

Application Performance Management for J2EE, .NET, and SOA Web Applications

Tidal® Intersperse®

In the ongoing quest for real-time automation of business processes, enterprise information systems are evolving towards more and more web applications and service oriented architectures (SOA). In an SOA, existing and new software assets are broken down into libraries of components and services available to multiple applications and processes. Component relationships and process flows are becoming increasingly complex and dynamic.

As the complexity, immediacy, and scope of enterprise processes increase, so does the risk of failure. Comprehensive, proactive management of next-generation information processes and services is critical. Traditional application management tools are inadequate to manage next-generation cross-application and, eventually, cross-enterprise business processes and services because they are static, invasive, and unaware of changing application context. Businesses need a management tool that provides real-time visibility, context, and control from the application and data level through the dynamic service and business process levels.

Tidal Intersperse is the only management solution designed with the reach and sophistication to manage production business systems and processes built on J2EE, .NET and SOA. It ensures the continuity of vital, automated business processes by managing and monitoring all relevant tiers, their constituent components, and the interrelationships between those components.

Application Performance Management for J2EE, .NET, and SOA Web Applications

Evolution of Enterprise Applications

Over the past two decades, information systems have undergone a profound evolution in functional and geographic scope, the complexity of integration, and the immediacy of information they supply. New n-tier applications use application servers and component technologies such as Java, .NET and SOA to provide widely distributed application access via the Internet and to provide multiple applications with shared transactional access to data. Both applications and data have become distributed and componentized. The application has become the principal building block of the business information system (Figure 1).

Today, businesses are looking for automation gains at the next level: end-to-end process integration through SOA based on Web services. Now, instead of applications sharing data, complete business processes are built from applications packaged as suites of services. Through component technologies and evolving Web services standards, these new service-oriented business systems will make applications and components available across the enterprise and, eventually, across a company's whole ecosystem of vendors, partners, and customers. The new automated business processes are real-time and dynamic, fluidly calling on available services to carry out complex tasks.

Risk Management in the Service-Driven Enterprise

Over the years, companies have used a variety of management tools to ensure the stability and reliability of their computers, networks, and applications. From the days of the mainframe to today, the fundamental needs and benefits of system

management have stayed the same, but today's systems are more difficult to manage, and they present a greater risk in the event of failure.

In the mainframe era, software was linear and application behavior was predictable. If an application failed, it affected one task and one department, and there was generally time to recover before the failure affected business results. With the deployment of fully integrated, real-time, J2EE and .NET applications and service-oriented business systems, the complexity and risk have increased exponentially:

- ~ Each process has many components, many "moving parts" that can cause failure either singly or through their interactions.
- ~ Failure affects people and functions across the enterprise and, potentially, across the business ecosystem, risking valuable business relationships.
- ~ In a service-oriented architecture, components and services can work together dynamically. We cannot predict or test for every possible combination.
- ~ Real-time processes leave no margin to recover from failures.

Business continuity is dependent on the ability to predict, diagnose, and resolve potential points of failure. But traditional management tools lack the integrated view required to manage today's complex business systems. They were designed for visibility into vertical application silos. Their instrumentation is invasive, proprietary, and application-modifying. Performance management is typically based on profiling more suitable to linear programmatic applications. Service-oriented information architectures demand proactive management tools that provide full visibility across applications, components, services, and dynamic processes in an integrated production environment.

Requirements for APM in Production Management

Multi-Dimensional View

Tidal Intersperse views the world "horizontally" as well as "vertically," giving system managers the ability to understand the composition of the application and integration principles underlying SOA applications. While traditional tools provide visibility only within a vertical application silo, Intersperse tools also instrument and monitor the integration between applications and components. Figure 2 shows a typical business workflow such as order processing that has been integrated



IT complexity and management costs increased dramatically

Fig. 1: The Evolution of Enterprise Information Architecture

by encapsulating applications as services. The figure illustrates the limited visibility offered by traditional management tools versus the comprehensive view of the data, application, and integration tiers created by Tidal Intersperse.

Context-Aware

Because process flow is dynamic in a service-oriented architecture, system managers must be able to grasp and simplify the complexity of composite applications. Tidal Intersperse allows users to view components and services in multiple business contexts. For example, the same inventory management application might be used as a service in the context of an order processing process, a purchasing process, or an accounting process. Tidal Intersperse provides for context awareness starting at the design phase. System architects can map predicted process flows into unlimited different views to aid system managers, line-of-business analysts, and other stakeholders in monitoring, managing, and analyzing system behavior and performance.

