

Network Databases: Past, Present, and Future

Robert E. Jensen, (Email: rjensen@trinity.edu), Trinity University
Petrea K. Sandlin, (Email: psandlin@trinity.edu), Trinity University

Abstract

The purpose of this paper is to review the past, present, and future in terms of three "generations" of network databases. Particular emphasis is placed upon the Internet and the booming explosion of restricted-access subsets of the Internet known as intranets. The Internet is a collection of millions upon millions of local, regional, national, and global networks. It commenced in 1969 with four supercomputers across the U. S. networked to the Pentagon. Very few people used the Internet for the first 20 years. Explosive growth took place after 1992 when the Netscape Navigator web browser incorporated HTML protocols to read HTML codes which were invented by particle physicists in Switzerland in 1990. This was the beginning of the "first generation" of network computing on what became known as the world wide web (WWW) or simply the "web." The first generation was mainly one way flows of information from web server computers to client user computers on the web. At this same time, the first generation of database interactive computing was confined to local and wide area networks (LANs and WANs). Although data files could be transmitted across the Internet using FTP and other protocols, databases could not interact on the WWW or the Internet as a whole. Most web applications are still in the first HTML generation. The second generation of networking and databases followed quickly when web servers became able to interact in a more formal way with remote clients on the Internet. With special types of middleware software, database servers could process data transmitted from remote Internet client computers. For example, customer orders and market surveys could be processed and server-side databases could be updated without human intervention. Middleware CGI scripting and later ActiveX and Java software enabled web servers, database servers, and remote clients on the Internet to become more interactive. The second generation is relatively new and growing in popularity at this time. The third generation is only just emerging and is best described as distributed network computing. In the second generation, middleware updates "front ends" of database servers on the server side when clients transmit signals. In the third generation, databases can be distributed globally and can communicate with each other with "back-end" distributed network computing. There is virtually no difference between having all databases on one computer with one operator versus having databases on 100 computers with 100 operators residing anywhere in the world.

Introduction

Whereas the Internet^{1W} commenced in 1969 with the linking of the Pentagon with four supercomputing centers at universities, the first generation WWW was conceived in 1990 by particle physicists (notably Tim Berners-Lee) at CERN in Switzerland. The CERN group developed the HyperText Markup Language (HTML) text-embedded coding tags and the HyperText Transfer Protocols (HTTP) for reading HTML at WWW sites. In 1993 there were only 50 WWW sites. Today there are over 20 million servers and an unknown number of remote clients accessing those servers. Since users come and go without registrations, estimates of over 50 million users are at best wild guesses.

The design of network computing is to have "pages" (documents) or other files on a server (or client server or web server or HTTP server)^{2W} computer that is networked either on public Internet networks (the Internet is a bundle of many such networks) or private intranet networks (that use Internet technologies). Users are connected to servers on computers called client or remote^{3W} computers.

First Generation: Static HTML vs. Dynamic Computing

Background

Very few people used the Internet for the first 20 years. Explosive growth took place after 1992 when the Netscape Navigator web browser incorporated HTML protocols to read HTML codes. This was the beginning of the "first generation" of network computing on what became known as the world wide web (WWW) or simply the "web." The first generation was mainly one way flows of information from web server computers to client user computers on the web. At the same time, the first generation of database interactive computing was confined to local and

wide area networks (LANs and WANs). Although data files could be transmitted across the Internet using FTP and other protocols, databases could not interact on the WWW or the Internet as a whole. Most web applications are still in the first HTML generation.

Between 1989 and 1990, the early developers of Version 1 of HTML embedded codes were mostly concerned with viewing text and linking to different text documents around the world. HTML code was embedded with HTML tags for certain actions. For example, the <HREF tag was followed by the "link" to another document on the same or another web server. When a reader clicked on that link, a web browser with the HTTP protocol would bring up the linked document. The text files themselves had HTM file extensions and were called HTM or HTML files. The client-server or host-server computer that houses these HTM files is often called the HTTP server. In this paper, the term "server" will be assumed to be a HTTP web server storing files that remote client computers access on the network using the HTTP protocol. If the server refers to another type of server, such as a database server, the description will be more explicit in order to distinguish it from a HTTP server.

Between 1990 and 1995, new versions of HTML arose that allowed for newer files and features such as graphics files (usually with GIF^{4W} or JPEG^{5W} compression), audio, video, and animation. In those early days of the WWW, most computing took place on the client computer after the files (e.g., htm^{6W}, gif or jpg, wav^{7W}, mov^{8W}, and avi^{9W}) were downloaded from the web server to the client computer.

In the first generation, HTML files were "static" documents on both the server and the client computers. Computing took place on both ends of a network link, but the network computing was limited to linking and downloading of files. Users could not do distributed network computing^{10W} under HTTP networking.

Readers with comments or questions are encouraged to contact the authors via e-mail.

Advantages of HTML

- HTML converters/editors are easy to use with low-cost^{11W} software (ranging from zero to \$150 in most cases). The latest word processing upgrades have limited HTML converters and editors^{12W}.
- Both HTML authors and users can be trained easily and inexpensively.
- HTML documents can be stored in cache^{13W} so that the server is not tied up every time the client user wants to return to an HTML file (files can be stored in the browser's cache for short periods of time even if the user does not formally download and save the HTML file in a designated directory).
- HTML documents can be easily printed using browser menu choices (linked graphics appear on pages as if they were pasted onto the document itself).
- It is very easy to modify the sizes of graphics images. A stored gif or jpg file can be viewed in a wide range of sizes (although increasing the size beyond the stored image size may result in pixelization).
- HTML documents are easy to search and have given rise to popular web search engines^{14W} (e.g., Yahoo, Altavista, Lycos, HotBot, etc.).
- An HTML document can be viewed on multiple platforms^{15W} (Windows, Macintosh, UNIX workstations, etc.).
- HTML on the web can be networked across existing Internet networks.
- HTML documents can contain links to graphics, audio, video, and animation files.
- HTML source codes are easy to view and modify--usually with the menu choice (View, Source) in a browser.
- HTML downloads are virus free when read in a web browser. This is not the case with many other types of downloads such as word processor files, spreadsheet files, and other types of files that web users may find available for downloading.

