



# **The Revolution in Managing Enterprise Transaction Systems**

A Whitepaper from BlueStripe Software

## Summary

In today's data centers, IT Operations teams must resolve problems with critical business applications quickly and efficiently. Many teams have realized that transaction monitoring is a vital piece of comprehensive availability and performance management. Unfortunately, transaction monitoring tools have failed to deliver the necessary capabilities of monitoring transaction performance and figuring out the cause of performance and availability problems.

In fact, none of the conventional IT Management tools are helping Operations teams maintain application availability and performance – from massive frameworks and specific technology silo tools to APM and BTM tools. At the heart of the management gap is the complex and dynamic nature of the application systems running the transactions. As a result, Operations teams try in vain to piece together data from multiple different sources to understand application systems; and solve problems through resource-consuming bridge calls and war room sessions.

This paper outlines a new approach to managing application performance and availability for IT Operations teams; one that manages transactions, the applications that run them, and the underlying infrastructure they depend on. This new approach starts with visibility of the entire application transaction system, not just piece-part technology silos. With real-time views of the complete application system - including all transactions, connections, and interdependencies, IT Operations can approach application transaction monitoring and problem solving in a revolutionary way.

### Key Takeaways:

- Transaction monitoring is critical to the success of the business, but Operations can only be effective by understanding the dynamic relationship between transactions and the infrastructure
  - An effective application support tool must manage all application system components, not just Java code, network packets, or system resources
  - Standard Systems Management, Application Management, and Business Transaction Management tools have too many data gaps, preventing them from solving availability or performance problems
  - Traditional approaches to managing transactions have not been successful due to fragile instrumentation and lack of integrated visibility into the systems that support those transactions
- BlueStripe takes a new approach that follows transactions hop by hop through the application system, isolating where time is spent and why

## Where Application Management Falls Short

Despite years of effort, managing the availability and performance of business applications is more difficult than ever. IT Operations teams struggle to manage performance and availability proactively, solve problems quickly and cheaply, and gain control and visibility as application systems move to the cloud or roll into production.

In conversations with hundreds of enterprises, BlueStripe hears the same problems and challenges. Managing business application performance and availability has become a core responsibility for today's IT Operations teams, yet they don't have the right level of visibility to be successful. In their own words, IT leaders at Fortune 500 companies have described the problems that keep them up at night:

- **Problems remain unsolved despite entire team effort:**  
*"There are a large number of components and servers in the environment, and it can be time consuming to find a problem. We need a tool that allows us to quickly isolate where the problem is without bringing my entire team into every incident, even with so many different components involved."*  
- VP of Applications and Infrastructure Management, Fortune 1000 Materials Company
- **Frustration in preventing problems:**  
*"Sometimes we think we have problems fixed and control over our applications, but all we're doing is system reboots, leaving underlying problems unaddressed. We don't have a way to ensure that problems won't happen again."*  
- Director of IT Operations, Fortune 50 Financial Services / Banking
- **Lack of visibility to all critical components:**  
*"It's really a political dilemma. The problem is that we have dependencies on systems outside of our data center and we can't get access to other people or resources that can help us find the slowdown. The vendors tell us the problem isn't in their area but we have no way to verify that."*  
- CIO, Web Site Hosting Company
- **Blocked virtualization and cloud initiatives:**  
*"My boss is wondering if this CRM application is really suited for virtualization, or whether we should just keep it on the physical infrastructure. I can't continue to virtualize unless I know how to measure its impact on application performance."*  
- Manager of Computer Operations, Major University

These challenges all spring from a common source: application management and transaction management tools have not kept up with the modern realities of the complex, dynamic transactional business services they are intended to manage. Even small organizations with basic two-tier or three-tier application systems face these issues. Complexity emerges quickly from even the most basic application system, making management challenges being felt across a wide range of organizations, from small local governments to the world's largest banks.

