lock escalation	Rare; page-level locking will be escalated to table lock if more than 200 pages in a table are locked concurrently by a single user
Data isolation levels	Committed read, repeatable read (hold lock)
Lock types	Share (read), exclusive (update); Release 10 adds intent (allows readers, pre-empts ability to escalate to exclusive)
Deadlock detection/resolution	Yes; aborts transaction with lowest cpu time investment

RECOVERY	
Transaction logging	Yes
Roll forward	Yes
To a point in time	No
AVAILABILITY	
Online backup	Yes
Incremental backup	Yes
Unattended backup	Yes
Backup across the network	Yes
Backup based on thresholds	Yes
Online database changes	Yes
Software-based mirroring	Yes; can dynamically turn this on and off
I/O REDUCTION TECHNIQUES	
Fast commit	Yes
Group commit	Yes
Parallel checkpointing on multiprocessor systems	No
SECURITY	
Login password	Yes
Access control	
User	Yes
Group	Yes; when a user or group is given permission to execute a stored procedure, database access included in the stored procedure overrides individual or group permissions
Application	Done with user-level access which is independent of operating system user authorization
Roles	No; Release 10 includes four predefined roles: user, system administrator, system security officer, operator
Database-level access	Yes
Table-level access	Yes
Row-level access	Yes
Column-level access	Yes
Access by time of day	No
Access by location (workstation)	No
Ability to define resource limits on user queries	No; can limit the number of rows actually retrieved or updated by a user or a stored procedure
Trusted DBMS	Release 10 is C2; B1 will be optional
IMPORT/EXPORT CAPABILITY	Via bulkcopy utility; default is ASCII delimited
Import formats	User-defined in bulkcopy
Export formats	User-defined in bulkcopy

Sybase Features Chart (continued)

DISTRIBUTED DATABASE	
Location transparency	No for SQL; yes when executing a stored procedure that sends a remote procedure call; yes for OmniSQL Gateway coming mid-1993
Distributed query processing	No; coming in OmniSQL Gateway
Distributed query optimizer	No; coming in OmniSQL Gateway
Distributed transaction processing (two-phase commit)	Yes; not automatic; built into application on front end
Support for data replication	No in SQL Server; coming in 1993 in Replication Server
Database-event driven	Yes in Replication Server
Transaction-based	Yes in Replication Server
Customizable	Yes in Replication Server
Foreign data sources	Yes with Replication Server Toolkit
Store-and-forward capability	Yes in Replication Server
Switchover capability	Yes in Replication Server
Access to heterogeneous databases	Yes; Open Gateway for DB2; Open Client and Server for CICS; gateways to Oracle, Informix, Ingres, Rdb, RMS; others possible using Open Server APIs; OmniSQL Gateway coming mid-1993 will support read/write access and distributed joins across Sybase, Oracle, DB2, RMS, C-ISAM
Maximum number of simultaneously connected databases	8 databases on one server per query; unlimited for stored procedures
INTERNATIONAL LANGUAGE SUPPORT	
Upper/lower case conversion	Yes
Sorting/collating sequences	Yes for 8-bit character sets
Error messages	Yes
2-byte character set	Yes; also includes support for 3- and 4-byte character sets
Translated documentation	Yes

Illustration 10. A summary of the key features of Sybase SQL Server, including the Release 10 enhancements coming in mid-1993.

Next month's *Open Information Systems* will address
Unisys ASD Framework.

For reprint information on articles appearing in this issue,
please contact Donald Baillargeon at (617) 742-5200, extension 117.

Open Systems: Analysis, Issues, & Opinions

FOCUS: TRANSACTION PROCESSING

IBM's CICS/6000: Is Nothing Proprietary Any More?