Disadvantages of HTML

- HTML is "document" rather than "data"

centered and does not facilitate distributed network computing^{16W} or relational database management utilities.^{17W}

- HTML is static and cannot make arithmetic calculations, date/time operations, perform Boolean logic, or revise data on the client or host computers. Users cannot add 2 + 2 in HTML code.
- HTML cannot be coded to conduct searches (although other software can be programmed to search HTML documents).
- HTML cannot tabulate survey responses (even though surveys can be conducted using forms in HTML documents).
- HTML cannot perform security operations (authorize password clearances, authenticate servers or clients, encode and decode transmissions, etc.).
- HTML cannot react to signals such as messages or user choices in a response.
- HTML on the web requires connectivity^{18W} to the web which, in turn, requires monthly or annual fees and the frustrations of delays caused by clogged networks having insufficient bandwidth^{19W} (especially for users that must use slow modem^{20W} connections).

Efforts are underway to create standards for a new Dynamic HTML (DHTML) in a Document Object Model (DOM). However, progress is slow and will take years according to "A Tangled Web of Standards," *Internet Week*, September 27, 1997, p. 1. For more on the WWW Consortium dealing with such issues, see <http://www.w3.org/>.

In the meantime, some vendors are coming out with limited dynamic HTML. Macromedia's Dreamweaver^{21W} uses cascaded style sheets (CSS) supported by the latest versions of HTML browsers. Dreamweaver builds timed animations and also has built in JavaScript snippets for popular applications such as form validation of user responses.

Dynamic Computing with Plug-in Apps

Browsers such as the Netscape Navigator automatically download certain types of files into

a temporary folder on the client's computer. The directory is referred to as the browser's cache directory or folder. The Internet connection stays open only long enough to download the files. Typical files such as htm (HTML), gif (GIF compressed image), jpg (JPEG compressed image), wav (Audio), mov (QuickTime video), avi (Microsoft video), and fli (Autodesk animation) files are initially downloaded to cache. Users must save them to other folders if they want to keep them for longer periods of time.

The htm, gif, and jpg files are processed directly by the web browser such that no special software needs be added to view the HTML text and images. However, browsers will not run wav, mov, avi, and fli multimedia files. Nor will they run other types of files such as ToolBook, Director, Authorware, or others. that might be encountered at a web site. Users must download or otherwise install browser "plug-ins" or other application (app) software that the browser can launch. Different users may use different applications software.

Although various types of files can be downloaded and run if the web server is properly configured to transmit the files and the client computer has the app software (often called "reader" software) for certain types of applications, dynamic computing can be done if installed software (often called a helper application or helper app) is installed on the client computer. The browser only serves to launch the helper application software that must be previously installed and configured for each client machine. In order to launch the app, the browser uses Multipurpose Internet Mail Extension (MIME)^{22W} to tell computers what kind of program to use to view or run a file. Mimes are typically plug-ins to browsers that help launch helper apps.

Helper app file extensions can become browser plug-ins by adding the helper app's file extension to the browser's directory of plug-ins.^{23W} The browser will then match the app's file extension (e.g., pda, wav, avi, mov, fli, tbk, etc.) with the MIME type. This plug-in requires installation of app software to execute certain

types of files that can be downloaded from the web while using a web browser. The browser can be made to automatically launch the other software as a plug-in declaration in the browser. For example, Adobe Acrobat PDA^{24W} (Portable Document Assistant) files are common on the web (these are not HTML files). Adobe offers a free reader app program that can be downloaded permanently into a client's computer. The browser can be set up so that whenever the user clicks on a PDA file at a remote site, the Acrobat Reader automatically reads the downloaded PDA file.

Although PDA files are static documents (users cannot even edit text in PDA documents), other types of files that can be similarly downloaded are not static. For example, Macromedia Director and Authorware files can be read from a plug-in called Shockwave.^{25W} Asymetrix ToolBook files can be read from a plug-in called Neuron.^{26W} Multimedia files (audio wav, video mov, video avi, animation fli, etc.) may be downloaded and installed on the client computer. Many are free.

Note that although the computing for most plug-ins takes place on the client's computer, the webmaster of the server computer must properly configure the server so that clients at remote sites can read the server files that are downloaded into the client's version of Shockwave, Neuron, and other readers of dynamic files. Not all files may be downloaded and executed in the first generation sense. Some files such as spreadsheet files must use third generation software such as ActiveX to be downloaded and run efficiently in client machines. This topic is discussed in the third generation section.

Illustrations of Shockwave and Neuron in education are listed at <http://www.trinity.edu/~rjensen/links/prelim.htm>.

One of the more interesting Shockwave web sites is the "shockwaved alternative" web site of the accounting firm of Arthur Andersen.^{27W} It is not possible to use the "shockwaved alternative" unless the Shockwave app is down-

loaded from a Macromedia web server and installed in the client computer.

