Maximizing business value by optimizing J2EE™ performance

Is your enterprise ready for high-performance SOA?

A Borland White Paper

June 2004

Abstract

Enterprises can gain more value from their J2EE™ infrastructure by adopting Borland performance management processes and tools throughout the Application Lifecycle, achieving lower costs, improved application responsiveness and optimized performance, while speeding development cycles with enhanced IT productivity. Implementing Borland methodologies and tools can deliver a competitive edge and the confidence to run a global business using the emerging Service Oriented Architecture (SOA).

Borland®

Executive Summary: The business value of J2EE performance

The importance of optimizing mission-critical business applications to achieve maximum performance and availability has been learnt the hard way. When applications go down or do not perform, the business cost of downtime can run into millions due to both missed revenue opportunities as well as the frequently complex and manual support processes that many organizations have in place.

This is especially true for applications built on the J2EE platform. The very nature of its tiered architecture makes application performance problems hard to detect and time consuming to resolve. With increasing pressure on IT organizations to be more responsive to business requirements while containing application development and support costs, the need to build cost-effective quality and performance into the application proactively from the outset has never been greater.

With the adoption of Service Oriented Architecture (SOA) to achieve a loosely coupled technology model that mirrors global business processes, the role of J2EE has established itself as one of the most prevalent standards for implementing SOA. The potential for performance issues is heightened due to the vast number of SOA dependencies that can impact the overall system performance, raising the need for increased visibility to track and trace SOA-based application calls across the entire J2EE environment.

To achieve maximum business value from these J2EE-based technology investments, the IT organization needs to be equipped with smarter tools for managing J2EE performance. Borland has an unparalleled reputation for providing IT organizations with proven performance optimization solutions that solve critical technology implementation issues throughout the application lifecycle. With Borland Optimizeit ServerTrace, all teams in the application lifecycle management process can benefit by obtaining the right breadth and depth of performance analysis information appropriate to help them carry out their role in accelerating problem resolution. As a result of this increased collaboration, IT organizations can realize the following key benefits:

- Time to market of new applications is reduced through higher development productivity
- Cost savings are realized through better utilization of existing J2EE tools and infrastructure
- New development trends (such as SOA) can be used to create differentiable competitive advantage
- Business threatening downtime of critical J2EE applications is dramatically reduced

Introduction

The successful enterprise IT organization is challenged to meet a number of critical business goals: deliver increasingly complex mission-critical business applications quickly and securely, and ensure that core business operations are strongly supported. Availability is critical as poorly performing applications have an immediate business impact, cutting revenue and alienating customers. These challenges must be met within the constraints of tight IT budgets and scarce internal corporate resources, while the external environment breeds new technology developments that determine how the enterprise derives competitive advantage.

The IT executive must meet these challenges head-on, contending with an environment where businesses are more interconnected and network-enabled than ever before, and information system capabilities will determine the success and failure of core business capabilities. **Today's technology executives must manage risk and ensure that their IT initiatives and investments deliver**:

- The **high availability** and **performance** required to meet the needs of the global business environment, ensuring that transactions are processed with utmost reliability and security
- **Collaboration** between disparate and geographically dispersed development teams around the globe
- The **foresight** and **capability** to support emerging technology initiatives such as service-oriented architectures (SOA)

Most enterprises have implemented business functionality as a framework of distributed multi-tier applications; commonly built on the foundation of the Java™ 2 Platform, Enterprise Edition (J2EE™) architecture. Over 85% of Fortune 500 companies[1] have utilized J2EE to integrate existing information systems with new services, and tie into maturing Web technologies to bridge end users to core business processes. The scalability and performance of J2EE applications has led to the platform becoming the de facto standard for server-side development, while the standards-based framework and vendor-independence has become the answer to IT leaders looking to reduce costs and manage complexity.

[1] Source, Sun Microsystems

Many IT executives are reacting to this trend towards the distributed J2EE application development model by changing how they participate in the Application Lifecycle Management (ALM) process. Issues with application performance and developer productivity in the ALM process indicate that appropriate tools are required to address concerns early in the development life cycle, before critical and expensive faults threaten business operations.

This White Paper investigates the challenge of current and future issues facing the IT executive who has committed to the J2EE platform for delivering key mission-critical applications. The synergy of effective processes and products used throughout the application lifecycle will be interpreted to show how IT leaders can take control and establish measurable benchmarks – cost containment, speed of application development, productivity improvements, and performance enhancements – while ensuring that the greatest potential is being realized from the J2EE platform infrastructure. Specifically, the benefits of Borland® Optimizeit™ ServerTrace will be illustrated to show its use as a critical tool that all members of the IT organization may use to speed diagnosis and resolution of J2EE performance issues.

