



# Maximizing E-business Application Performance

Using Sitraka JProbe and BEA WebLogic

October 2002

A Sitraka White Paper



## Contents

---

<b>Management Summary .....</b>	<b>3</b>
<b>BEA WebLogic Server .....</b>	<b>3</b>
<b>Scalability .....</b>	<b>4</b>
<b>Reliability .....</b>	<b>4</b>
<b>Sitraka JProbe .....</b>	<b>5</b>
<b>Using Sitraka JProbe and BEA WebLogic Server Together .....</b>	<b>6</b>
<b>Complementing JProbe – Sitraka PerformaSure .....</b>	<b>7</b>
<b>Conclusion .....</b>	<b>7</b>

Copyright© 2002 Sitraka Inc.

Note: Sitraka and JProbe are trademarks of Sitraka Inc. BEA and BEA WebLogic Server are trademarks or registered trademarks of BEA Systems Inc. Sun, Sun Microsystems, the Sun logo and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other brand names are registered trademarks or trademarks of their respective holders.

## **Management Summary**

---

For an e-business application to be successful, it must perform so well that every user finds it both responsive and always available. The application must be able to handle significant increases in user load; it must also be available to users wherever and whenever they need it—around the world and around the clock.

BEA WebLogic Server is one of the leading application servers on the market today. Sitraka JProbe is the premier performance tuning toolkit that helps developers diagnose errors and inefficiencies in Java application, servlet, JSP and EJB code. By using JProbe and BEA WebLogic Server together, organizations can implement a strategy that enables them to meet the performance, scalability and reliability requirements that today's users demand.

## **BEA WebLogic Server**

---

BEA WebLogic Server powers some of the most demanding Web and Java 2 Enterprise Edition (J2EE) applications in production today. With its advanced architecture and support for J2EE standards, BEA WebLogic Server is a mission-critical component of many e-business applications. BEA WebLogic Server supports advanced features such as scalable clustering for core J2EE services like Enterprise Java Beans (EJB), Java Servlets, JavaServer Pages (JSP) and Java Messaging Services (JMS) and includes enhanced support for technologies like Extensible Markup Language (XML).

J2EE standards represent the most significant advance in enterprise software development since the relational database. With J2EE, companies can develop complex, distributed enterprise applications independently of the underlying computer hardware, operating system or database. This allows enterprises to develop applications using a single, cross-platform technology while still utilizing their existing data management and business systems. Companies adopting J2EE benefit from shorter time to market and lower development and administrative costs. J2EE delivers unprecedented developer productivity through the Java programming language, the reusability of EJB components and standardization across platforms—all of which enable developers to focus on writing business logic and addressing business-critical issues.

## Scalability

---

Ensuring high performance is critical to the success of Web-based applications. As a user base grows from dozens to hundreds to thousands of simultaneous users, the capabilities of an application server need to grow as well. Scalability can be defined as an application's ability to maintain constant user response times regardless of how many users are on the system simultaneously. Users accessing applications over the Web don't see the other thousands of users accessing it. They each see themselves as unique—the only user in the whole world using the application. Maintaining that illusion for thousands of customers, around the world and around the clock, is a challenge.

BEA WebLogic Server has advanced facilities for seamless scalability, expandability and high availability via its clustering functionality. BEA WebLogic Server provides both Web page and EJB-component clustering—without the need for special-purpose hardware or operating system services. Web page clustering handles transparent replication and load balancing of the presentation logic providing responses to Web clients. EJB component clustering handles replicating, load balancing and failover for EJBs (components that encapsulate business logic). Both Web page and EJB clustering are critical for delivering global scalability and high availability for e-business systems.