Dynamic Operations with JavaScript and Java

Instead of totally separate app programs, Netscape Corporation developed a way to perform dynamic operations (calculations, Boolean logic, etc.) by embedding dynamic codes among the HTML codes. Netscape Corporation developed JavaScript^{28W} codes (not to be confused with Sun Corporation's full-blown Java^{29W} software) that can be embedded in HTML documents much like HTML tags are embedded in text documents. The Netscape Navigator executes JavaScript on the client's computer without having to send data back for calculations on the web server. For example, JavaScript can be used to perform calculations, perform Boolean logic, perform security checks, conduct searches, send messages, and perform other dynamic tasks.

Two of the main executable content power user alternatives in browsers are Java and ActiveX. Both will be discussed at a later point as third generation options. Both can, however, be used for client side computing. Java applets^{30W} can be embedded in HTML much like JavaScript is embedded in HTML documents. Sun Microsystem's Java^{31W} began as an easier way to write C++ code, but Java has since evolved to be much more than that. Java is portable, downloadable embedded "applet" code that can be easily integrated into browsers. Both the Netscape Navigator and Microsoft Internet Explorer browsers integrated Java into their latest upgrades. Any Java class or applet that is sent down to the browser can be executed interactively. Illustrations of applets can be found at <http://www.sun.com/javastation/demos/>.

JavaScript and Java can be used in the context of first generation client-side computing, in second generation network computing, or in the context of third generation distributed network computing. Java is destined to become the platform-independent "write-once, read everywhere code." At the time of this writing, Microsoft Corporation is a stumbling block because of

its refusal, in a bid to make its version of Java more proprietary, to include Java Remote Method Invocation (RMI) and Java Native Interface (JNI) in the Microsoft version of Java and the Microsoft Internet Explorer browser. Sun Microsystems has sued Microsoft to either force Microsoft to include these interfaces or to not write Java development software. Irrespective of how the lawsuit is resolved, major players like Sun Microsystems, IBM, Borland International, Symantec Corporation, and others will probably make Java the development tool of choice for dynamic networking applications.

Second Generation: Interactive Servers and Clients

Online Transaction Processing (OLTP)

Second generation network computing extends data transfers from the client computers back to the web server and/or database server computers. Software for doing this is commonly termed "middleware."^{32W} Middleware is software that mediates between an application program on a server and a network of client machines. Middleware manages the interaction between applications across the heterogeneous computing platforms of client computers. Java will likely replace CGI in the second generation because of its ability to be used on multiple platform servers as well as client computers.

The best known, but fading, second generation UNIX-based^{33W} dynamic computing utility on HTTP web server computers is called Common Gateway Interface (CGI).^{34W} CGI allows the server-side computer to react to actions of users on client-side computers. CGI is a UNIX-based set of programs/devices that supply interfaces between browsers and servers on the Internet. It is a popular older standard for running external programs from a World Wide Web HTTP server. CGI specifies how to pass dynamic components to the executing program as part of the HTTP request. For example, it will allow answers typed into an HTML form on the client computer to be tabulated and stored in a database on the server-side computer. Com-

monly, the server-side CGI program will generate some HTML which will be passed back to the client's browser. It can report to a client user that a form is not filled out properly or report the invoice total of an order. In addition, the CGI program can access information in a database, format the results as HTML, and transmit a message to client machines. CGI is not a programming language. There are various "CGI" scripting programs; PERL is a common choice for writing CGI scripts in UNIX code. Some HTTP servers require CGI programs to reside in a special directory, often "/cgi-bin," but better servers provide ways to distinguish CGI programs so they can be kept in the same directories as the HTML files to which they are related. In order to improve performance, Netscape devised NSAPI and Microsoft developed the ISAPI standard which allow CGI-like tasks to run as part of the host server process, thus avoiding the overhead of creating a new process to handle each CGI invocation.

In spite of its widespread use, CGI is giving way to Java and ActiveX discussed previously in this paper. Java applications are easily ported both across multiple-platform clients and multiple-platform servers. This means that it no longer matters whether any networked computers are using Windows, Macintosh, UNIX, or some other type of operating system. However, Java and ActiveX are much more suited to the third generation of distributed network computing which is discussed later in this document.

Electronic data interchange (also called Electronic Data Invoicing) is more concisely referred to as EDI. EDI evolved as a dedicated telecommunications line between a vendor and a supplier, dealer, or customer. EDI brokerage is very expensive and usually is limited to large organizations that deal extensively in invoicing. EDI replaces paper flows with electronic ordering and invoicing.

EDI, itself, is now being replaced by the use of extranets, which are subsets of the Internet. They operate much like intranets, but are restricted Internet connections between vendors

and suppliers, dealers, or customers. For example, in October 1997, Toshiba America started using an extranet with which over 350 dealers could order products, parts, and supplies from a secure web server. Extranets utilize web technologies.

Illustrations of Second Generation Network Computing

To view an illustration of second generation network computing, click on <http://www.dell.com/store/index.htm> and note how any person in the world can choose different combinations of attributes of a computer and signal Dell Corporation to provide a price quotation. After choosing a given combination of hardware and software features of a particular computer, a user may click on the "Update Price" button. The signal goes to the Dell server-side computer that re-computes the price and then transmits that price back to the client machine. In addition, users may fill a shopping cart with hardware and software, compute an updated price for the entire invoice, and place the order from a client machine.

Federal Express^{35W}, UPS^{36W}, and various other package delivery services have established web sites that allow customers to access parts of their internal database system. These companies have invested a great deal of money and time over the past several years to make their database systems valuable tools to track packages. Previously, customers would have to call the company on the phone or make an office visit, and a customer service representative then would have to log into the database in order to track the package. Customer direct access reduces the need to have customer service agents and other employees servicing customers.