# The Mission of Application Transaction Management Tools

There are two fundamental reasons that Operations teams have reached a breaking point that management tools can't cope with: *overwhelming complexity* and *dynamic change*.

Applications are more complex, interconnected, and business-critical than ever before. To be successful, management tools must deal with the fact that applications are both a service that your business provides, *and* a set of interconnected IT infrastructure.

An **application transaction system** is the complete set of components, servers, and processes that can affect an application's performance.

The only way to successfully manage application performance and availability is to have visibility into transactions, the applications that run them, and the underlying systems they rely on.

An effective application support tool must manage the entire application transaction system, not just Java code, individual transactions, or system resources.

The second reason for the frustration with application management is the unprecedented rate of change affecting application transaction systems. Applications have become enormously dynamic, with IT often rapidly leading the business to new opportunities. Unfortunately, management tools have not kept up with the pace of change. The dynamic nature of load balancers, clustering, virtualization, and private cloud has created a set of problems not covered by old tools.

To work effectively in IT Operations, application and transaction management tools must automatically keep up with the pace of change, and they must integrate into existing processes without creating unrealistic expectations for IT Operations.

## The Limits of Conventional Approaches

Conventional application management and transaction management tools require manual updates to maps, massive instrumentation, and arduous application definition setup. These requirements create a high cost of ownership to achieve visibility – a visibility that is ultimately incomplete. A vicious cycle takes hold where tools get out of date and then get used less and less – which makes them get even further out of date. Eventually, the tool becomes shelfware.

To understand more about why conventional tools struggle to manage highly complex and dynamic application systems, it is worth examining four broad categories of management tools.

### Infrastructure and Silo Management Tools

Silo tools focus on a single area – such as network, virtualization, or storage - and may touch on application systems, but they are not designed to manage applications. Any view of application behavior is indirect. Without broad application context it is usually speculative and inefficient to focus on a given silo tool when an application is having problems.

Infrastructure management tools are a special type of silo tool that focus on servers and their resources. This is a critical point of visibility, but like other silo tools the view of the application system is still missing. Most enterprises

already have good infrastructure tools in place. Some of these tools identify which servers are available, while others analyze what's installed on those servers, but they ultimately lack dynamic application context.

A final way that tools in this category claim application awareness is by doing sophisticated analysis of system data, with techniques such as mathematical heuristics, log analysis, and complex event processing. The problem is that the underlying data lacks any notion of application interactions, so any conclusions are uncertain and incomplete.

### Business Transaction Monitoring Tools (BTM / TPM)

Forward thinking IT Operations teams have concluded that they must consider transactions as a key part of managing business application availability. A set of new solutions deliver Business Transaction Monitoring (BTM) or Transaction Performance Monitoring (TPM) to take advantage of that desire. Unfortunately, the actual solutions from these vendors take approaches that are difficult to setup, arduous to maintain, and fall short of the Operations team goal of actually solving application performance and availability problems.

One problem with BTM tools is that they require significant coding (or instrumentation) of each transaction node for tracking purposes. This time consuming process usually requires highly skilled resources from both the vendor (to do the instrumentation) and the customer (to identify WHERE to instrument). Not only is this approach costly and invasive, it **WILL MISS** significant components of the transactions unless every single transaction node is identified and instrumented ahead of time. This invasive, costly approach also means that changes to applications or transaction paths will require similar costly instrumentation efforts, with the same risks, all over again.

Another BTM implementation problem is the requirement to keep every transaction for analysis and reporting. This, of course, leads to "MDB Syndrome" (Massive DataBase) problems, maintaining terabytes of performance data every month is just the tip of the iceberg. Whenever problems occur, the Operations team must sift through the hundreds / thousands / millions of transactions to figure out which ones are applicable to the problem.

