

Building a Flexible Foundation for e-Business

By Tony Bove

To support growing numbers of customers and transaction volumes, an e-business needs a reliable and scalable infrastructure. This article focuses on the infrastructure provided by BEA WebLogic Server, a core product in the BEA WebLogic E-Business Platform.

e-business plans must be as fluid as the swiftly changing environments in which they operate. Software, systems, and the entire architecture supporting an e-business must naturally mold and adapt to changes without costly, time-consuming infrastructure overhauls. To support growth and flexibility, an e-business infrastructure needs to be reliable, scalable, and based on open standards to ensure interoperability with future applications.

BEA WebLogic® Server, a core product in the BEA WebLogic E-Business Platform™, provides a foundation for effective e-business. In addition, because it is an adaptive software platform, it can become a powerful catalyst for innovation. As the Java™ application server with the leading market share worldwide, BEA WebLogic Server offers comprehensive support for Enterprise JavaBeans™ (EJB) technology and a wide spectrum of Web and wireless e-commerce applications. Component-based application development, deployment, and management, combined with adherence to industry standards, provide flexibility.

In order to create applications that adapt quickly to changing business conditions, business logic must be separate from the underlying transaction processing logic. BEA WebLogic Server code may simply define transaction attributes and use the transaction services already built into the infrastructure. This allows business logic to be modified without introducing errors into the transaction processing logic.

Application servers for business logic

BEA WebLogic Server is an application server that offers server-side support for developing and deploying business logic.

Several benefits of running processes on application servers are described below.

Scalability and reliability. Application servers can be clustered to distribute services across multiple server installations, creating a stable foundation for applications across an enterprise or multiple networked enterprises. Clusters can scale to include hundreds or even thousands of cooperative servers, which may be accessed by tens of thousands of clients. Clients do not “see” clusters any differently than individual servers, but clusters provide greater scalability and reliability.

Simplified management. Server-side applications are easier to monitor, control, and update. Sharing business logic across a variety of client applications—Web browsers, wireless clients, Java applets, and Microsoft® COM+ components—provides flexibility. The integrated management provided by application servers enables comprehensive views of application resources, network resources, system resources, and diagnostic information. Deeper and wider management functions allow integrated, unified management of both stand-alone servers and distributed server clusters.

Security. Business logic often manages or otherwise involves sensitive, private information. Storing business logic on the server allows user access to be controlled dynamically and revoked at any time. Application servers use standard secure protocols, such as Secure Sockets Layer (SSL) or HTTP Secure (HTTPS), instead of proprietary database protocols, which may be less secure.

Network efficiency and performance. Storing database-intensive business logic closer to the database may save network

traffic and reduce access latency, enhancing network performance. Similarly, running CPU-intensive applications on servers saves processing cycles on thin clients such as Personal Digital Assistants (PDAs), wireless devices, and low-memory PCs.

Java application servers

Java application servers allow component-oriented development. Java developers can assemble applications from reusable JavaBeans, rather than repeatedly re-engineering applications from scratch. For example, the EJB specification includes provisions for container-managed transactions. Once the attributes of a container-managed transaction are defined, the EJB container automatically starts and commits transactions as requested. High-level representations—such as customer records, accounts, and transactions—may be passed as objects among several components.

Java application servers offer compelling benefits that other types of application servers may not. A simpler, higher level alternative to C++, Java provides the following benefits:

- ▶▶ Deployment on clustered servers supports scaling.
- ▶▶ Useful classes make it easier to extend the language with high-level, reusable business abstractions.
- ▶▶ Authoring tools for Java code provide enhanced robustness, quick testing, and simplified debugging.