Application server clustering is only part of the solution for ensuring scalability. While the application server takes care of many important parts of e-business systems, numerous other components exist as well: databases, dedicated Web servers and, of course, the e-business application itself. Conflicts and misconfigurations can cause poor performance and are very often difficult to diagnose. Even more importantly, bottlenecks within the application will lead to poor scalability. In situations where an application bottleneck is jeopardizing scalability, simply adding more CPUs to the BEA WebLogic Server cluster isn't going to help. The application itself imposes a fundamental limit on scalability. Additionally, clustering is an expensive way to scale e-business applications. Eliminating application bottlenecks can increase a server's ability to support simultaneous users by 10% to 100%. Eliminating application bottlenecks is where JProbe can help.

## Reliability

---

*Reliability* is another element of performance critical to successful e-business applications. Reliability can be defined as a Web site's ability to be constantly reachable, around the clock and around the world, regardless of how many users are accessing it. Users in general have grown accustomed to highly available Web services. Their expectations have been set by systems engineered for high reliability, such as the telephone network, which has better than 99.9% uptime. Reliability is just as important as scalability because users who are unable to access systems are likely to look elsewhere for the information they

need. In a retail sales e-business, this means lost sales. For an enterprise information system, this means user downtime resulting in significant costs.

BEA WebLogic Server can be used to create high availability systems through its advanced clustering features. In addition to increasing scalability through clustering and load balancing, clustering increases reliability via failover. Failover means that when one node in a BEA WebLogic Server goes down due to an error or a hardware failure, user sessions are redirected to another node in the cluster.

Failover is an excellent way to insulate systems from hardware failures, but what if there's a software failure? All the nodes in a cluster run the same code. If a software error brings one node down, it will likely bring the entire cluster down. The only way to improve the reliability of a software application is by rigorous testing and analysis with tools that help developers see exactly what's going on inside their programs. Problems like memory leaks and loitering objects, which can lead to "out of memory" errors over time, lead to system crashes. Threading issues such as deadlock or race conditions can make an application server appear to have crashed when the code has simply stalled while waiting for a required resource. JProbe Memory Debugger and JProbe Threadalyzer help solve these problems, increasing reliability.

## Sitraka JProbe

---

JProbe is a comprehensive, integrated toolkit for diagnosing and eliminating code errors and inefficiencies in Java applications. JProbe products help e-business application developers understand precisely what's causing problems in an application, right down to the offending line of source code. JProbe paints graphical pictures of everything from memory usage to calling relationships, letting developers navigate their way to the roots of problems quickly and easily.

JProbe can also analyze interactions with Java modules outside the application being analyzed, even if source code for those modules is unavailable. JProbe identifies problems caused by interactions with third-party components, queries to a relational database via Java database connectivity (JDBC) and calls to other distributed elements of the application through remote method invocation (RMI). While it isn't possible to modify parts of an application without source code, understanding where performance problems lie is essential to solving the problem.

JProbe's ability to gather information about applications with or without source code is one of the many advanced capabilities that make it easy to use in any situation. For example, if you discovered that a particular call to a third-party Java module is a key

bottleneck, you might want to implement caching algorithms to possibly decrease the overhead time spent in these calls.

The JProbe product line consists of four different tools to help e-business application developers: JProbe Profiler, JProbe Memory Debugger, JProbe Threadalyzer and JProbe Coverage. JProbe Profiler identifies where time is being spent inside an application, helping developers increase the efficiency and scalability of their code. JProbe Memory Debugger pinpoints memory leaks and memory mismanagement, both of which can seriously hinder scalability. JProbe Threadalyzer tracks down threading issues such as deadlock and data race situations. JProbe Coverage ensures complete test cases by identifying which parts of an application have and have not been executed during testing.

The JProbe line of products, when used as part of the development process, can ensure that critical performance, scalability and reliability issues are identified and addressed. Sitraka application engineers recommend that developers follow a step-by-step approach to performance tuning, concentrating on four areas of testing in the following order: thread analysis, memory debugging, profiling and code coverage.

