



White Paper

360° Application Performance Management

Best Practices for Gaining Comprehensive Visibility Into Enterprise Application Performance

Publication Date: December 2008

Abstract:

APM focuses on monitoring and managing the performance and service availability of software applications. Enterprise companies are increasingly seeking holistic, or 360°, APM solutions, capable of monitoring and managing a broad range of applications, in order to:

- Provide early detection/preventative care relating to application issues to reduce the number of trouble tickets.
- Determine root cause more quickly and have the power to fix a problem then-and-there in order to reduce the MTTR (mean time to repair) and minimize impact on customer experience, SLAs and revenue.
- Deliver a comprehensive, integrated solution that reduces overall TCO and makes the goal – end-to-end application monitoring and control – attainable.

This whitepaper covers the challenges that organizations face in providing solutions for 360° visibility, requirements for a solution, recommended architectures and best practices which have been gathered from many enterprises addressing the same issue.

Copyright

© 2009 Sherrill-Lubinski Corporation. All Rights Reserved.

Trademarks

SL Corporation, SL-GMS RTView, RTView, and the SL logo are trademarks or registered trademarks of Sherrill-Lubinski Corporation in the United States and other countries.

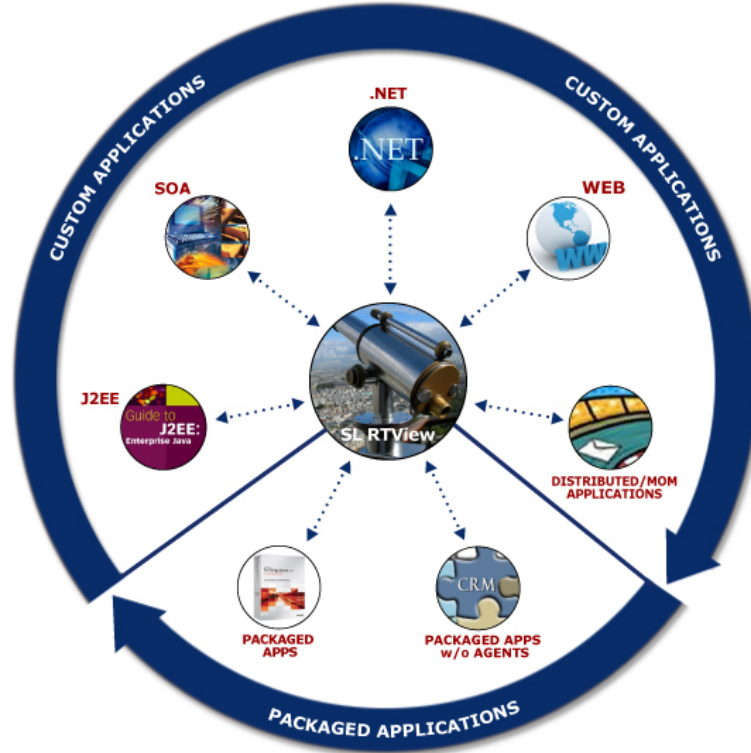
Contents

Executive Summary	4
Background	5
The Infrastructure and Operations Challenge	7
Application Developer and Support Team Challenge	7
360° APM Architecture	8
Business Value of 360° APM	9
Architectural Requirements for 360° APM	9
Access to All Data	10
Data Calculations and Service Model Definition.....	11
History	12
Rules Engine	12
Alert Management.....	12
Scalability and Support for Distributed Architectures	13
Flexible Real-time Dashboards and Reporting.....	13
Ability to Connect With Incident Management Systems.....	14
Best Practices	14
The Reality:	14
Where to Start:	15
The Result:	16
About SL Corporation	16

Executive Summary

Application Performance Management is increasingly becoming a critical part of enterprise operations. It is no longer a “nice to have” to improve customer experience, increase loyalty and reduce support costs. Outages in supply chain, eCommerce, telecom, data services and financial services become headline news indicating how well your enterprise can competitively deliver. There are some solutions that can help monitor packaged applications that are likely supporting your internal business units.

Unfortunately the majority of the critical external applications that an enterprise depends on for competitive advantage are custom developed in-house and are not covered by those monitoring solutions that focus on packaged applications. What is needed is a 360° view into all enterprise applications that allows for rapid problem alerting and analysis, and reduces the number of trouble tickets by predicatively analyzing when systems begin to perform out of normal or compliant bounds. Many organizations are looking for best practices to effectively manage this critical hole in IT operations support in an effort to reduce the risk of public failures, and to move their organizations into standard ITIL recommended practices for Incident Management, Availability, Capacity and Service Level Management.