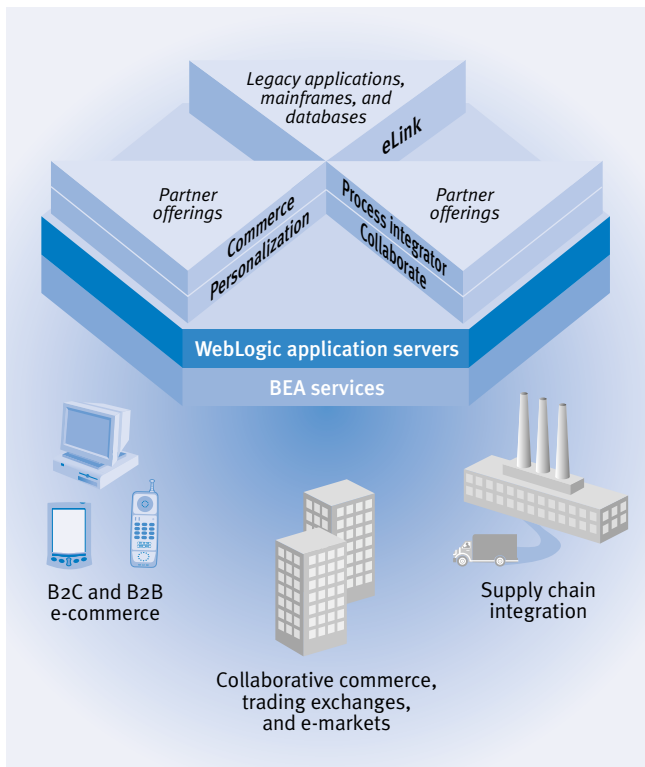


Figure 1. BEA WebLogic Server and E-Business Platform

- ▶▶ Compatibility with any operating system platform, the Internet, intranets, and extranets provides flexibility.

BEA WebLogic Server complies with Java 2 Enterprise Edition (J2EE) standards and supports EJB 2.0, which offers improved messaging and database integration.

BEA WebLogic Server capabilities

BEA WebLogic Server 6.0 combines the capabilities of a Java-based application server, Web server, messaging server, transaction server, and integration server to ease integration challenges and provide a foundation for the BEA WebLogic E-Business Platform. Figure 1 shows the architecture of the BEA WebLogic Server and E-Business Platform.

BEA WebLogic Server provides essential features for developing and deploying mission-critical e-business applications:

- ▶▶ **Certified standards support:** Comprehensive J2EE support, including EJB 2.0, servlets, JavaServer™ Pages (JSP), Java Messaging Service (JMS), Java Naming and Directory Interface (JNDI), Java Management Extensions (JMX), Java Database Connectivity (JDBC), Java Cryptographic Architecture (JCA), and Remote Method Invocation (RMI); enhanced XML support
- ▶▶ **Enterprise performance and scalability:** Software clustering of dynamic Web pages (servlets and JSPs) and EJB components; client connection sharing and database resource pooling
- ▶▶ **Client options:** Support for programmatic, browser, and wireless clients
- ▶▶ **Administration:** Comprehensive, Web-based administration console for configuring and monitoring WebLogic Server deployments; integration with leading management frameworks and products
- ▶▶ **Security:** SSL, Access Control Lists (ACLs), relational database management system (RDBMS), UNIX®, and custom security realms for integration of encryption and authentication services into e-commerce solutions
- ▶▶ **Development and deployment flexibility:** Tight integration with leading databases, operating systems, Web servers, and development tools

How WebLogic works

BEA WebLogic Server listens on the network for client requests. Clients specify a URL, which provides the network protocol and location of the desired BEA WebLogic Server resource. Once the server establishes a connection with a client and negotiates details such as protocol, encryption, and authentication,

BEA WEBLOGIC SERVER HARDWARE SIZING

BEA WebLogic Server operates with standard operating systems and hardware. The table outlines the Dell hardware configurations recommended for various types of installations. Since platform certification is an ongoing process, visit <http://e-docs.bea.com/wls/platforms/index.html> for the latest certified platforms.