Another example is the virtual Ernie^{37W} web consultant of Ernst and Young.^{38W} Ernie resides on a server-side computer and dynamically interacts with paying customers to give advice on financial reporting, taxes, financial planning, real estate, information systems, computers, and other areas.

Yet another example is provided by the MapQuest^{39W} web site best known for responding (at no charge) to requests for locations of a village, town, city, or country road. Maps are returned with alternate levels of zoom options. There are also other services provided by the CGI bin at MapQuest, including the following: Find a Place, Driving Directions, Dashboard, MoveQuest, TravelPlan USA!, Personalized Maps, Map Shortcuts, New Technology, Enter Our Contest, ShopQuest, and Go To Map.

An illustration of querying in second generation software is provided by the name, address, and phone number database of Switchboard.^{40W} The user submits a search request using a web browser, and the message is sent over the Internet or intranet to an HTTP web server, in this case, Switchboard. HTML documents and CGI programs are stored in the web server, and Switchboard's server-side computer accesses a certain CGI program which sends a request as a Structured Query Language (SQL) argument to a database server. The database server locates the requested information and transmits the information back to the web server, and the web server formats an HTML page and sends it back to the web browser on the client computer.

Advantages of CGI Scripting

- CGI scripting merges the static (document-centric) HTML world with the dynamics of data-centric computing.
- CGI scripting can be used to receive incoming requests from a client computer, process those requests, send out SQL queries^{42W} to a database server, process the returned query answers, and prepare a HTML report to be sent back to the client computer that initiated the request or to a subset of client computers.
- CGI scripts run on multiple platforms^{43W} (Windows, Macintosh, UNIX, etc.) provided both server-side and client-side computers are configured properly. For example, when results of a survey or orders for products arrive at the server, it does not matter what operating systems are used on the remote client computers.
- CGI scripts are designed to operate under the popular HTTP protocol. Because of this, CGI scripting provides a means of open-ended serving (e.g., any potential customer might respond from anywhere in the world).
- CGI scripts are a way of creating an intranet.^{44W} These scripts can be used to generate interactions and automatic messaging between restricted subsets of employees or dealers.
- CGI scripts reside on web servers and do not require installation of special software on the client computers (other than a browser that will run HTTP protocols). They require no special installation of helper apps such as Adobe Acrobat, Shockwave, and Neuron readers.
- Users who are familiar with browsing the web generally require no added training to provide inputs since they only have to deal with HTML documents (e.g., surveys or order forms in HTML codes).
- CGI scripting is a simple way to handle hundreds, thousands, or millions of users. For example, from its web site alone, Dell Corporation sells over \$2 million in hardware and software orders per day.
- CGI scripting reduces labor costs. If a Dell Corporation customer places a web order, Dell does not have to pay labor costs to respond to a telephone conversation, read a letter, or place a work order.
- CGI scripting enables the server-side computer to become a virtual wizard (employee) who knows many things and does many things like a robot.

Creation of CGI Scripts

There are several different tools and approaches that are available today: CGI scripting, hand coding in traditional languages like C or C++, and extending HTML with certain tag elements. For CGI scripting, the dominant approach is a UNIX scripting language called PERL. Several database vendors and universities adapted PERL and made some extensions to it that allowed it to access databases. There are versions of PERL called ORAPERL and

SYPERL that allow the user to access ORACLE and SYBASE databases. Typically, the actual script code is part of the HTML page, so when that page is sent back, that script is then run within the server.

There are some performance issues with CGI options. First, CGI scripts may not be as fast as a compiled executable program. Second, an individual CGI process must start up each time the web browser clicks the web server. So if any user makes a new or renewed request to the web server, the web server has to start a whole new process and initiate an entirely new connection to the database. CGI scripts are not efficient with large numbers of users, and many users may get turned away due to those inefficiencies.

Beyond CGI with Customized HTML Server Tags

Another approach several software developers have taken is to extend HTML with their own customized (proprietary) tags. These tags lie dormant until requests from the client machine are sent back to the web server computer. Within those tag markers, server systems can utilize 4GL (fourth generation database languages).^{45W} This has been done by vendors such as Informix^{46W}, Sybase^{47W}, and Cold Fusion.^{48W} Essentially what happens is that the HTML document itself contains a 4GL combined with Structured Query Language (SQL) statements. When those HTML tags are sent back to the web server, the web server shifts into a specific application on a web and/or database server.

When a server runs into its own proprietary tag, it will execute the command (such as make a query to a database server). The database server returns the information to the web server. Again, within the proprietary tag environment, vendors typically have laid out how they want to present the information in the HTML document itself. This is a nice, simple approach that is very obvious and easy for people who have both HTML knowledge and fourth generation language knowledge. However, it does have some limitations in that the code is

actually being put into the HTML document. The graphical user interface is not separated from the actual application and database access.

A disadvantage of using customized tags is the high overhead costs that can result if changes need to be made. If there are hundreds of pages that have SQL code^{49W} in each page, as the data structure changes, the changes have to be made to each of the appropriate pages. Also, customized tags tend to lock users into a specific 4GL. This is a good approach, however, for users who are reliant upon certain 4GL and have made major investments in them.

Beyond CGI with Fat Client OLTP Computing

Another approach that application development tool vendors are taking is trying to adapt existing tools to the web. Appropriate "fat client" software must be installed on the client computer (which excludes users not having this software at hand). An example is PowerBuilder,^{50W} which provides a mechanism whereby developers can regulate HTML pages that make a call to the web server. The web server, in turn, fires off a local PowerBuilder application on the client PC. That client PC is really using the PowerBuilder application within the browser window (an actual PowerBuilder window can be seen), but the client is now in direct communication with the database. This has certain advantages in that database owners are able to reuse existing applications fairly quickly and easily.