Big Blue's Gamble Could Finally Legitimize Unix as an OLTP Platform

The open systems movement has a streak of mythology—the programmer demigods at Bell Labs, the free-spirited champions at the University of California at Berkeley, and the heroic quest for perfect portability. The dark antithesis of this vision has been the "proprietary platform." "Proprietary" was, of course, little more than a code word for "made by IBM," and, among proprietary products, the prime example had to be the CICS transaction processing monitor. But as Unix passes its quarter-century mark, the change in the landscape could hardly be sharper. Internecine squabbling has driven software developers and software innovations from Unix to desktop systems. The "lean and powerful" Unix environment is dawdling behind DOS and Macintosh in giving powerful functionality to end users. But nothing more completely demolishes the historical Unix mindset than IBM's reimplementation of CICS on Unix. IBM's CICS/6000 project not only brings mainframe-style transaction processing to Unix networks, it also clarifies just what portability and interoperability—that is, the entire open systems initiative—can, and should, be all about. It offers a model of openness that allows the development and use of vendors' technology advancements within the framework of well-defined interfaces and *measured,* rather than absolute, portability.

Of course, CICS has been the dominant transaction processing environment in the computer industry. Even the sharpest critic of IBM would concede that CICS implements a clear, comprehensive, and robust online transaction processing (OLTP) framework. And it offered the critical guarantees called the "ACID properties." ACID stands for *atomicity, consistency, isolation,* and *durability.* Basically, if a system has these properties, a company can trust that it will essentially never lose any data on that system. CICS defined not only a mechanism for transaction processing, but the understanding of what it meant to develop an OLTP system. This archetypical quality of CICS makes the development of CICS/6000 more than just another product release—it's a watershed in the status and perception of transaction processing.

CICS/6000 in Operation

CICS/6000 is not a rehosting of mainframe CICS to Unix. Instead, IBM took the CICS interfaces and reimplemented them on top of the Encina distributed OLTP system from Transarc Corporation (Pittsburgh, Pennsylvania). Encina is essentially a Unix product developed in C to run over the Distributed Computing Environment (DCE) from the Open Software Foundation (OSF). As such, its architecture doesn't look anything like that of its mainframe predecessor.

CICS SOFTWARE REHOSTS EASILY. A main point of CICS/6000, of course, is for IBM to salvage as many of its customers as it can in the face of steady defections from the mainframe platform. CICS/6000 implements all of its standard CICS directives. Mainframe Cobol programs that run under CICS can be compiled on RS/6000 for CICS/6000 using Micro Focus Cobol and a special interface library supporting CICS and IBM's Virtual Sequential Access Method (VSAM) for files.

Ongoing Development: C, Cobol/CICS—or a 4GL? CICS/6000 allows easy migration of existing software, but there remains the question of how to proceed with the development of new applications. We actually see little reason to rush away from CICS as a development framework. CICS is a mature and well-defined OLTP programming environment. The fundamental characteristics of the OLTP system—the critical ACID properties—are assured by the underlying Encina software regardless of the development approach. Our inclination would be to make a programming trial with an OLTP 4GL. As an example, Independence Technologies (Fremont, California) offers such a product, iScreen. But until transactional 4GLs mature, most organizations will use ordinary C or Cobol. We anticipate many will use both languages side-by-side, according to the circumstances of program development or maintenance work.

SAVING DEVELOPMENT STAFF. In some sense, the migration that CICS/6000 eases most noticeably is that of the *developers*. The way that mainframe-based computing operates has made it necessary for many "glass house" installations to employ good-sized staffs of Cobol programmers. Many of these people have accumulated years of experience and have senior-level positions—they're valuable employees. But, if you move them to the Unix/C distributed environment in one fell swoop, the change is very drastic. Under CICS/6000, mainframe programmers don't need a lot of training to get productive on Unix. They can just keep writing CICS and Cobol. In fact, for years now, third parties have sold Unix implementations of mainframe-style tools, like text editors. Since Cobol programmers will no longer need to learn the Unix programming model to accomplish their work, we feel that CICS/6000 offers as much in handling personnel issues as it does in handling technical ones.