Non-Invasive

Other management products are really profiling tools, using a range of techniques, most typically byte-code instrumentation, to extract management data from the managed region. Byte-code instrumentation was originally designed for the development environment, and it causes a number of problems when deployed on production applications. It degrades performance by bloating the byte-code, and it sometimes alters VM configuration, causing side effects from loss of VM tuning to full-blown application failure. Byte-code instrumentation can also block other dynamic applications that use byte-code generation, including some persistence

techniques, aspect-oriented programming tools, debugging and logging tools.

Tidal Intersperse is ideal for production environments because it allows users to easily and non-invasively instrument and monitor components and services. Intersperse does not depend upon invasive instrumentation for high-level application monitoring, rather it uses management data that is automatically available in the J2EE and .NET environments. It also has the ability to deep-dive into specific problem areas by dynamically turning on byte-code instrumentation when an issue is identified. This helps provide detailed diagnostics when issues arise, while keeping overhead low in a normal state. The lower dependence on instrumentation means there is minimal performance impact and minimal risk of causing application failure or interfering with other applications and tools.

Tidal Intersperse: Management Technology for Java, .NET and SOA Environment

Tidal Intersperse is the first of a new generation of tools created specifically for production management of J2EE, .NET and SOA applications and processes. Tidal Intersperse is multi-dimensional, context-aware, and non-invasive. It ensures the continuity of vital automated business processes by managing and monitoring all relevant tiers, their constituent components, and the interrelationships between those components.

Tidal Intersperse is built around the Java Management Extensions (JMX), the Java industry standard created by the Java Community Process to address the management needs of service-oriented architectures. JMX leverages the dynamic,

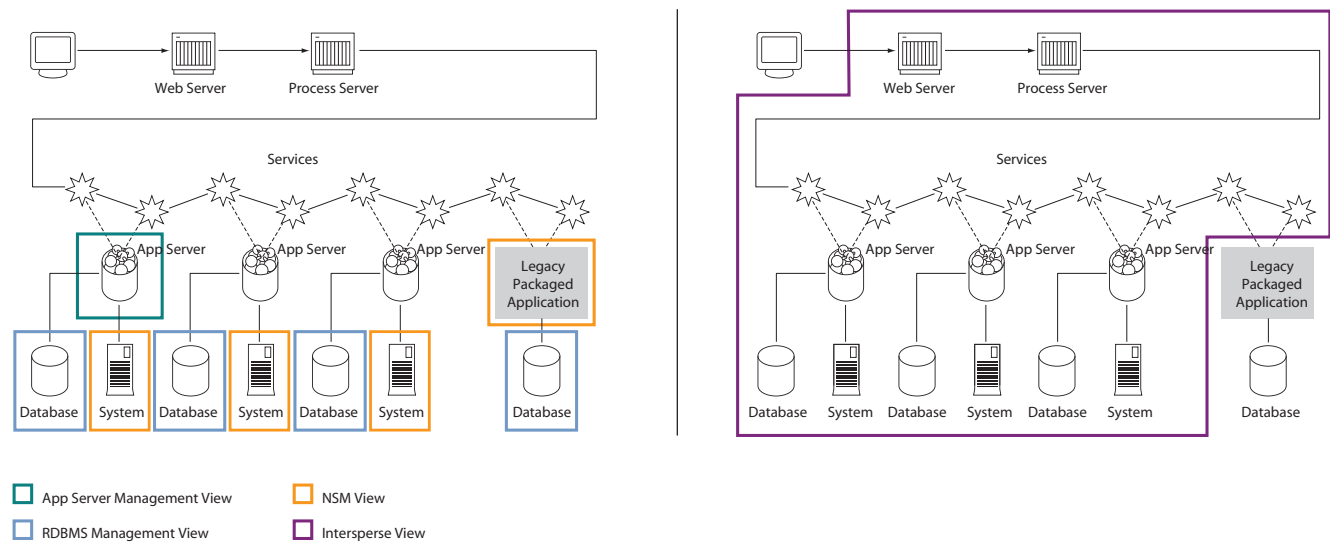


Fig. 2: Network, Data, or Application Visibility vs. SOA Visibility

Tidal Intersperse

Application Performance Management for J2EE, .NET, and SOA Web Applications

flexible, and portable nature of Java technology to manage components, applications, and services that interoperate dynamically across heterogeneous networks and platforms.