This whitepaper covers the challenges that organizations face in providing solutions for 360° visibility, requirements for a solution, recommended architectures and best practices which have been gathered from many enterprises addressing the same issue.

Background

There are a variety of different commercial solutions that are available to IT Operations which are meant to manage application performance and availability. Many of these solutions however are geared toward packaged applications or applications which have pre-defined types of performance-related metrics that can be gathered. This leaves a percentage of applications that either cannot be covered by those solutions or require costly customization efforts to do so. Some of the most difficult applications to gain performance visibility are custom-developed applications. Unfortunately in many enterprises, these applications provide the most critical services that result in the deepest financial impact if performance falters. What is needed is an Application Performance Management system that involves all critical applications and provides a

360° view of enterprise performance to operations staff, application support teams and line-of-business stakeholders.

IT Operations has always had a difficult time using any standard Service Management tool to properly monitor and determine root cause problems for custom-developed applications. This is because these applications involve multiple components that operate across different application layers and distributed enterprises. These components may be custom-built software services, as well as software infrastructure components such as database servers, message brokers or application servers. Therefore when the help desk receives trouble tickets related to these custom applications, the responsibility for analysis and resolution most often comes down to the application developers or application support teams.

In organizations where custom-developed applications are the most critical part of the business, this has created a very pressing initiative to reduce the number of trouble tickets and MTTR (Mean Time To Repair). Because of the critical nature of these applications, this has given rise to dedicated departments – either in development or as an offshoot of operations – where their main objective is to determine best practices and lower TCO for this type of incident management.

IT application developers have traditionally built a variety of handcrafted solutions to debug their applications and to provide information about application performance after deployment. These solutions range from running some scripts, writing application metrics to log files or databases, and instrumenting with JMX or WMI, to completely handcrafting specialized dashboards that monitor application metrics.

To complicate matters, a variety of application performance monitoring tools have been introduced in the market that most often are trying to solve a specific and isolated type of performance problem. Some examples of these solutions are synthetic transaction monitors, packet analysis probes, and JVM monitors. In practice, these tools are typically used for deep analysis by people trained both on the details of the application in-question, as well as the tool itself. In a large organization, this situation often leads to hundreds of point solutions which may involve thousands of specialized agents and their associated incident rule engines, dashboards and reporting facilities.

The Infrastructure and Operations Challenge

When this type of problem is tackled by the infrastructure and operations side of the business, they are often drawn toward Business Service Management (BSM) solutions or Application Performance Management (APM) solutions that are geared towards monitoring packaged applications like those from the former Peoplesoft, Oracle, or SAP.

BSM solutions are really not applicable to these sorts of problems because they are oriented in a bottom-up methodology. They discover hardware components and the software executing on that hardware, store the dependencies in a configuration management database (CMDB) and implicitly derive application performance from the underlying hardware performance metrics. For custom applications, this won't work for two reasons: 1) custom applications can't be discovered in this manner, and 2) the underlying hardware metrics like CPU usage and memory usage do not have defined mappings to actual overall application performance.

APM solutions for packaged applications are also limited in that, most often, 1) they do not have access to the software component metrics necessary for a broad spectrum of custom applications, and 2) they do not have the flexibility to visualize the data in a way that represents how the application really performs in a distributed environment.

Whether the proposed solution is called BSM or APM, both are even further complicated by SOA and Virtualization. SOA applications can be dynamic and perform load balancing over a number of software components, and Virtualization can abstract the relationship between software component and dependent hardware. Without a way to capture the relevant metrics and relate them to the actual process flow, these solutions can't capture overall application performance.

Unfortunately when Operations tries to shoehorn monitoring of custom applications into these environments, most often they fail, are very costly to create and maintain, or don't live up to the original expectations.

Application Developer and Support Team Challenge

When application developers or application support teams address this problem, the main difficulty is usually coming to consensus. Each team has its own pre-existing tools for analyzing application performance that are perhaps limited in their scope, but might

otherwise be quite useful in their particular situation. A solution which could incorporate data from these existing tools and even provide drill down capabilities to specialized tool dashboards can help alleviate any fears of change. The major problem however is that these teams are professionals at creating solutions. When presented a problem, they tend to lean toward designing their own solutions and until they get deep into it are not aware of all the necessary components for an enterprise solution to succeed.