BUT WHERE ARE THE DATA? The objective in a migration is not simply to get software running properly on a new platform. The software has to have something to work on. IBM provides gateways for connecting to mainframe-based DB2 and IMS databases, or ISAM/VSAM files, under the CICS OLTP framework. But the data are still remote. In order to finish the move from the mainframe to the RS/6000 host, the data must be extracted from the mainframe storage and made available under AIX/6000. This is, unfortunately, not a simple case of dumping data onto a tape and loading them on the target machine. The RS/6000's data formats are different from those of its mainframe predecessors. However, we view the data translation process as a relatively minor impediment to the migration of CICS systems. In ordinary mainframe-to-Unix migrations, data rehosting is much less costly than software rehosting.

IBM INTEGRATES CICS/6000 INTO AIX ADMINISTRATION. We see administration as a definite strength in CICS/6000. IBM has provided CICS/6000 add-ins for its System Management Interface Tool (SMIT) for AIX on the RS/6000. SMIT is one of the few truly comprehensive administrative shells for Unix. Using SMIT, you will be able to start up and shut down CICS/6000, and set operating parameters such as the number of concurrent CICS applications that can run at one time. You can also configure CICS/6000 to run automatically at system startup and shut down automatically with the system, without having to cobble together Unix shell scripts. The Encina piece of CICS/6000 is not as well integrated with AIX system management, but, for most sites, the CICS/6000 SMIT functions are likely to suffice.

Encina-Based Technology

The watchword of online transaction processing is *dependability*. Corporations bet their business operations on OLTP systems. Ordinary applications are susceptible to failure; if something goes wrong, you have to recreate the data. The objective of OLTP systems, however, is the ability to recover from a failure predictably and get back up and running with no loss of work. In this section, we will see how Transarc implemented these qualities in the Encina transaction system that underlies CICS/6000.

PROPAGATING TRANSACTION STATES. Like so many critical software technologies, Transarc's Encina OLTP system grows from a single, highly powerful, abstract mechanism. At Carnegie Mellon University (CMU), Transarc's founders devised an abstract model for controlling transactions over a network. Under Encina, the software that *implements* the ACID properties—the DBMSs or file access managers—is divided from software that *propagates* those properties. By propagating transactions in the abstract, Encina frees OLTP applications from the details of the servers they use. It is unnecessary for Encina developers to write complex code handling all of the myriad failure points in the distributed system. They simply write the code to the abstract, server-independent Encina interfaces.

Transarc implements this transaction management by augmenting the DCE remote procedure call (RPC) definition with transactional functions. The DCE RPC allows programmers to invoke software from anywhere on the network without hand-coding the necessary communication protocols or data-format translation. Transarc's revision of the RPC, Transactional RPC (T-RPC) operates by piggybacking its transaction management protocol on the standard RPC. This integrates the OLTP mode of operation with the standard RPC-based approach to programming distributed systems.

DEFINING A TRANSACTIONAL PROGRAMMING PARADIGM. Because of the paradigm under which it runs, the OLTP application runs differently from an ordinary one. At the programming level, there is an analogous difference in form. The programmer has to write the application with a constant awareness of how each line of code affects the ACID properties, particularly for failure cases. Encina greatly simplifies this work by defining a Transactional C language. Transactional C is not a material alteration of the C language; it exists simply in a set of C "header" files. These provide syntactical extensions to C for OLTP. The main extension is a "transaction" program block. You prefix an ordinary block of C code with the keyword "transaction"; at the

end of this block, you write two more blocks under the labels "onCommit" and "onAbort." This makes it very easy for the programmer to handle failure cases. The Transactional C software also works with the Encina Transactional RPC to manage transaction states.

While Transactional C may seem to be a significant alteration of "standard" C, many companies employ similar types of mechanisms for C coding. Most corporate development shops and virtually all software vendors use a whole set of C header files to standardize their programming. They use the headers to simplify and formalize portability approaches, testing mechanisms, software configuration—*and error handling*.

RUNTIME CONFIGURATION. True to Unix form, Encina runs in an extremely lean fashion. Most of the functionality resides in the Transactional C/Transactional RPC runtime libraries in the transactional application programs. There are also "application server" processes, which execute requests sent to them via Transactional RPC. There are only a handful of background or daemon processes. The Encina Monitor is the main one; it starts the application servers and routes T-RPC requests to them, manages system shutdown, and maintains a record of the distributed transactions that are executed on the system. A distinct logging process actually stores the transaction records for the Encina Monitor. When CICS/6000 is operational, it has its own monitor daemon that coordinates CICS operations over a network; the CICS/6000 monitor can work with peer monitors running under OS/400 and OS/2 as well as MVS.