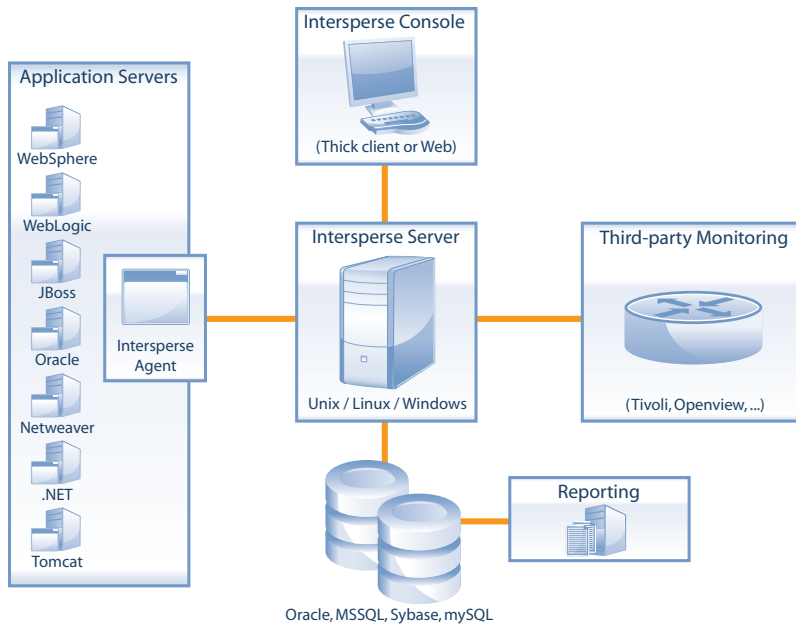


Fig. 3: Architecture of Tidal Intersperse

Tidal Intersperse has three main components:

~ **Intersperse Console:** A Java graphical user front-end for displaying application events, status, behavior, and configuration, and for analyzing and tuning applications in real time. Intersperse Console is a standalone, integrated collection of control panels, object browsers, dashboards, event and status monitors, and real-time graphs and displays. Nearly all aspects of the console are user-configurable, from the overall collection to the aesthetic details of individual dashboards and graphs. Wizards are used extensively to guide the user through the configuration and instrumentation process.

A read-only web interface is also available for day-to-day monitoring purposes.

~ **Intersperse Agents:** J2EE and .NET compliant components that are deployed in the target applications servers to instrument the application and to mediate between the target applications and the Intersperse Server. The Tidal Intersperse agent architecture is standards-based and non-intrusive. Typically, even fairly extensive MBean and WMI monitoring and instrumentation add no more than a few percentage points of performance overhead to the original applica-

tion. Object and method byte codes are used only for method invocation analysis¹.

~ **Intersperse Server:** J2EE server components containing the process and monitoring logic to implement standard and user-defined monitors, events, and actions. Intersperse Server connects to the agents under its command to retrieve instrumentation and structural information in the target applications and containers. The server also connects to the database to store and categorize events for later use by the console.

Tidal Intersperse gives users the ability to:

Discover

Tidal Intersperse automatically discovers and instruments components and services across a variety of common SOA platforms. Tidal Intersperse's powerful discovery features use agents to automatically detect and display a complex application's internal structure and configuration, giving visibility into an application's object attributes and relationships.

Map

Tidal Intersperse maps components in their business contexts and aggregates them into relevant multi-dimensional business metrics. Application mapping allows users to graphically define different logical views of an application's status, events, objects, relationships, etc., and to supplement predefined views and relationships with custom versions tailored to specific user types, applications, and business process models. Maps correlate lower-level operational and structural aspects such as database connectivity, heap memory usage, thread deadlocks, and object attribute values to business process-level issues such as service level agreement (SLA) shortfalls or order fulfillment workflow problems.

The mapping step greatly simplifies system management in the production environment. Often when complex production applications fail, the system architects or developers are called away from other projects to diagnose problems. By capturing the designers' knowledge of critical areas when the system is deployed, maps enable operations people to better manage the system without tapping developers.

Monitor

Tidal Intersperse's real-time monitoring system provides views of important statistics, a robust event-driven notification system, and vital alerts. Configurable dashboards graphically display metrics such as CPU utilization, disk space usage,

¹ Performance overhead is especially difficult to guarantee or predict exactly, and some tuning may be required to minimize long-term impact under final deployment conditions. Tidal Intersperse is an ideal tool to help with this tuning.

component execution time, number of successful invocations, and other statistics. Tidal Intersperse's rule-based service level agreement (SLA) and event definition features allow users to specify complex state- and trend-based alert and action rules simply and graphically in a single tool.