Despite the above problems, users of existing BTM & TPM tools complain of an even bigger problem. They complain that although they invest in the application layer instrumentation, build and manage large MDBs, and deal with sifting through the data, the tools only tell them which server node is the bad node. It doesn't help them understand WHY. The BTM & TPM tools can follow packets and flows, but they cannot go down through the stack to identify which individual component is the problem and why.

### Traditional Application Performance Management Tools

Most of these tools focus exclusively on Java/.NET components, specifically on code internals and Java/.NET transactions. The standard technology approach for these tools is byte-code instrumentation (BCI). While this category provides visibility into Java/.NET internals and a narrow slice of transactions, it does not provide hop by hop transaction visibility across all tiers.

Owners of Java/.NET silos find these tools useful, but the available data does not allow IT Operations to manage interconnected application systems holistically. IT Operations managers are still missing full visibility, and they find that there is no one person or resource that understands the entire application system. Architects and specialists remain on call 24/7 for problem management,

When critical **application transaction systems** are at stake, performance management tools must provide direct, accurate measurement of all application interactions.

Existing tools only offer visibility into some parts of your application systems. There is a need for a tool that takes a new approach.

and the full set of deployed applications is often unknown. Surprises are common, problem solving is slow and expensive (or impossible), and application changes are difficult to control.

### Management Frameworks

This is a well-known category of large, expensive, complicated suites that attempt to tie many tools together - sometimes more than 20. These tools tend to focus on server and network availability, and are often broken by modern, dynamic applications running in the cloud and across virtual machines. Many frameworks have added a transaction tracing tool to the collection of individual tools. These "BTM" like tools fall prey to the same problems of stand-alone BTM tools (too much setup, brittle to change, no actionable information for solving problems).

Aside from very high total cost of ownership, IT managers often report that these tools alert on the wrong things and only allow them to monitor what they already know about. IT personnel have to teach the tools what they know, rather than learning new things from the tools they have invested in.

In cases where the management framework integrates a large number of tools, the result has all the incompleteness described in the earlier categories, but with the even greater administrative cost of so much tool proliferation and upkeep.

# FactFinder: Application Transaction Management for IT Operations

BlueStripe was founded to make the people who manage the health and availability of application systems in IT Operations successful. BlueStripe’s FactFinder solution is unmatched in managing complex application transaction systems and solving the hardest problems that affect critical business transactions. It achieves this by monitoring transactions, the applications they run on, and the underlying systems those applications depend on – together – in a single tool.

FactFinder provides actionable information to measure, improve, and resolve issues with application performance and availability. FactFinder is unique because it automatically identifies all applications, and combines this information with critical system infrastructure activity. This is what allows our users to find application problems where narrowly focused toolsets aren’t looking. FactFinder also uniquely combines performance data with comprehensive, fully automatic dependency maps. This integration allows you to understand all interactions and behavior in your application system.

### SOLVING APPLICATION TRANSACTION SYSTEM PROBLEMS WITH FACTFINDER

FactFinder identifies process-to-process connections, isolates application-specific transaction dependencies, and measures performance hop-by-hop and down through the stack. Our data originates with the BlueStripe Collector, an agent that runs on each server in your environment, gathering metrics for every TCP connection on every connected process.

Information from BlueStripe Collectors is stitched together to create a unified view of all process-to-process dependencies, including virtualization context and system metrics, to create dynamic application system maps.

The map shown here provides a view of all machine-to-machine dependencies for an application system. You can follow transactions hop-by-hop, and can drill down into every connection and every process that affects transaction behavior.

Transaction dependencies are analyzed to automatically discover application systems, and these application systems become the focal point of all your management tasks. All performance data, including transactions, storage, servers, processes, and connections, is shown in this context. BlueStripe supports any application in physical, virtual, and cloud environments, and it does this with zero required configuration, coding, and maintenance.