CICS/6000 ADDS ANOTHER LAYER TO ENCINA. IBM made no special alterations to Encina to support CICS; rather, it took the existing Encina interfaces and functions and used them to re-implement the CICS facility. This, unfortunately, adds yet another runtime library and another layer of software between the program logic and the software that implements that logic. We have depicted the configuration of runtime libraries in Illustration A.

CICS/6000 SHOWS WIDE GATEWAY CONNECTIVITY. A little-discussed but important component of the CICS/6000 equation is gateways to existing CICS systems. Transarc ships Encina with an SNA/CICS gateway. With CICS/6000, IBM also provides interconnects to its other CICS implementations. In addition to these, Transarc also has a gateway to XA-compliant database management systems. XA is a standard protocol defined by X/Open for transaction management among relational DBMSs. These facilities are difficult to assess at the surface—in many cases, the "standards" to which software vendors claim adherence are either too vague to guarantee interoperability or too minimal to provide any substantial function. We don't see the Encina-CICS/6000 gateways as falling into this line of hand-waving. The interface definitions are robust and provide sufficient functionality for gateways to actually mean something.

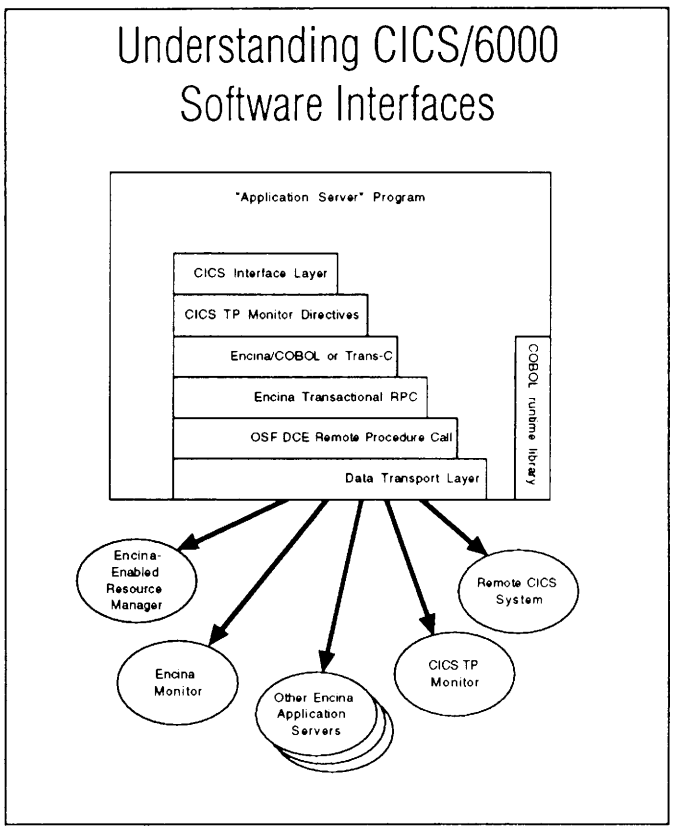

Illustration A. CICS/6000 builds up a stack of runtime libraries to implement the CICS OLTP environment under a Unix/Encina hosting.

We consider this functionality to be crucial. The mainframe CICS gateway is the linchpin for performing an incremental migration of applications from the mainframe to Unix. Now, skeptics will observe that IBM has a financial interest in making such migrations incremental—as incremental as possible—in order to continue bringing in support moneys for the big iron. But the absolute necessity of performing off-mainframe migrations incrementally, and under complete administrator control, means that gateways back to the mainframe are a primary purchase issue rather than an add-on.