Analyze

Robust analysis capability allows Tidal Intersperse users to effectively triage performance incidents and assign them to the right team for resolution. Powerful tools for trend analysis, graphical drill down, impact diagrams, call trace visualization and SQL statement tracing further help root cause system problems and quickly hone in on the source of the issue.

Control

Tidal Intersperse streamlines application management by automatically correcting error conditions, which can deliver first level response to many incidents reliably and quickly. A rich set of corrective actions is available for any event reported by the system, ranging from changing configuration, enabling and disabling components, running scripts, to collecting more instrumentation data for further analysis. Tidal Intersperse users can also employ this ability to manually control components state, settings and data collection.

A Web Application Management Scenario using Tidal Intersperse

Let's look at how a JMX based APM tool such as Intersperse would be used with the on-line order processing system (Figure 4). In the service-oriented architecture, applications such as sales order entry, customer account management, order management, inventory management, and accounts receivable are encapsulated as services that can call each other at will to complete the order processing task.

Deployment

Setting up system management with Tidal Intersperse is simple. Let's say that the order processing system is implemented as Web services hosted in an application server such as BEA WebLogic. All Tidal Intersperse components are supplied as standard Java application archives (war, ear, and jar files). They would be deployed into the application server just like any other Java components. Tidal Intersperse might be deployed either during development or when the application goes into production.

Setup for System Management

When Tidal Intersperse is deployed, an Intersperse Agent automatically uses the JMX API to traverse the J2EE object hierarchy and discover all Java components: EJBs, servlets, connectors, portal elements, etc. It builds a "replica" that is used by Tidal Intersperse to present views of the component tree that can be navigated by server, server group (cluster), or J2EE component type. Initial discovery typically takes 10-20 minutes, depending on the number of objects in the system, clustering, etc. Once the replica is built, Tidal Intersperse periodically checks for new components and automatically updates the component list.

Next, the components are organized into relevant contexts that will be used to manage the system. A system architect, developer, or other person knowledgeable about the application uses the drag-and-drop mapping tool to create a set of operational views that can be shared with other users. For example, a developer's view might include a number of individual components while an operations manager's view would include only high-level information such as overall performance of the whole process and status that's relevant to service-level agreements (SLAs) with vendors or customers.

Tidal Intersperse enables developers to configure common views such as disk space, CPU utilization, and queue status that are defined by industry best practices. Based on his or her knowledge of the system and likely points of failure, a developer might also set up a view that just shows information such as DB-related elements or JDBC connection pools. Developers or operations managers also create views for different business stakeholders such as line-of-business analysts or business process owners, so that each stakeholder sees only the information relevant to their job.

Once views are set up, it is easy for developers or operations managers to define events, alerts, and actions to monitor and manage system health and performance. Tidal Intersperse has a number of built-in monitoring points. For example, every component has default characteristics that can be used for reporting or to trigger alerts and actions - such as to notify users whether a system is up or down, whether a resource is available, etc. Users can set up additional alerts driven by events such as available memory or JDBC connections reaching defined thresholds, status changes such as a component going off line, or failure to meet SLA parameters.

Application Performance Management for J2EE, .NET, and SOA Web Applications

Once events are defined, they can be tied to actions, allowing Tidal Intersperse to correct problem situations automatically. Any event can be tied to any object function in the application, or an operating system command line operation can be defined as a threshold crossing action. So threshold events might be tied to actions such as automatically increasing the number of JDBC connections or the frequency of garbage collection.

Day-to-Day Management

In everyday use, an operations manager might keep the Intersperse Console open on his or her desktop to monitor the status of various resources, or use the dashboard to watch performance trends. In a service-driven enterprise, order processing might be one of several processes that use the same underlying applications, so the operations manager might be watching several managed views at the same time. If potential problems arise, Tidal Intersperse alerts operations staff via e-mail, pager, or other channels. For planning and analysis purposes, Tidal Intersperse generates basic reports on system status and trends, or integrates with standard reporting tools such as Actuate and Crystal Reports.

The beauty of Tidal Intersperse is that it corrects most problem conditions in the course of day-to-day system management, before they affect operations. Because of the rich system of events and actions created during system setup, by the time the IT team is alerted to a threshold event, Tidal Intersperse will have already taken action to correct the situation. For instance, exceeding a threshold on memory consumption could automatically start a garbage collector to free memory. If Web servers are problematic, Tidal Intersperse can start and

stop them, or launch more Web servers to scale for throughput. The situation is corrected before users are affected, leaving operations staff to deal with the problem area at their leisure.