The downside, as mentioned above, is that only fat clients with PowerBuilder installed can use the system. The PowerBuilder application actually has to reside on each of the client PCs. There are, of course, network mechanisms to help facilitate downloading the fattening software, but it does introduce some complexities and some maintenance on the client side. The software is more complex and less user friendly than most common helper apps such as Shockwave. Lang and Chow (1996) describe the benefits of PowerBuilder as follows:

This approach has several benefits. The obvious benefit is that a Web database application can include full-strength OLTP database interactions. Another important benefit to this approach is version control: it is well known that maintaining correct versions of client applications across a population of users can be difficult and can lead to errors. Distributing client applications as plug-ins will greatly simplify the process. Each time a user launches an application from a link on a Web page, only the latest copy of the client application will be downloaded and executed. Also, users will only need to learn one program (the Web browser) and use it to access many client applications. By doing this, you can reduce the cost and time of training. And finally, users won't have to download and install a new plug-in for every client application that they want to use. That's because once the PowerBuilder plug-in program is installed on the Web browser, it will be able to run any client application that has been created for it.

PowerBuilder 5.0 plug-in applications use a session protocol rather than HTTP. This makes it easier to program transactions but it also reduces the total number of clients that can connect to a database server during a given time. So, PowerBuilder 5.0 includes code that makes a database connection when a transaction is being submitted and disconnects when a transaction has been completed. This works automatically behind the scenes, which simplifies programming.

*Plug-in applications built with PowerBuilder 5.0 will work over the Internet or on Intranets by using the session-oriented protocols provided by DBMS vendors or by third parties (for example, DataRamp for ODBC data sources...or SQL*Net for Oracle).*

Database Publishing on the Web & Intranets, © 1996 Corilis Press,^{51W} by Curt Lang and Jeff Chow, as quoted at <http://nt.powerbuilder.com/vol3num2/web1/powerbui.htm>.

A Comparison of Eight Relational OLTP Networking Systems

Relational database management systems (RDMSs)^{52W} can now be managed via intranets or the Internet. The newer software are built specifically for building web to database applications. They typically are visually developed environments, very much like the client server tools that people are familiar with, such as PowerBuilder. They are typically based on a scripting language or some sort of code generation, and they have an application engine of some type that interprets and executes the application. They fit into the web three-tier development model by trying to separate the HTML from the application itself, which allows the user to have reusability and maintainability. An excellent example of one of these types of tools is Sapphire/Web from Bluestone.^{53W}

Nance (1997) compared eight RDMS systems and reached the following conclusions:

Assigning an expert database programmer to a Web site creation-and-maintenance project is akin to having a house painter show up every day to touch up nicks and scrapes on your walls. Given the variety of paint and painting methods, and the fact that learning how to wield a paintbrush is easy, bringing in a professional to handle such tasks would be a waste of his skills and your money. The same goes for hiring a programmer to create and maintain your Web site.

More flavors of database tools exist than there are shades of white house paint. This is fortunate, because you need to assemble a range of tools for relational database management systems (RDBMSs)--query tools, schema design tools, replication tools, monitoring tools, optimization tools, security tools. A relatively new and popular category of database tools is Web-oriented. The latest version of RDBMS products from IBM Corp., Microsoft Corp., and Oracle Corp. include software aids for incorporating database content into Web pages; many third-party software vendors also offer similar tools for your database toolbox.

We evaluated these Web-database integration tools on an intranet consisting of Pentium Pro-based NT Server 4.0 machines, an Ethernet LAN running TCP/IP and a variety of clients (Windows NT, Windows 95, OS/2 Warp and Macintosh System 7). In each case, we connected via Open Database Connectivity (ODBC) or native interface to Oracle 7.3, SQL Server 6.5 and DB2 Universal Database 5.0.

Tools for accessing a database in a Web environment differ in three key aspects: database interface, Web server interface and visual design environment. In our evaluation of the current crop of database and Web integration tools, we examined carefully the extent and quality of code generation, we looked closely at how the product's user interface helped build Web pages containing database content, and we tested the variety of presentation options for database material. To get top marks, a product needed to exhibit several characteristics: native connectivity to the major databases and ODBC connectivity to all other databases; NSAPI, ISAPI, WRBAPI and CGI connectivity to Web servers; production of robust, uncrashable JavaScript, VBScript and Java code; no-programming-required, drag-and-drop construction of database-oriented Web pages and associated software; and the ability to display database content in table, list and single-entry formats.

The following eight toolsets were reviewed: Bluestone Software's Sapphire/Web 4.0; Borland International's IntraBuilder; EveryWare Development Corp.'s Tango Enterprise; IBM's NetData; Microsoft's Active Server Pages technology, Web Assistant and Visual Interdev; Oracle's Developer/2000; Prolifics' Prolifics 2.0; and Sybase's NetImpact Dynamo. In addition, we reviewed two mainframe-oriented tools to help you Web-enable legacy applications--Host Publishing System from Attachmate Corp. and InfoSession from PLATINUM technology. (See "Mainframe Tools for Web Database Access," p. 60.)

Of the tools we evaluated, Bluestone's Sapphire/Web stood out for its clean, intuitive interface and excellent code generation, and it earns

our Editor's Choice award. Not to be easily outdone, Borland's low-cost IntraBuilder exhibited a rich, yet right-to-the-point interface and comprehensive database support; we give it our Best Value commendation. However, all of the tools reviewed had strong points that merit your consideration.