At this point, distributed OLTP is a market in its infancy. With penetration just starting for single-vendor

distributed OLTP, there is nothing by which to assess inter-OLTP-system gateways. Later on, however, we look for these gateways to become an organic part of distributed OLTP operations. Not only will transactions casually span 5, 10, or 20 machines, but—depending on gateway throughput—they may cross between CICS/Encina and competing solutions with equal grace.

Competitive Unix OLTP Approaches

While we see CICS/6000 as a watershed in the Unix/distributed OLTP market, it is not really because of any technology breakthroughs. The fact is that distributable and interoperable OLTP systems have been available under Unix for several years. They lack the cachet of the IBM label, but are, in many other respects, well able to compete with Encina-CICS/6000. In this section, we will summarize the most established products.

USL'S TUXEDO. Tuxedo, from Unix System Laboratories (Summit, New Jersey), is the granddaddy of Unix OLTP systems, with origins nearly as far back as those of CICS itself. Tuxedo ports run on a dozen or more platforms, mostly by the instrumentality of consultant/integrators Independence Technologies (Fremont, California). Although it was developed as a native Unix transaction monitor, CICS interoperability was an early and ongoing assumption for the product. Tuxedo has been continually upgraded over the years and can now operate in a fully distributed mode. Tuxedo actually drove the definition of X/Open's TX interface, a generic OLTP protocol, as well as XA. However, the Tuxedo OLTP protocols are completely (dare we say it about USL?) proprietary above the transport layer. While Encina leverages session-layer and presentation-layer definitions, and naming and security services from the OSF Distributed Computing Environment, Tuxedo implements all of these under its own software. This could severely hamper interoperability not only for distributed OLTP, but for distributed applications and management as a whole. However, USL has announced plans to integrate capabilities to use the DCE naming and security services. And it is undeniably the most mature product in the market.

NCR'S TOP END. Top End was developed by NCR Corporation (Dayton, Ohio). Although it was originally held to NCR platforms, AT&T's purchase of NCR set in motion a number of changes. After the merger, Top End bested Tuxedo to be named the AT&T "strategic transaction monitor product." Concurrently, porting efforts with Independence Technologies got underway. Top End also offers the rare feature of a graphical (X Window-based) administrative facility. Although it entered the market somewhat late in the game, it has a solid reputation in Unix OLTP.

UNIKIX FROM INTEGRIS. UniKix is an offering from Groupe Bull's Integris subsidiary in Billerica, Massachusetts. UniKix includes a number of distinct products completely targeted on mainframe CICS interoperability and migration. Migration capabilities for moving databases from SNA to Unix are very thorough. Integris's XPU4 and XPU5 gateways to SNA/CICS are the most comprehensive, allowing UniKix to offer transaction services into the mainframe network. Integris even sells a number of communication controllers to wire Unix systems into SNA. If you need something practical and need it now, UniKix looks like an excellent choice. But this precise targeting toward low-level interoperability may have come at the expense of tracking high-level standards and interfaces.

VISYSTEMS' VIS/TP. Less effective than Transarc, perhaps, at working the crowd of system vendors is VISystems of Dallas, Texas. Its VIS/TP transaction processing system is an unabashed reimplementation of CICS with VSAM and DL/1 subsystems. VISystems hints that its OLTP competitors are wasting time reinventing the wheel; its obvious preference is to rehost it. This tight definition of objectives makes VIS/TP arguably the most powerful base for migrating mainframe CICS software to Unix. VISystems actually has patented a technology it uses to achieve application portability between the mainframe and a number of Unix platforms. However, it also has some gaps in integrating the OLTP system with the Unix environment. Interoperability with Unix DBMSs and the networked environment appears simply to have been put off. However, VISystems asserts that it will address these needs promptly.

DCE MAKES ENCINA-CICS/6000 THE CHOICE WITH A FUTURE. Of all these alternatives, none provides integration with any high-level protocols whatever. First of all, none uses a standard presentation/RPC facility such as the DCE RPC, or Sun RPC, or OSI ROS. Second, none uses a standard naming or directory service such as the DCE's or Sun's Network Information Service (NIS+). So far as they support distributed OLTP, the packages implement corresponding functions; Tuxedo, in particular, has a thorough implementation of both capabilities. But they lie outside any managed network environment. Meanwhile, Encina is completely DCE dependent but gains the benefits therefrom. Encina OLTP objects reside in the DCE name space. And by building from the DCE RPC, Transarc was able to define a full transactional programming environment without any

hassling with data formats. It's hard to find so solid a basis for growth in the alternatives.