Dealing with System Failure

While traditional management tools rely on device-centric technologies or one-way instrumentation such as BCI, Tidal Intersperse's JMX-based instrumentation is bi-directional and object oriented, which greatly simplifies problem resolution. Not only can Tidal Intersperse alert users to potential problems, it also allows them to interactively exercise object functionality in the production environment to diagnose or correct problems. For example, an application server might have an alert defined when available memory drops below a certain threshold. In response to the alert, the operations manager can manually change parameters to run garbage collection more often.

With event-driven alerting and fine-grained control of the production environment, Tidal Intersperse allows operations managers to head off most failures before they happen. But let's see how Tidal Intersperse might be used in case of catastrophic failure. In Figure 5, market conditions have caused an unexpected and unprecedented flood of orders that require special handling. As the order volume increased, the system has frozen. A service somewhere in the process flow is not responding and other processes are coming to a halt waiting for responses. By the time the operations manager receives a message and looks at the Intersperse Console, there are dozens of errors and alerts showing.

At this point, the operations manager knows only that there has been a cascade failure. With a traditional profiling-type management product, the manager would have no simple way to diagnose the problem and prevent a future failure. Developers might be called in to analyze system logs or stack dumps. With Tidal Intersperse, the operations manager can quickly use the root cause analysis tool to trace back through the events and determine the origin of the failure (Figure 6). In our scenario, the operator finds that a JDBC connection pool not heavily used in the normal flow of order processing was emptied during the flood of special orders. The JDBC issue impacted an EJB, which impacted the Web service using the EJB, and the failure eventually propagated across the entire order handling process.

To correct the problem, the operator interactively increases the size of the JDBC connection pool and then sets new thresholds,

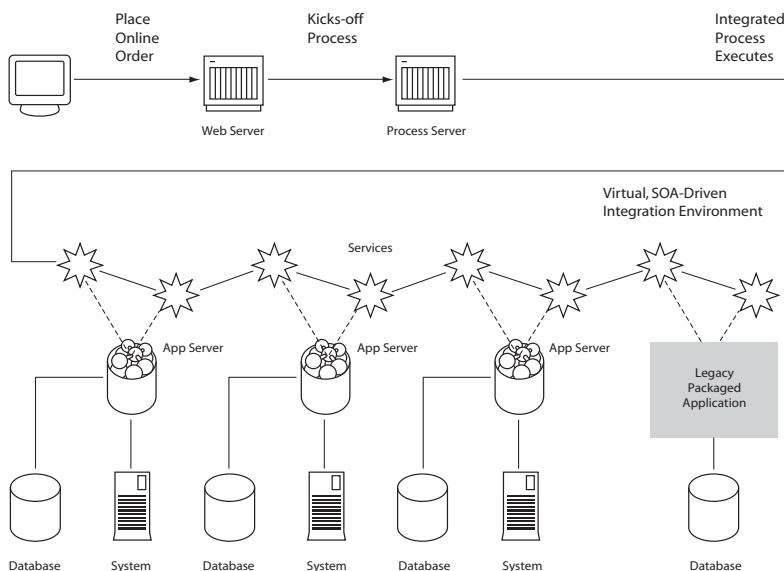


Fig. 4: A Service-Oriented Architecture for Order Processing

alerts, and actions on that component to prevent the problem from happening again. The problem is solved quickly and painlessly.

Make the Proactive Choice

The next generation of enterprise information systems are being built with service-oriented architectures that enable complete process integration across the enterprise and, eventually, across whole business networks. Along with the benefits of this integration come heightened risks because of the greater scope, immediacy, and complexity of these process-driven systems. To fully reap the benefits and mitigate the risks, businesses need a new kind of SOA management tool that provides proactive, context-aware visibility and control at all levels of the service-oriented architecture.

Tidal Intersperse is the first and only management tool designed to help master the rapidly growing complexity of SOA business systems. It gives developers, operators, and analysts a comprehensive view of all relevant tiers, the ability to proactively monitor and analyze system performance in business contexts, and the control to automatically or directly correct problems in the production environment. As organizations evolve their business systems to service-oriented architectures, they cannot afford to be without Tidal Intersperse.