Bluestone Software Sapphire/Web 4.0 connected to more databases and Web servers, generated more robust code in more languages, and offered easier-to-use drag-and-drop creation of Web pages containing database material than any other product we reviewed. It's a winner.

A full-featured development environment for Web-based programming, Sapphire/Web made short work of creating a complex, robust, database-oriented Web application in our testing. Particularly impressive in our lab-based workout were Sapphire/Web's code-generation wizards. Depending on whether we selected Java Framework or ActiveX Framework, these programming robots emitted Java, ActiveX components, JavaScript and VBScript for us. The generated code was the most robust of all the code produced. Additionally, Bluestone says it's continually producing new wizards and improving existing ones.

"DBMS Tools for Web Integration: Pour It On!" by Barry Nance in *Network Computing Online*, September 8, 1997
<http://techweb.cmp.com/nc/817/817fl.html>

Summary of OLTP Database Networking Issues

While the first generation of global networking was document centered with static HTML codes embedded in text documents, the second generation commenced with helper app browser plug-ins and CGI applications which provided dynamic linking of clients, web servers, and database servers. Several complexities inherent in second generation online transaction processing need to be considered.

Scalability^{54W}

Scalability refers to how well a system

performs as it grows. For example, a central server with ten clients may perform efficiently; however, it has a scalability problem if it fails with an increasing number of clients.

Another scalability issue is the connection to the database server. The connection to the database needs to be persistent as well as multiplex. If the connection is persistent, clients do not have to open new connections each time they access the database, and that improves performance. When the connection is multiplex, clients can log in using a generic log-in name. This allows the web server to keep only a few connections open to the database, which is far more efficient for the application as well as for the database.

Oracle^{56W} provides a network computing architecture underlying foundation that applications can then plug into, and these applications are called cartridges. Cartridges can communicate to back-end Oracle databases, and they can also communicate to web servers. This middleware, with the web server, also can provide a new mechanism to maintain persistent connections from the web browser to the web server and the database. This technology is very powerful, and it enables users to move to the next step in developing and deploying web based applications.

Start-up Costs

One disadvantage of using CGI is the start-up costs inherent in its use. Each time a client uses a CGI application, that process has to start up again, and then, typically, that process has to make a new connection to the database—sometimes an expensive operation.

The first solution generated to solve this problem was the use of Application Program Interfaces (APIs).^{55W} Netscape Corporation and Microsoft both provide APIs called NSAPI and ISAPI that essentially extend their web servers, and they provide developers with a way to put application code actually within the web server. This means that a separate process does not have

to be started each time one of these applications is accessed, and, since that application is always running, it can maintain connections to the database.

There are some disadvantages in using these new APIs. Errors in application code can wind up crashing the entire web server. In addition, overhead is involved in developing the APIs and the APIs are specific to Netscape or Microsoft browsers.

Disconnected States

A disconnected state is another common issue to resolve in networking. A client's browser connects to a web server only when receiving a file, and a web server connects to that browser only when it is sending a file. When the file is loaded in the browser, the network connection is broken. The web server does not necessarily know how to reconnect to the client or what at what point the client is at in processing the file. The web server might require clients to do a sequence of operations, but it has no control over whether they will make it through all of those operations, nor does the server know exactly where they are in the process.

There are several approaches to this issue. The first is the use of cookies,^{62W} which were invented by Netscape. The basic idea is that the web server can embed a cookie within an HTML document that gets sent back to the browser. The browser does not display the cookie; the cookie is actually encrypted. It can be up to 2Kb long, so some information, but not an unlimited amount, can be stored in the cookie. The cookie is sent back loaded with information when the client connects to the web server. For example, the identity of the client user can be stored in the web server. Cookies are actually applets that enable a web site to collect information about each user for later reference (as in storing cookies in the cookie jar). For example, a cookie allows client customers to fill their orders (shopping carts) and then be billed is to store the information in the hidden field based upon the information in the cookie.

A second approach similar to the use of cookies is the storage of information in the hidden field of frames. Again, the information is not displayed. This method does not have the automatic encryption capabilities that cookies have, but it does give the same basic approach. This is useful for a user base that has browsers that do not support cookies, although both Internet Explorer and Netscape Explorer have cookie jars.

Another approach is to use a generic state server which has the capability to store transaction information. It may be very useful to the seller to receive information about the state of progress of a client user (e.g., the number and types of orders placed in a shopping cart at a certain point in time). For example, the fact that a customer ordered a shirt as the first item and then ordered a pair of deerskin mittens as the second item might be important to the seller. While a generic state server has the ability to store transaction information, it also has the capability to erase that buffered information after a specified period of time. This may be useful because the client user may choose those two items, but may never complete the actual transaction.

Security

Two basic issues revolve around Internet security when dealing with web based applications. The first issue is the connection between the browser and the web server. Encryption^{57W} technologies have evolved into the standard encryption mechanism that is used in browsers and servers today which allows a message to be encrypted as it is moving between the browser and server. Encryption is cryptographic conversion of data into cyphertext in order to prevent any but the intended recipient from reading that data. There are many types of data encryption, and they are the basis of network security. Common types include Data Encryption Standard^{58W} and public-key^{59W} encryption. The browser and server understand the key to unlock the encryption algorithm.

The best known security device is called SSL (Secure Socket Layer).^{60W} The SSL provides a mechanism for authentication. For example, in banking transactions, a SSL guards against unauthorized clients logging in and conducting banking transactions on someone else's account.