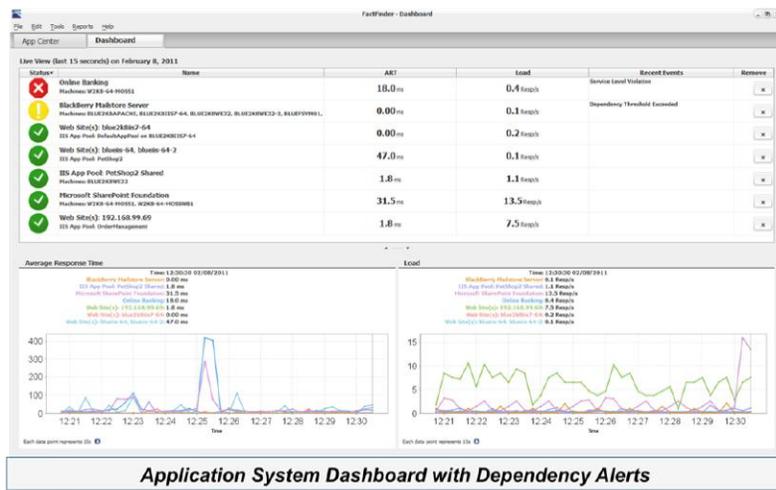
## Transaction Monitoring and Mapping

IT Operations needs accurate, automatic, and comprehensive transaction path and application topology maps to understand the impact of dependencies on application health and to avoid problems with critical business transactions. FactFinder enables this by:

- Dynamically mapping all application transaction paths and interactions
- Monitoring & alerting on transaction requests to any application component
- Integrating with third party management tools & processes

A Fortune 100 insurance company illustrated the benefits of this approach when it was confronted with availability problems in a critical application. Their existing tools couldn't see all critical dependencies, and they required manual updates after every application change. Over time, the error-prone, labor intensive process created incorrect information about the application transaction environment. When users called to complain, Operations and support struggled to address the availability issue. IT Operations had hoped to rely on the maps provided by a well-known tool vendor, but with out-of-date information, these maps had crucial gaps in application transaction system visibility.

After a few weeks of frustration, the company brought in BlueStripe. FactFinder mapped the application topology and transaction paths (with all dependencies) within an hour. FactFinder showed that a custom C++ application component had a dependency (unknown by the Ops team) on a third party credit card service that would hang transactions and ultimately trigger the slowness.



This was the information needed to fix the problem, but the team went further with FactFinder. They set up a transaction system dependency alert for the credit card service, proactively notifying support when transactions would begin to back up due to the third party company. This alert was automatically rolled into the FactFinder Dashboard (shown here), to warn the team before end users could be impacted.

FactFinder transaction dependency alerts alert Operations teams to issues on any component across the transaction – anything that the transaction depends on.

These monitoring capabilities can be integrated with any third party tool using standard mechanisms, which the team used to feed the proactive alerts into their NetCool framework.

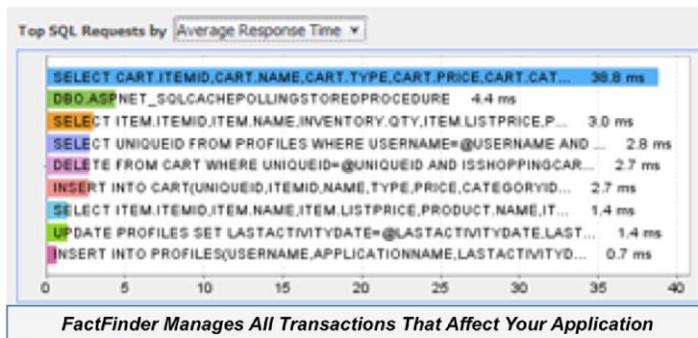
## Root Cause Analysis Using Transactions

FactFinder provides root cause analysis for any application transaction system problem. This is done in one unified view that:

- Follows transactions from process to process
- Measures response times in every tier
- Drills down into servers, processes, and transactions to isolate issues

FactFinder allows users to quickly and easily identify the owner of any problem by following transactions right to problematic components, whether a process, a transaction protocol, file system, virtual machine, network, database, etc. Identifying that a problem is poor code design, configuration issues, virtualization performance, or storage bottlenecks allows assignment to specialists only when it's isolated to the appropriate silo they own, rather than bringing the entire specialist team into every problem.