## **Using Sitraka JProbe and BEA WebLogic Server Together**

---

To analyze code with any of the JProbe Suite tools running inside BEA WebLogic Server, all developers need to do is start the JProbe Application Server Integration tool, select the version of BEA WebLogic Server, configure the details of the BEA WebLogic Server location, and specify a JProbe settings (JPL) file to use. JProbe can then begin collecting the data necessary to provide performance intelligence on an application.

Prior to profiling, developers should decide whether they would like to profile for memory heap data, performance data or both at the same time. If performance data is the choice, for example, a good strategy is to deselect the ‘Record Performance from Program Start’ option. This way JProbe won’t collect data on the BEA WebLogic Server classes while starting up but will instead focus on the application itself. Once the settings are configured, developers can begin to manually collect performance data using the JProbe control panel.

Developers can also use performance triggers and filters to customize their performance testing. Developers can set a performance trigger in JProbe that automatically starts data collection when a certain event occurs—for example, when one of their EJBs or servlets is first invoked. With filters, developer can see the big performance picture by profiling both their application code as well as the code within BEA WebLogic Server. Alternatively, they can filter out the BEA packages and classes to get performance data specific to their application. For example, developers can filter out any third-party class libraries, or, to be even more aggressive, filter out all classes (‘Ignore Method Data’ for classes ‘\*’) and then

filter their own code back in ('Collect Line Data' for 'mypackage.\*'). By using filters effectively, developers will reduce the time required to collect the performance data relating to an application and, more importantly, reduce the scope of analysis to the critical areas of investigation.

The most up-to-date information on using JProbe and BEA WebLogic Server together is available in the JProbe Application Server Integration Portal for J2EE, at [www.sitraka.com/jprobe/j2ee](http://www.sitraka.com/jprobe/j2ee). Support for BEA WebLogic Server versions 5.1.0, 6.0, 6.1 and 7.0 are all included with JProbe 4.0 running on Windows, Solaris, AIX, Linux and HP-UX. (Note: not all versions of BEA WebLogic Server are supported across all operating systems.)

## **Complementing JProbe – Sitraka PerformaSure**

---

Also available from Sitraka, Sitraka PerformaSure complements JProbe by providing high-level transaction-centric diagnosis of an entire distributed J2EE application. Together, JProbe and PerformaSure form an integrated J2EE Performance Assurance solution for BEA WebLogic, enabling performance teams to track down J2EE performance and scalability problems wherever the root cause.

## **Conclusion**

---

Sitraka JProbe helps companies using BEA WebLogic Server achieve higher levels of performance, scalability and reliability in their e-business applications. Because JProbe integrates seamlessly with BEA WebLogic Server, developers can easily understand and solve any and all performance issues related to the scalability of applications before they affect the end-user experience. By using JProbe and BEA WebLogic Server together, developers can implement a strategy that ensures they meet performance requirements. Users may not recognize maximum performance in an e-business application, but they will certainly notice when performance is *not* up to their high expectations.

Additional information and free evaluations of JProbe products are available on the Sitraka Web site at [www.sitraka.com/jprobe](http://www.sitraka.com/jprobe). See why BEA developers and consultants use JProbe as a key part of their performance tuning process.

### **About Sitraka**

A proven leader in J2EE Performance Assurance, Sitraka delivers advanced diagnostic solutions that help companies to pinpoint and eliminate performance hazards in mission-critical J2EE applications. Sitraka products include PerformaSure, a transaction-centric

J2EE diagnosis solution, JProbe performance tuning tools, JClass Java components, and DeployDirector, a Java application provisioning and management solution. Through industry partnerships with companies such as BEA Systems, IBM, Sun Microsystems and Mercury Interactive, Sitraka ensures that our products integrate seamlessly with the latest development environments and platforms. Sitraka products are sold and supported directly through our North American headquarters in Toronto, European headquarters in Amsterdam and global network of resellers. Visit Sitraka on the Web at [www.sitraka.com](http://www.sitraka.com).