...BUT PERFORMANCE MAY LAG IN NEAR TERM. Ongoing questions about the performance of the DCE have surfaced about Encina as they have elsewhere. The OSF-specific communication layers appear to be inefficient and slow. This could be a very serious problem for Encina; more than one of Transarc's competitors voiced hesitancy about using the DCE because of its performance. OLTP must not only be reliable, but it must also be fast. Near-term adopters of CICS/6000 or Encina should be sensitive to their performance needs and establish with their system vendors just what should be done to meet those needs.

A GENTEEL COMPETITION. Within the computer industry, the Unix market seems to suffer worst from a sort of "market cannibalism." Vendors have frequently launched discreet or not-so-discreet campaigns to discredit each other, resulting in damage to the entire customer base. Thus far, the Unix OLTP market seems relatively free of this destructive tendency. We attribute this primarily to action by Hewlett-Packard and Sun Microsystems. The former, in particular, is aggressively bringing OLTP packages to its Unix boxes, working with every vendor mentioned here—even IBM, from which it has licensed CICS/6000. But the OLTP vendors themselves are also wasting little time in infighting. Most or all are tying into each other's products using the X/Open and Encina interfaces. In our view, the presence of the CICS environment as a de facto standard and the obvious need to grow the Unix OLTP market without discord have kept the competition quite civil, almost friendly.

Conclusions: A New Perspective on Openness

CICS/6000 LEGITIMIZES OLTP. In terms of technology, Unix OLTP should have been a done deal some time ago. Tuxedo could easily have become a sort of "Unix CICS" already. Unix OLTP products were delivered in 1990 and 1991 which, except for market apathy, should have been the critical mass for the whole industry. The handwriting has been on the wall for mainframe systems for years, but IS managers and purchasers have been reluctant to act on it. In our view, they were unsure what platform would provide a reliable new base with an affordable and moderate migration path. CICS/6000 should meet this market need perfectly—a need not for technology, but for assurance. Frankly, we suspect nothing would better benefit USL, NCR, Integris, or VISystems than for CICS/6000 to meet with great success.

PURE PORTABILITY ABANDONED (FINALLY!). It must be noted that even an "open" CICS runs against the grain of the Unix openness cult. Under the traditional dogma, you only program to facilities you *absolutely* know exist on all current or possible target platforms. This mindset is a main reason for Unix's stunted development of high-level interfaces like those common on the desktop. Extremely few vendor-developed subsystems, protocols, or even file formats have established themselves. (Only Sun's NFS comes to mind.) All of these have, literally or in effect, been released into the public domain as source code. We feel CICS/6000 can and will change this. Commercial Unix sites have been dying for an OLTP monitor, and we think they'll bite on CICS/6000 or alternatives like UniKix. We think that the need, and the fact that this need lies outside the high-minded academic and technical installations, will allow CICS/6000 to override the quaint but self-defeating idea of perfect portability.

DELIBERATELY CHOOSING SOFTWARE DEPENDENCIES. The underlying issue in portability is the existence of dependencies among software modules and subsystems. If you develop an application using some third-party subsystem, what happens if you want to move to a platform not supported by that vendor? The possibility of facing this hypothetical situation long ago induced most Unix development shops to avoid third parties and "roll their own" facilities, like indexed file-access libraries. However, the complexity of these necessary software subsystems is growing. CICS is a class of software that few Unix installations could afford to implement. And project managers will find that dependencies on internally developed software can be as costly as those on third-party software. CICS/6000 highlights the choice lying before Unix IS managers for several types of software, namely, between a calculated compromise of portability and simply giving in to stagnant technology and incomplete solutions.

Interoperability: Realistic—But Dare We Expect It?