The second security issue is preventing unauthorized users from entering a database. Firewalls^{61W} have emerged as a standard way of separating the external world from the internal world. Typically, companies implement their Internet with external web servers separated from the internal side with firewalls. What has emerged in technology is a way to have application servers running inside the firewall communicating to the web server outside the firewall. This means that users or potential hackers cannot come in and access the actual application server. They cannot access the actual database library or the database itself. This provides security of the actual application, of the business logic, and of the data from potential hackers in the outside world.

Information Warfare Risks

There are ever increasing risks of networked databases being inflicted with fraud, theft, computer viruses, worms, trojan horses, logic bombs, trap doors, electronic jamming, HERF Guns, and all the bad things that crackers (hackers with evil intent) can do to a system in criminal and warfare acts. There are also risks of unintended crashes due to equipment failures, poor internal controls, and other risk factors. Some of the most important security problems are discussed below:

1. Many (some estimates run higher than 90 percent) computer crimes are never detected. This is largely due to the sophisticated ways in which high-tech criminals can network among a rat's maze of networked computers and leave global trails that are virtually impossible to follow. Valuable products and services can be lost while being accounted for as legitimate transactions due to well-designed

criminal software or poorly designed legitimate software. Intentional sabotage may be erroneously written off as acts of nature.

2. Computer crimes that are detected are difficult to prosecute (estimates are that less than one percent of the crimes detected lead to convictions). This is partly due to the tendency of courts toward leniency for white collar crimes. It also is a result of a traditional judiciary preference for direct evidence (eye witness, smoking gun types of evidence) which is commonly lacking in computer crime.
3. An employee or outside cracker may plant a type of "time bomb" in a system that "sleeps" for months or years before becoming active. This complicates detection of a crime and ultimate prosecution. In addition, sophisticated software trap doors may be planted for criminals to enter a system at will without being detected.
4. Legitimate hardware and software vendors may be unaware of security risks and customers may assume that, because of the grand reputations of the vendors, such risks are not present. For example, neither Microsoft Corporation nor the many Microsoft Windows NT users were apparently aware that passwords were easy to "crack" when computers using Windows NT were networked to computers having other operating systems.
5. Database systems that become more centralized for networking efficiencies and effectiveness become more vulnerable to monumental system crashes that can have an impact on an entire organization, even a global organization. Crashes that occur, for whatever reason, may become disasters on a grand scale (e.g., the shut down of an entire airline reservation system or securities market).
6. Ways of conducting information warfare are being invented that may bring down giant organizations or even entire nations. In the paper cited below, Haeni (1997) contends that the means of information warfare are now

being developed by high-tech societies and armies, and these may be more destructive on a global scale than nuclear, biological, and poison gas weapons.

For further particulars on network database security issues, refer to an online paper called "Information Warfare: An Introduction," by Reto E. Haeni, Cyberspace Policy Institute, The George Washington University, January, 1997. Both HTML and PDF versions can be found at the following web address:

<http://www.seas.gwu.edu/student/reto/infowar/info-war.html>.

A good summary of terminology and network security issues can be found at this web address:

<http://www.bus.orst.edu/faculty/brownc/lectures/virus/virus.htm>.

A helpful set of links on web safety is provided by PBS's Life on the Internet at

<http://www.pbs.org/internet/relatedsites/bs/index.html>

Server-side processing in the second generation is becoming increasingly powerful. However, the second generation may place excessive strains and overhead costs upon web servers and database servers. That's where the third generation, the network computing generation of web applications, comes into play.

Third Generation: Distributed Network Computing^{63W}

Client and Server Back-end Interactions

A third generation of distributed network computing is emerging using what is termed "back-end computing."^{65W} Back-end computing allows clients to have much more interaction with servers, especially database servers.

While first generation static HTML documents and second generation dynamic plug-in apps served many needs, there were severe limitations in becoming data-centered, as opposed to document-centered, on the web. One of the major limitations of the second generation was that dynamic plug-ins required clients to install special app software and configure their computers to run downloaded programs. This was not seamless and automatic network com-

puting. Also, server computers had to be specially configured for each type of app needed for files placed in the server. This was an improvement over the first generation, however, in which the web server and client computers did not dynamically interact.

In the third generation, back-end computing is the final stage in a process or a task not apparent to the user. A common usage takes place in a compiler. A compiler's back-end generates machine language and performs optimizations specific to the machine's architecture. The term can also be used in the context of Open System Interconnect (OSI) network applications. A standard for layering of protocols (protocol stack) was developed in 1978 as a framework for international standards in heterogeneous computer network architecture. The architecture is divided into seven layers (lowest to highest): physical layer, data link layer, network layer, transport layer, session layer, presentation layer, and application layer. Generally each layer uses the layer immediately below it and provides a service to the layer above in a "back-ended" way.

Third generation computing takes advantage of the new interactive server/client technology like Sun's Java^{66W}, Microsoft's ActiveX/CORBA^{67W}, Microsoft's Visual Basic VBScript^{68W}, Netscape's JavaScript^{69W}, and VRML.^{70W} Users on the client side can interact in various ways such as perform sensitivity (what-if) types of analyses. Both Netscape and Microsoft have upgraded their browsers for running Java. Other options are not as general. Only Microsoft's Internet Explorer can execute VBScript and ActiveX applications. Most browsers other than Netscape Navigator cannot execute the popular JavaScript.

The third generation of distributed network computing will be a foundation for the powerful trends in "virtual community" interactive computing. An excellent book by Hagel and Armstrong titled *Net Gain: Expanding Markets Through Virtual Communities*^{64W} provides a good analysis of this trend.

Distributed Network Computing Illustrations

One of the best illustrations of third generation distributed network computing can be seen in the use of the new network computers (NC),^{71W} which rely upon servers for operating systems and software that in other computers are normally kept on resident hard drives. This is third generation interaction in its purest sense, between servers housing the software needed and data which is networked long distances to remote client computers which do not have the software.