About Tidal Software

Tidal Software is a leading provider of application scheduling and performance management software that radically simplifies IT operations by automating and integrating performance and process management. Tidal's solution puts IT operations management into the business process using new levels of automation, visibility, and control over systems. Tidal makes applications such as SAP®, PeopleSoft®, and Oracle E-business Suite™ more efficient, reliable, and secure to return greater business value. Tidal also gives IT deep visibility into and precise control over new SOA-based composite solutions consisting of packaged applications and custom components in Java and .NET. General Mills, HP, ING Direct, Microsoft and T-Mobile are among the Tidal customers who use Tidal to reduce operational costs by running IT at higher efficiency rates and containing overall data center footprints. Privately held, Tidal is venture-backed by Kleiner Perkins Caufield & Byers, Novus Ventures, Panorama Capital, and VantagePoint Venture Partners.

For more information, contact Tidal Software at 1-877-55-TIDAL or visit www.tidalsoftware.com.

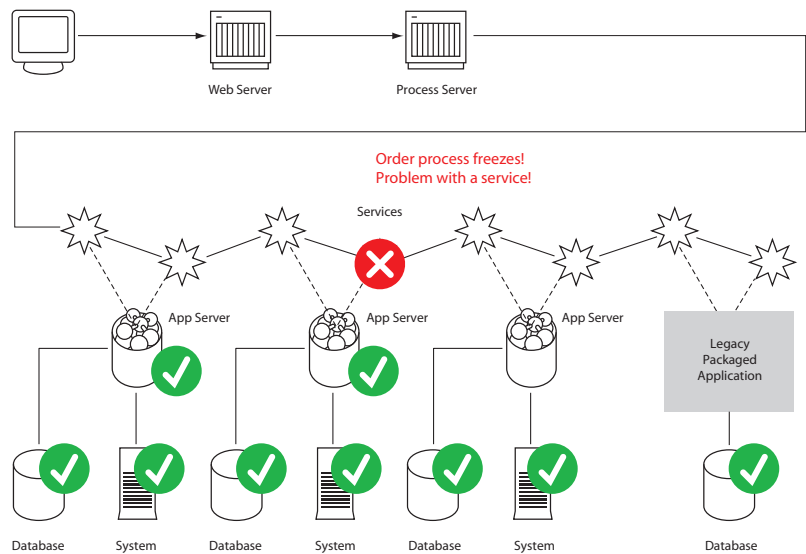


Fig. 5: System Failure in SOA—How do you locate the root cause in a dynamic architecture?

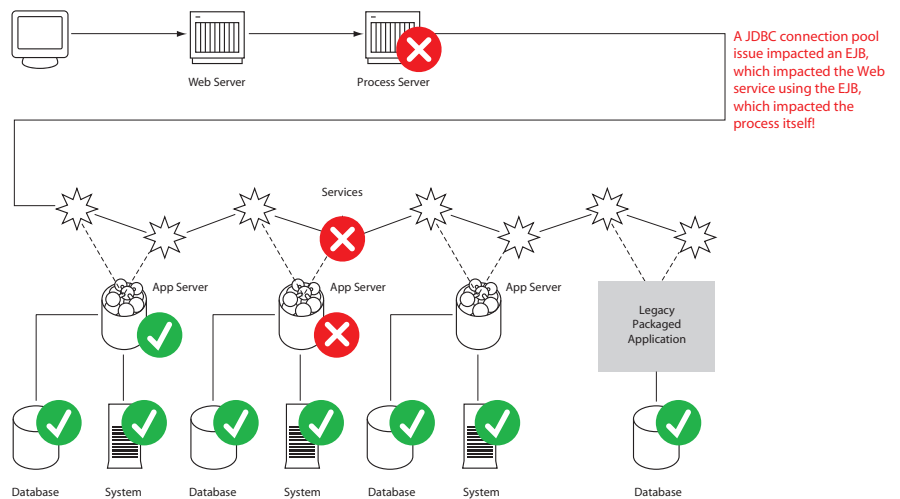


Fig. 6: Root-Cause Analysis Pinpoints the Source of Failure—Intersperse Manager correlates data across the tiers, enabling the operator to isolate the cause of the problem.

TIDAL[®]
S O F T W A R E

Corporate Headquarters

2100 Geng Road, Suite 210, Palo Alto, CA 94303
1 (877) 55 TIDAL

EMEA Headquarters

Tidal Software Limited
Siena Court, The Broadway, Maidenhead, SL6 1NJ
United Kingdom
+44 1628 509 020
+44 1628 509 120 (fax)

info@tidalsoftware.com

www.tidalsoftware.com