Portability may be receding as a key component of "openness," but interoperability is not. In fact, interoperability is simultaneously more clearly defined, more measurable, and more achievable than portability—by a great margin. API definitions are becoming more precise and more machine-independent, and their semantics more clearly specified. We should be entering a "plug-and-play" world. But we need more time to see how the OLTP market develops in this regard. For some vendors, the critical gateway back to the mainframe needs fuller development before it is tied into competitors' systems. And it is hard to say whether the vendors will be able to justify the cost of implementing multivendor distributed OLTP. The

development of the Unix OLTP market over the next year or so, particularly with regard to how well CICS/6000 sells, will be the main factor.

Openness Takes on the Layered Look

CICS/6000 has the capability of transforming the open systems market. Despite its technical strengths, Unix has no momentum whatever in many high-growth segments of the software market: E-mail and workflow, word processing and document preparation, graphics, conferencing. At the same time, it has never really gained the confidence of the IS community that it should take over handling mission-critical corporate data and applications. CICS/6000 can change, at least, the second half of the equation, making Unix a credible OLTP platform. But if the Unix market buys into CICS/6000 as a software model, it might break the logjam in end-user tools. CICS/6000 is a package running on a package (Encina) running on a package (OSF DCE). The layers are deep and the dependencies serious, but the result is simultaneously powerful and well-integrated. Well-defined interfaces lay out interoperability very clearly. We think CICS/6000 truly shows the type of packaging Unix vendors need to adapt to compete with high-volume desktop systems.

—A. Wolfe

Transarc Corporation
The Gulf Tower, 707 Grant Street, Pittsburgh, PA 15219

Interactive Development Environments, Incorporated
595 Market Street, 10th Floor, San Francisco, CA 94105

Integris
302 Concord Road, Billerica, MA 01821

VISystems, Incorporated
14755 Preston Road, Suite 200, Dallas, TX 75240

Patricia Seybold Group's

MARATHON WEEK

Actual User Case Studies

Patricia Seybold Group's
MARATHON WEEK
Conference and Technology Spotlight

April 18 - 23, 1993 • Copley Marriott
Boston, Massachusetts

Attend the *Only* Conference for *Users* Migrating to Open Systems, Client/Server, and Distributed Object Computing

3 Complete Technical Programs
Cross Participation Encouraged

Distributed Object Computing Technology Forum
How Can I Use Distributed Object Computing Technology Today to Build Mission-Critical Applications?

Open Systems Forum
How Can Organizations Build Production Systems, Save Money, and Gain Interoperability with Open Systems?

Seybold Executive Forum℠
How Can Attendees Design a Real Business Solution That Implements Open Distributed Systems?

To Register or for a Complete Program Brochure, Call (617) 742-5200, or (800) 826-2424

Patricia Seybold's Computer Industry Reports

ORDER FORM

Please start my subscription to:

		U.S.A.	Canada	Foreign
☐ *Patricia Seybold's Workgroup Computing Report*	12 issues per year	$385	$397	$409
☐ *Patricia Seybold's Open Information Systems*	12 issues per year	$495	$507	$519
☐ *Patricia Seybold's Distributed Computing Monitor*	12 issues per year	$495	$507	$519
☐ *Paradigm Shift—Patricia Seybold's Case Studies in Distributed Computing*	6 issues & tapes per year	$395	$407	$419
☐ *Paradigm Shift—Patricia Seybold's Case Studies in Distributed Computing*	6 issues per year	$295	$307	$319

Please send me a sample of: ☐ *Distributed Computing Monitor* ☐ *Open Information Systems* ☐ *Workgroup Computing Report* ☐ *Paradigm Shift—Case Studies in Distributed Computing*

Please send me information on: ☐ Consulting ☐ Special Reports ☐ Conferences

☐ **My check for $_____ is enclosed.** ☐ **Please bill me.** ☐ **Please charge my subscription to: Mastercard/Visa/American Express** (circle one)

Name: _____ Title: _____

Company Name: _____ Dept.: _____

Address: _____

City, State, Zip code, Country: _____

Fax No.: _____ Bus. Tel. No.: _____

Card #: _____

Exp. Date: _____

Signature: _____

Checks from Canada and elsewhere outside the United States should be made payable in U.S. dollars. You may transfer funds directly to our bank: Shawmut Bank of Boston, State Street Branch, Boston, MA 02109, into the account of Patricia Seybold Group, account number 20-093-118-6. Please be sure to identify the name of the subscriber and nature of the order if funds are transferred bank-to-bank.