CBS News used a third generation JavaScript distributed network application during the 1996 elections in the United States. CBS used a combination of client-side interactive processing with back-end access to real time election results. A geographic map of the 50 states appeared on the web page. If the user clicked on a given state such as Texas, election results for Texas appeared in a frame on the top of the screen. On the back-end, the web server computer extracted election data, in real time, from all the Texas precincts. Users could follow both the national and state election results in real time, including results of referendum proposals.

Other examples of third generation distributed network interactions can be found at Virtual Reality Modeling Language (VRML) sites. VRML is an Internet standard for 3-D animations. Information on this topic can be obtained by using almost any WWW search engine and a combination of search terms. For example, using <http://www.altavista.digital.com/> with the search term "VRML" resulted in 80,073 hits on December 22, 1996 and 359,660 hits on September 19, 1997. VRML is designed to be a dynamic extension of the HyperText Markup Language (HTML) standard that became the main component in the invention of the WWW. In other words, VRML is intended for the WWW. In addition, VRML is intended to be somewhat like VR in its ability to immerse participants into simulated 3-D worlds for education and entertainment. Also, VRML can bring animation, audio, and 3-D reality to a Multi-user Dimension (MUD)-type creation of imagined

worlds and avatars in those worlds. However, VRML can also be used in a more mundane commercialization of the technology, such as entering a simulation which enables the user to operate a product such as a new model of automobile or to enter a kitchen in an apartment complex that is still under construction.

A good VRML repository can be found at the San Diego Super Computing Center at <http://www.sdsc.edu/vrml/>. At this web site, links and applications of third generation network interactions in various disciplines are available. Another informative site is <http://hiwaay.net/~crispen/vrml/>.

Java versus ActiveX

Netscape's JavaScript and other alternatives like VRML have specialized and limited distributed network computing power. Two of the main executable content power user alternatives in browsers are Java and ActiveX. Sun Microsystems's Java was introduced as an easier way to write C++ code, but Java has since evolved to be much more than that in the third generation of real time distributed computing across networks. Java is a portable, downloadable embedded "applet" code that can be easily integrated into browsers. Both the Netscape Navigator and Microsoft Internet Explorer browsers integrated Java into their latest upgrades. Any Java class or applet that is sent down to the browser can be executed interactively. Java is also the basis of new network computers (NC) that rely upon servers for operating systems and software that in other computers are normally kept on resident hard drives. In 1997, IBM Corporation's Taligent^{72W} subsidiary became the world leader in Java development.

Microsoft Corporation's ActiveX, on a somewhat different tack, competes head to head with Java. Microsoft takes the wealth of OLE^{73W} objects that are in the market and uses ActiveX to provide them on the web. For example, it is possible to transmit spreadsheets across the web using ActiveX. ActiveX is based upon a form of registration that lends it more security than can

be obtained with Java applications in the third generation. Before clients download an OLE control they can identify where that control is actually coming from. One drawback is that only the Internet Explorer browser is enabled to execute ActiveX controls.

Another emerging technology (rooted in Java, ActiveX, VBScript, or Java Script) is the technology of enabled client views or "view classes" of databases at remote sites. These view classes also provide database functions (depending upon security controls) on the client side (e.g., the ability to make edits, queries, logins, forms, reports, etc.). Remote clients can integrate their view classes and applets on top of "data-enabled frameworks." This means that users can have simultaneous views of the data locally on network client machines even though they are viewing the same databases.

Open Database Connectivity (ODBC)

Open Database Connectivity (ODBC)^{74W} refers to a standard for accessing different database systems in Visual Basic and Visual C++. Applications in most any software (e.g., Asymetrix ToolBook)^{75W} can submit statements to ODBC using the ODBC type of SQL. ODBC then translates the code for use in common database systems such as Microsoft's Access^{76W}, Borland's Paradox^{77W}, Borland's dBase^{78W}, Microsoft's Excel^{79W} and Btrieve Technology, Inc.^{80W} databases.

ODBC is based on Call-Level Interface and was defined by the SQL Access Group. Microsoft was one member of the group and was the first company to release a commercial product based on its work (under Microsoft Windows) but ODBC is not a Microsoft generated standard. ODBC drivers and development tools are available now for Microsoft Windows, UNIX, OS/2, and Macintosh. These were developed for local computing and are not designed especially for networking.


ODBC is opening up the Internet for database connectivity, however. Popular HTML

editors such as Microsoft FrontPage and SoftQuad HotMetal Pro now have utilities for SQL. Hypermedia web authoring systems like Asymetrix ToolBook and Macromedia Authorware can similarly generate SQL for ODBC.

Emerging Distributed Network Computing Standards: IIOP, RMI, and DCOM

Development tools for data enabled frameworks are starting to emerge, and new standards such as CORBA's IIOP, Sun's RMI^{81W}, and Microsoft's DCOM are appearing. RMI is part of the Java programming language library which enables a Java program running on one computer to access the objects and methods of another Java program running on a different computer. Some vendors are also having their own proprietary extensions as well. IIOP^{82W} is built upon CORBA^{83W} technology. DCOM^{84W} is Microsoft Corporation's standard for distributed network computing.

A good reference for more extensive distributed network computing terminology and links to technical details can be found in *World Wide Web and Object Technology* by Ashish B. Shah at

<http://ei.cs.vt.edu/~wwwbtb/fall.96/book/chap19/index.html>. 

Footnotes

Footnote links to web sites are given at:
<http://WWW.Trinity.edu/~rjensen/260notes.htm>