Managing J2EE performance is a critical business need

The failure of business-critical applications result in significant costs to the enterprise, including lost customers, damaged partner and supplier relationships, and lost productivity from the IT organization. With J2EE implementations, complexity only compounds the potential cost and scale of performance problems. As J2EE adoption grows, complexity increases and managing performance issues becomes even more challenging:

- Obtaining granular information about the root-cause of performance issues becomes more difficult to isolate and pinpoint
- Reporting and sharing of performance metrics becomes extremely challenging and demands discipline from the IT organization, as the code base is broadly distributed
- Finger-pointing becomes endemic, as delays in determining the problem source may cause functional groups to become defensive, attempting to prematurely assign blame elsewhere
- Performance problems may only become apparent late in the development cycle and in the production environment, where diagnosis becomes more difficult and costly, and the implications more severe

It quickly becomes apparent that ad-hoc solutions cannot be applied if it is unclear where performance problems are occurring in the J2EE environment. For the IT executive, it becomes clear that a well-designed development methodology backed with solid performance management tools deliver maximum performance in the most cost-effective manner.

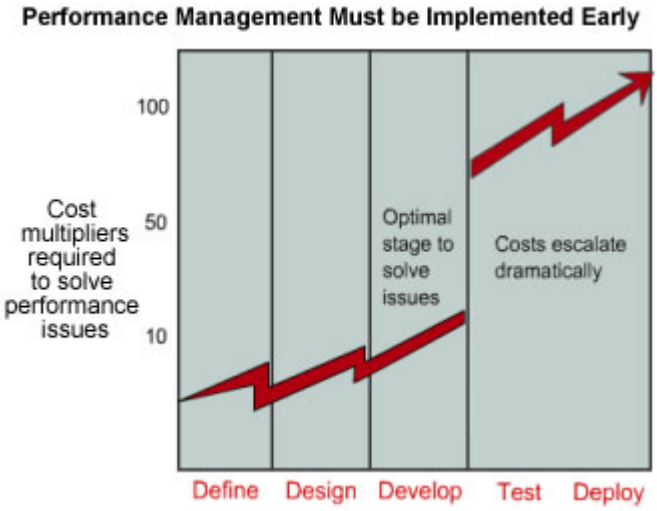

Figure 1: The cost of resolving performance issues increases greatly in later stages of the application lifecycle

Meeting IT Goals: Optimizing Business Technology

Unifying the Distributed Enterprise: ALM methodology serves the IT organization

Ideally, business applications are developed through a repeated cycle of steps from requirement identification to end user adoption: definition, design, development, testing, and deployment. Planning for success around these phases is known as the application lifecycle management (ALM) process, which provides structure and an increased probability of success in application deployment.

When problems emerge with a live business application based on the J2EE platform, IT groups often default to a reactive mode. IT architects, developers, and QA teams rally in an attempt to identify the root cause of the problem, often without sufficient data to capture the cause and severity of the issue.

Problems in J2EE applications are compounded by the complexity of the distributed environment: global enterprises rely on heterogeneous IT systems to link to customer, supplier, and partner systems. This complexity makes it challenging to diagnose the root cause of performance issues, which could exist deep within the J2EE infrastructure: a simple server configuration, untested application, overtaxed database or any number of other issues could be contributing to poor application performance.

The result is that without an ALM methodology, release and repair cycles are often long and complex, while ownership responsibility for performance issues often remains unclear. To resolve this, performance concerns must be integrated into all stages of the application lifecycle as part of the ALM process:

- Developers must write code reflecting J2EE standardized services, while utilizing tools that can quickly and efficiently detect system-level performance issues on local workstations
- Test engineers must undertake full functional testing and load testing of the J2EE system in a pre-production environment, to model real-life use cases
- The production environment must be capable of being monitored and analyzed, in the event that coding errors not caught earlier in the application lifecycle can be diagnosed and rectified quickly.
- Tools must encourage communication and information sharing between functional teams, so that problem diagnosis can rapidly lead to resolution

Building quality into the ALM process ensures that the IT organization can stand behind the delivery of applications and services with assurance. For mission-critical applications, it is also necessary to provide tools to monitor and analyze the production environments so that when problems arise, the root cause can be quickly determined and delegated back to the responsible area for quick resolution.

Borland provides a comprehensive set of tools to ensure quality and performance are integrated into the ALM process. Borland Optimizeit solutions include:

- **Borland® Optimizeit™ Enterprise Suite**, an award-winning set of tools used during the development stages of the application lifecycle, providing individual developers with a focused view into performance issues in their code

- **Optimizeit™ ServerTrace**, which focuses on J2EE performance metrics in the testing and deployment phases of the application lifecycle, providing monitoring and analysis of the entire distributed environment

Optimizeit ServerTrace extends its value throughout the entire IT organization by monitoring production and pre-production J2EE environments, providing a view into the behavior of the entire distributed system, and allowing for quantifiable and measurable analysis of performance issues. Optimizeit ServerTrace fits seamlessly into the J2EE infrastructure by providing tight integration with industry-standard application servers and ALM tools, such as IDEs, load testers, test frameworks and production monitoring frameworks that are already in use by the IT organization.