Send to: Patricia Seybold Group: 148 State Street, Boston MA 02109; FAX: 1-617-742-1028; MCI Mail: PSOCG
To order by phone: call (617) 742-5200

IO-393

Topics covered in Patricia Seybold's Computer Industry Reports in 1991 & 1992:

Back Issues are available, call (617) 742-5200 for more information.

Office Computing Report

1992—Volume 15

#	Date	Title
3	Mar.	Requirements for Workflow—What Should We Expect from the Vendors?
4	Apr.	Desktop Multimedia—Moving beyond the Chicken and the Egg
5	May	Borland International—A Database-Centric, Object-Oriented Approach to Desktop Integration
6	June	Apple's Macintosh—Can It Become "the Cadillac of Collaboration"?
7	July	Business Intelligence—A Framework for Data Analysis Applications
8	Aug.	The Quest for Common Mail APIs - Clearing up the Confusion
9	Sept.	BeyondMail for Windows—Epitomizing the Mail-Enabled Application
10	Oct.	Microsoft's Workgroup Strategy—Moving Group Functionality into Windows
11	Nov.	Visual Programming—Application Design for End Users and Professional Developers
12	Dec.	The Notes Phenomenon—The Industry Reacts to Lotus Notes

1993—Volume 16

1	Jan.	Microsoft Access—"Cirrus" Database Project Gets down to Earth

Workgroup Computing Report

2	Feb.	WordPerfect Information Systems Environment—WordPerfect Reveals Its Blueprint for Workgroup Support

UNIX in the Office

1992—Volume 2

#	Date	Title
3	Mar.	Europe's Harness Project—Integrated Technology for an Open, Object-Oriented, Distributed Applications Platform
4	Apr.	The X Window System—Where is Its Future?
5	May	HP's Master Plan—Winning Is Everything in Palo Alto
6	June	Digital's DECworld Gems—Alpha and Accessworks Shine

Open Information Systems

7	July	Integrating Applications in the Real World—Evolution, Not Revolution
8	Aug.	Windows NT 3.1—Microsoft's Bid for Desktop Dominance
9	Sept.	Oracle's Version 7—Can It Leapfrog the Competition?
10	Oct.	Galaxy from Visix—Application Portability Breakthrough?
11	Nov.	European Open Systems Architectures—Europe's Vendors Strike out for Open Distributed Systems
12	Dec.	The Unix Data Center—Fact or Fiction?

1993—Volume 8

1	Jan.	X/Open in the 1990s—Making Open Systems Safe for Users
2	Feb.	Highly Available Open Systems—Expanding Today's Definition

Distributed Computing Monitor

1992—Volume 7

#	Date	Title
6	June	Distributed Printing—Major New Approaches Begin to Relieve One of Distributed Computing's Most Frustrating Problems
7	July	The New E-Mail APIs—Finally, Real Progress toward Mail-Enabled Applications
8	Aug.	Distributed Object Computing—The Merger of Distributed Computing and Object-Orientation into a New Architecture
9	Sept.	Gradient DCE for PCs—PC-DCE Spurs OSF to Make PCs Peers to Unix and Other More Powerful Platforms
10	Oct.	Database Interoperability—A Comprehensive Approach to Database Access for the 1990s
11	Nov	Transarc Encina—Will Distributed OLTP Systems Overrun the Mainframe's Last Stronghold?
12	Dec.	UI-Atlas Distributed Management—Object-Oriented, Distributed Management for the Unix System V World

1993—Volume 8

1	Jan.	Component Software—A Market Perspective on the Coming Revolution on Solutions Development
2	Feb.	Switched Internets—The Coming Gigabit Revolution in Enterprise Networking