Craig S. Mullins

Return to Home Page

May 1996

Computing : News&Review

Industry Trends Impacting Database Administration

by Craig S. Mullins

The IS organization has evolved from one of centralized control to an open, heterogeneous environment. Once-upon-a-time the mainframe was the centralized point of control for all application development and data management activities, but today it is only one small piece of an increasingly complex puzzle.

Managing in a Heterogeneous Environment
The biggest challenge faced by database administration groups is the task of managing heterogeneous database environments. Most organizations have more than one DBMS and many have three or four. It is not unusual for medium to larger shops to have DB2, Sybase, Oracle, Informix, and Microsoft SQL Server installed and housing production data. The fact that each of these are relational DBMSs does little to diminish the pain of

administering and managing the data stored therein.

Why is this so? First and foremost, each RDBMS vendor has implemented different procedures and syntax for implementing and managing databases. Although ANSI is working on a standard SQL, none of the vendors implement plain vanilla ANSI SQL. Each RDBMS supports a certain level of ANSI SQL compliance, but product-specific extensions abound. Try learning Sybase's Transact-SQL and see how difficult it is to translate into Oracle PL/SQL or Informix SPL. Additionally, the user interface varies from product to product making it difficult for the DBA to switch from, say, Oracle to DB2.

Furthermore, heterogeneity poses problems that would exist even if all of the DBMSs were 100% open and interoperable. How do you implement changes on multiple platforms and keep the data in multiple DBMSs synchronized? Unless a database administration tool is used that understands the open enterprise environment of the organization, synchronization must be done manually. How do you coordinate application changes with database changes? When a data structure changes, quite obviously some application program somewhere is going to need to access that data. Therefore, the program must be changed. To reduce errors and enhance integrity it would be nice to integrate the database management tool with an automated software distribution tool. This becomes increasingly complex in client/server environments where application programs needs to be distributed to hundreds or thousands of client nodes across the network.

Mining Warehoused Data

Perhaps the hottest trend in database management today is data warehousing. Data warehousing is just a new name for an age-old technique: separating production transactions from ad hoc, analytical queries. Of course, organizations implementing data warehouses today are ostensibly following a methodology that includes data refining (scrubbing and transformation), data movement (replication and propagation), and data documentation (storing metadata in a corporate repository). The true value of data warehousing is not only providing businesses with more information, but more accurate information in a more timely manner. In this way the data warehouse enables businesses to quickly react to rapidly changing business conditions and thereby gain a competitive advantage. So the data warehouse becomes as critical in terms of administration and management as the production systems are. And, guess what, the data warehouse is often implemented in incremental data marts using multiple database structures on multiple platforms. This further complicates the job of the DBA by increasing heterogeneity.

An offshoot of the data warehousing trend is data mining. Data mining is the process of discovering heretofore unknown information and patterns lurking within the organization's databases and data warehouses. Just imagine what happens to data access and database performance when automated data mining tasks are executed against legacy data.

Databases Are Going Online

The rapid acceptance of the Internet and the World Wide Web as a source for information is speeding the necessity for company's to hook their databases into the 'net. Web browsers are fast becoming the de facto method for gathering and sharing information outside (Internet) and inside (Intranet) corporations. Now that Java can be used to write applications to be executed on the Inter- and Intranet by your favorite Web browser, many administrative quagmires will develop. How do you ensure that Java applets can access your corporate databases over a vast network? Furthermore, how do you tune a database that could potentially be accessed by every Web user in the world? And how is down-time managed? Anyone who has surfed the web knows that a site can be available one second, but not necessarily the next. These are new issues that DBAs are not accustomed to dealing with.

A side effect of the Internet is the ready availability of a virtual user group — Usenet. Usenet newsgroups provide on-line access to a vast number of database experts with newsgroups available for DB2, Informix, Oracle, Sybase, Microsoft SQL Server, and Ingres.

Extending the Database

Traditionally, DBMS products stored data and nothing else. But all of the major RDBMS products of today support procedural logic in the form of triggers, functions, and stored procedures, otherwise known as Server Code Objects (or SCOs for short). Of course, not all of the RDBMSs support every SCO, nor do they implement SCOs in the same way. This greatly complicates the job of the DBA. But storing procedural logic in the database is here to stay because it enhances performance of client/server applications, eases security, and promotes reusability.

Last, but definitely not least, is the freight train known as object-orientation, or OO. The basic idea behind OO is that process should be encapsulated with data thereby creating complex objects (instead of the simple rows and columns we use with RDBMSs). Objects increase the possibility of reuse and provide structures for dealing with the complicated data that relational technology has traditionally struggled with (e.g., CAD/CAM drawings, bill-of-material hierarchies, etc.). Several pure object-oriented DBMS products are available, but none of them provide the full-functionality of the major RDBMSs making it unlikely that we will have to "unlearn" relational technology and learn OO technology in its place. All of the major RDBMS products are being re-vamped to support objects in upcoming releases, creating what is being called an Object Relational DBMS, or ORDBMS. However, the advent of ORDBMS products will further complicate database administration as objects will need to be modeled, implemented, and maintained.

Synopsis

As I sit here looking out into the future I see a Java application running within a Web browser accessing object relational data from DB2, Oracle, and Informix. Down the hall the marketing department has just discovered a new customer purchasing pattern in the sales database using an automated data mining tool. The DBAs are invoking database changes in DB2 and Sybase from a single management console with no database down time. Or am I just dreaming...

From *Computing News and Review*, May 1996.