	Installation		
	Small	Medium	Large
Load (transactions per second)	Up to 100	Up to 200	Up to 300
Apache Web Server	PowerApp.web 120, number of processors depends on content	PowerApp.web 120, number of processors depends on content	PowerApp.web 120, number of processors depends on content
BEA WebLogic Application Server	PowerEdge 2450, two processors, 2 GB of memory	PowerEdge 6450, two Xeon processors, 2 GB of memory	PowerEdge 6450, two Xeon processors, 2 GB of memory
Oracle Database Server	PowerEdge 6450, at least four processors	PowerEdge 6450, at least four processors	PowerEdge 6450, at least four processors
Storage	PowerVault 210	PowerVault 210	PowerVault 210

it processes the client's requests by executing Java classes on the client's behalf.

On the front end, a BEA WebLogic Server may act as the primary Web server or process requests redirected from other Web servers, such as Netscape Enterprise Server®, Microsoft Internet Information Server (IIS), or Apache (see sidebar). Alternatively, other Web servers may handle requests for static HTML Web pages and pass servlet and JSP page requests to a BEA WebLogic Server.

In the middle tier, BEA WebLogic Server hosts business logic in EJB containers. EJB containers allow developers to encapsulate business logic without negotiating complex details of transaction management, security, or naming and directory services.

On the back end, BEA WebLogic Server can connect to a wide range of network-accessible services such as relational databases, naming and directory services, messaging systems, and legacy applications. The EJB architecture also simplifies the connection between the application and database tiers. The advanced container-management persistence features of EJB 2.0 allow the development of portable applications that are database-independent and free of database access code.

Distributed transactions

Some transactions involve coordinating several systems. A single transaction may involve checking inventory, validating a credit card, and confirming delivery availability. Distributed transactions across the Internet are typically conducted asynchronously

rather than in real time. Asynchronous transactions allow the initiator to begin the transaction and then tend to other processes as the transaction proceeds, without regard to time passed, network outages, or other factors. This helps to ensure that the transaction is completed, despite possible delays and disruptions on the intervening systems.

Data integrity requires all the information on both sides of a business process to be updated at the same time. Transaction and application servers must coordinate complex transactions across multiple systems, guaranteeing transactional integrity in an environment where multiple users and systems may access resources simultaneously.

BEA WebLogic Server's distributed transaction manager oversees such complex transactions, allowing developers to create Java-based applications to perform transaction chains distributed across multiple systems. BEA WebLogic Server centrally coordinates the various systems involved, including databases, mainframes, messaging systems, and other applications.

Asynchronous messaging

Messaging provides an effective means of transmitting events between applications. Asynchronous messaging allows for the development of loosely connected applications, which are typically more resilient in the event of failure and more easily extensible as new applications are developed. The integration of JMS and EJB allows such loosely

BEA WebLogic Server can connect to a wide range of network-accessible services, such as relational databases, naming and directory services, messaging systems, and legacy applications.