Optimizeit ServerTrace also provides a reporting framework that allows test teams, software developers, operations groups, business process owners, and any parties involved in the ALM process to quickly share information pertaining to performance issues, whether the team members are in the next cubicle or halfway around the world. This allows users to easily generate rich reporting on demand that is tailored to the needs of their peers and managers, enabling quick isolation and resolution of performance problems. Reporting allows users to immediately assign the responsibility for countering paralyzing performance issues to the appropriate owner, effectively reducing organizational friction and finger pointing, and decreasing time spent to resolution. **The result: time to resolution and costs are reduced, and IT organization productivity increases dramatically.**

Preparing for change: getting ready for Service Oriented Architecture (SOA)

Most IT organizations are looking at the adoption of service-oriented architecture (SOA), moving away from application-centric development processes to a new methodology that offers advances in software reuse, flexibility, and connectivity to customers, partners, and suppliers.

An SOA specifies how entities interact, allowing one entity to perform a service on behalf of the other. Each interaction is self-contained and loosely coupled, so that each interaction is independent of any other interaction. Interactions are also relatively inexpensive to implement, and alternative services may be quickly established or replaced.

SOA is revolutionary in how services can map to business processes, without requiring the business users' deep knowledge of the underlying IT infrastructure. This alignment is unique for process owners in the organization; for the first time, they may use a technology model that mirrors business processes. The IT organization and business process owners can then focus on relationships and logic, while speaking the same SOA language. Defined on open standards, and with most technology vendors and partners lining up to back the architecture, SOA adoption is extending quickly across the landscape.

J2EE has emerged as one of the most prevalent standards for the implementation of SOA. As Web Services spans the distributed computing environment, extending to partners, customers and suppliers, the increases in complexity introduce the potential for performance issues on both sides of the corporate firewall. For example, object request bottlenecks that may appear in the SOA implementation of a supplier's inventory database may have repercussions for the downstream manufacturer, through a shared Web Service. Maintaining acceptable response times and availability are critical for the organization utilizing an SOA, and may be enforced through the use of service level agreements (SLAs) between all users. The key to enforcement of SLAs may be difficult, as the distributed service-oriented model makes it difficult to measure and pinpoint performance bottlenecks in the end-to-end SOA systems. However, the enterprise can eliminate the performance bottlenecks in their SOA applications through the use of J2EE performance tools. Remote SLA enforcement capabilities may be implemented in J2EE performance monitoring frameworks, alerting process owners when remote service providers are not meeting their SLA targets. .

Borland Optimizeit ServerTrace offers the ability to track and trace SOA-based application calls across the entire J2EE environment, giving detail on system performance within the corporate IT environment, and details on service calls outside the firewall. This gives the IT organization visibility over the vast number of SOA dependencies that can impact overall SOA system performance, and the ability to measure and enforce SLAs. This ability to view across the distributed environment with Optimizeit ServerTrace gives an unparalleled perspective on the system performance: the user can drill down on the service calls offered across the distributed system, quickly identify expensive and time-consuming resources and bottlenecks, and share the results with the responsible owners (developers, business analysts, QA teams, and/or architects) through deep diagnostics reporting.

Meeting enterprise goals: ALM tools deliver necessary application performance

With the enterprise opening up core processes to customers, partners, and suppliers through the delivery of business applications, competitive advantage can be derived by offering applications optimized for performance and deployed in quick release cycles. Users will embrace applications that are scalable, reliable and secure, while under-performing applications will be plagued with user dissatisfaction and lack of acceptance. As such, it is necessary to tune the J2EE infrastructure to ensure that competitive advantage may be realized.

Application performance management has become a critical component of the ALM methodology. Development and test teams must design and run extensive testing to verify that applications have the ability to respond to load, and to provide scalability to meet increasing usage needs. Rapid application evolution means performance analysis must be performed quickly and efficiently. Often, the time-to-market demands may mean that mission-critical applications will not be sufficiently tested before being deployed into the live production environment, a situation that may have serious performance implications. And even well-designed applications that have been thoroughly tested in QA may face performance problems while running in a production environment.

Performance requirements for any J2EE system can best be met by capturing performance metrics for the overall distributed system and for each J2EE component by using a performance monitoring tool such as Optimizeit ServerTrace. The J2EE architecture is designed for developing multi-tiered distributed applications that provide interconnections between top-level tier presentation components, middle-tier business logic components, and back-end enterprise information systems such as database connectivity, messaging systems, and directory services. The challenging task of capturing performance metrics in such a way that problems may be effectively diagnosed and resolved may be completed using Optimizeit ServerTrace. When the inevitable performance issues arise, quick cycling to diagnose and repair the problem is a key indicator of process success.

The collection and analysis of performance metrics must be accomplished with a minimal impact on system resources of the J2EE architecture, or else the capture of performance data itself becomes a drag on system performance. Optimizeit ServerTrace utilizes non-intrusive methods to gather performance metrics, providing the assurance that metrics gathering does not impede performance.

Conclusion

Today's enterprise executive must build excellence into their IT organizations, to ensure that applications can be developed quickly, while providing scalability and performance in support of mission-critical business processes.

Borland has an unparalleled reputation for providing IT organizations with the tools that can deliver application excellence. With Borland Optimizeit ServerTrace, this reputation is solidified by ensuring performance is built into applications across the entire application lifecycle, through effective J2EE system monitoring, collaborative analysis, and diagnosis.