Even in Java/.NET applications where a code internals tool is in place, it's enormously important to have visibility into every transaction dependency that can affect the application system. This reality was acutely felt by a major web site hosting company that was unable to solve an embarrassing performance problem with Java transactions that lasted several weeks.



BlueStripe FactFinder discovered that Symantec anti-virus processes deployed on critical servers would make updates during peak business hours, driving a huge amount of load through a critical file system and slowing down all transactions. The anti-virus updates were supposed to run in the middle of the night, and the team was shocked to find them behaving this way.

In a similar case, a Fortune 1000 materials company had deployed a popular Java/.NET management tool, but they struggled to diagnose

a memory leak that required them to restart their application every 24 hours. BlueStripe was brought in just before IT Operations was to redeploy the entire application from scratch. Within an hour, FactFinder showed that custom code was attempting to establish a connection with a phantom instance of SQL Server every 12 seconds. These failed connections caused the memory leak, and until IT Operations had complete visibility into all application system interactions (including broken interactions), the problem could not be understood.

All too often, problems remain unsolved despite the dedication of an entire team. With FactFinder's unmatched visibility, support teams are now able to isolate problems systematically, without speculatively resorting to silo tools. This has allowed FactFinder customers to solve problems both cheaper and faster. The next step organizations take with FactFinder is to add application system benchmarking and forecasting to their Operations processes, thereby preventing issues from ever occurring.

## Benchmarking and Forecasting Application System Performance

FactFinder allows IT Operations to manage and benchmark applications across migrations, such as production deployments or virtualization projects. This core capability allows Operations to catch problems before they impact users. FactFinder provides the following benefits:

- Benchmarking of application transaction systems
- Automatic comparison of applications and transactions before & after any change
- Enablement of confident application migration and updates

An example from a Fortune 50 financial services firm helps demonstrate the enormous value of these capabilities. The bank could detect transaction problems once they occurred, but it did not have actionable information to measure and improve application performance using a systematic process. They simply had no way to avoid problems proactively, and could not get control of their applications or transactions.

With FactFinder SnapShots, the bank could benchmark applications and transactions across pre-production testing phases, isolate any problems that occurred, and quickly compare their benchmarks against production performance when any application changes were rolled out. This allowed Operations to validate that the application structure was reliable, that transactions were spending time in application tiers according to architectural goals, and that any problem that escaped into production could quickly be checked against the staging environment to understand where any changes had occurred. They also used these indicators of testing gaps to continually improve their testing processes.

After they achieved visibility into the entire application transaction system with FactFinder and were able to benchmark performance on all interactions, overall production incidents decreased by 40% and mean time to resolution (MTTR) was reduced by 75%.

## Conclusion

If IT Operations teams are going to meet the challenge of managing application availability and performance, they must manage the entire application transactions system – including transactions, the applications that run them, and the underlying systems they rely on. A tool that only manages one of these pieces, whether Java code, URL response time, or system resources, just isn't enough. BlueStripe FactFinder provides complete visibility into all interactions and dependencies among transactions, servers, systems, and processes. This visibility allows IT Operations to effectively manage application transaction system availability and performance across all critical management tasks, including monitoring and mapping, root cause analysis, and benchmarking & forecasting.

FactFinder follows transactions in real-time wherever they go – across tiers, across platforms, across architectures, even into virtual machines, private clouds, and interactions with third parties. This is done automatically, and updates dynamically, without a dedicated administrative staff, and can be installed in hours, not days or weeks. Organizations that have embraced this new model of application transaction system management are finally able to gain control over entire application transaction systems, and systematically improve performance and availability of critical business transactions.