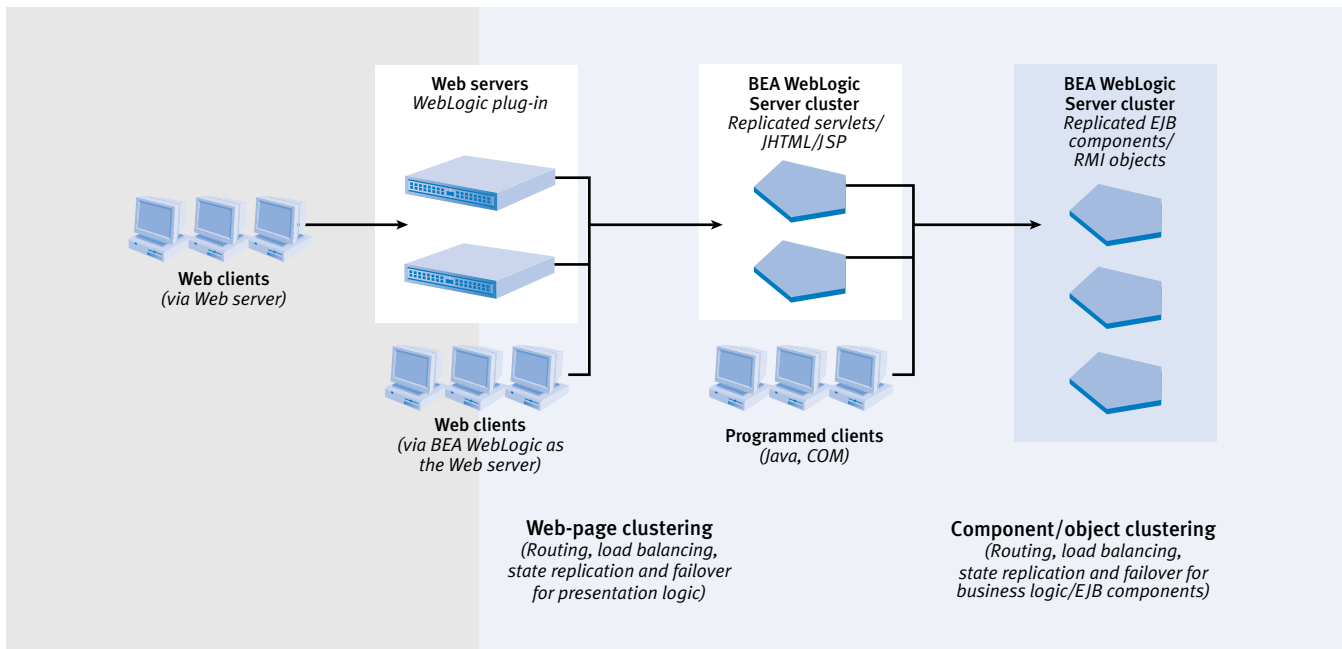


Figure 2. WebLogic Server cluster

connected systems to make full use of JavaBeans. A JavaBean can send asynchronous messages via the JMS application programming interface (API).

Furthermore, message-driven JavaBeans can receive and act upon JMS messages without involving an application client user interface. JMS allows database, file, or in-memory persistence; this flexibility gives developers the choice between publish/subscribe and point-to-point messaging. WebLogic JMS also supports either multicasting or TCP/IP transport protocols. JMS can directly access the EJB and transaction services needed to leverage a message-based architecture from a single messaging foundation.

These capabilities make EJB well suited to the development of shared business services and allow asynchronous data processing applications to be built on the J2EE platform—opening a realm of possibilities once restricted primarily to mainframes.

Server clusters provide reliable infrastructure


A WebLogic Server cluster is a group of WebLogic Servers that work together to provide a more powerful, reliable, and scalable application platform than a single server could provide (see Figure 2).

A WebLogic cluster uses the redundancy of multiple servers to insulate clients from failures—if a server fails, its processes roll over to another server. WebLogic clusters bring scalability and high availability to J2EE applications, while remaining transparent to application developers.

BEA WebLogic Server provides support for load balancing and allows applications to switch to another database system for

distributed processing or in failover situations. Replicating EJB sessions in memory further protects business logic.

Server-side code also benefits from load balancing and failover support. Most J2EE application code runs in the middle tier and can use services distributed among several WebLogic Servers. For example, a servlet running on Server A could use a JavaBean on Server B and read messages from a JMS queue on Server C.

BEA WebLogic Server provides a reliable and flexible infrastructure that allows companies to leverage their investments in existing systems while realizing the operational efficiencies and cost-savings that e-business provides. 

Tony Bove (tbove@bea.com) is director of customer messaging at BEA Systems, Inc. He has been a writer, analyst, and journalist in the computer industry since 1976. He founded *Publish* magazine and the newsletters “*Inside Report on New Media*” and the “*Macromedia User Journal*.” Tony has also written more than a dozen computer-related books including the best-selling *The Art of Desktop Publishing*.

FOR MORE INFORMATION

For more information, service, and support, please visit www.bea.com or call 1-800-817-4232 or 1-408-570-8000