Many of these teams eventually brought in traditional APM solutions to ensure the overall quality of services to users, but still were unable to fully solve the problem. Within distributed systems, the APM market started with operating systems agents that merely monitored CPU, disk, network traffic and other metrics. Software vendors then began offering agents that not only monitored the operating systems, but elements of the applications themselves. Early agents focused on monitoring database applications, web servers, email and other stand-alone business applications like SAP and Peoplesoft. When users experienced problems, however, these monitoring tools could not identify the cause of the problem. Application developers and support teams began pointing fingers at each other, as they could not identify what component of the infrastructure caused the degradation. The countless hours spent tracking down the root cause were more like detective work than running an IT organization. This frustration caused many organizations to seek tools that could provide end-to-end monitoring and identify the root cause issues.

The APM software market today includes nearly one hundred vendors that claim to offer APM solutions, but most focus on a narrow slice of monitoring. These software vendors offer synthetic transaction monitoring, network packet capture, application agents, or visualization portals. A few vendors have developed or acquired multiple tools to offer a combination of these solutions to their customers. Very few offer a holistic APM solution that collects data, analyzes the data and provides a single interface to the end-to-end performance of the application.

360° APM Architecture

What is needed is single-pane-of-glass visibility, a 360° view, that can show aggregated real-time and historical information about application performance as well as drill down to software-component root causes, and that can be easily understood by operations staff, application support teams and line-of-business managers.

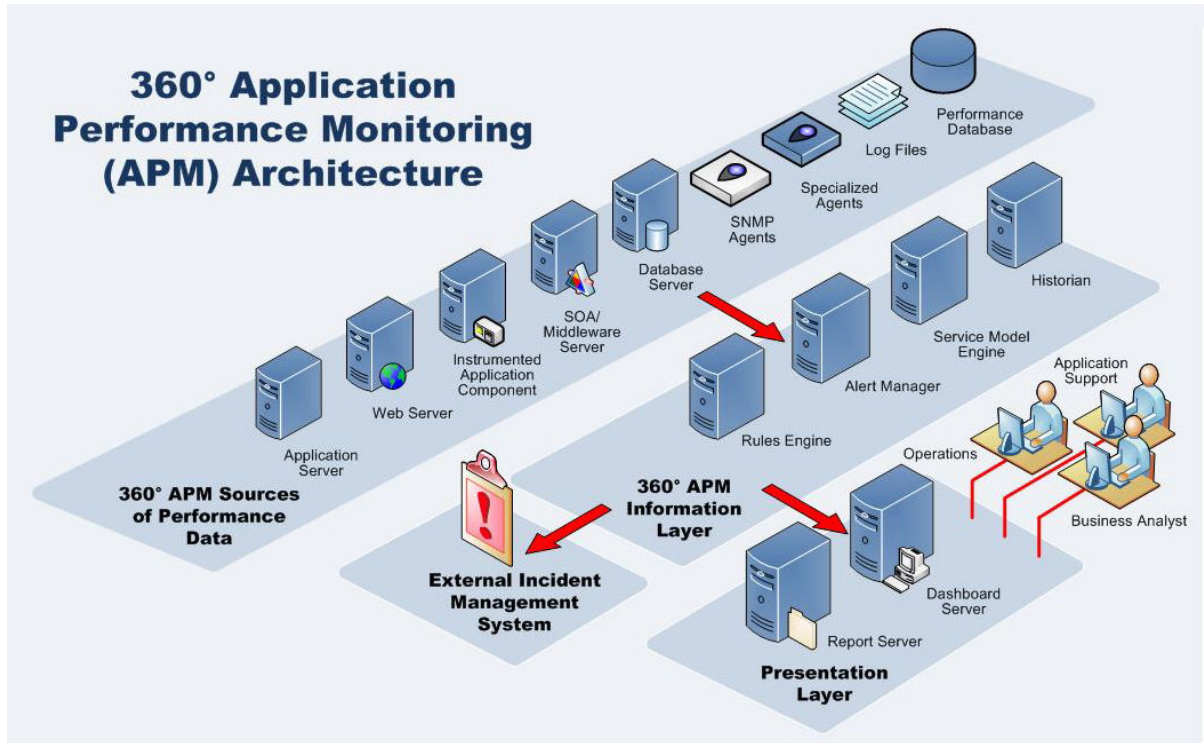
It sounds like a simple goal but it can become quite complex when you try to encompass the many varying applications of interest, their different development environments, operating systems and dependent software components.

Business Value of 360° APM

The business value of APM will vary based on the type of customer that is purchasing the solution. Most APM interest exists in enterprise environments of all sizes. For those organizations, business units that rely on critical applications that drive revenue or internal efficiencies cannot afford to experience application performance issues or application failures. If those issues occur, the business experiences a revenue impact or productivity hit that can typically be quantified. APM provides value to the application engineering teams as well as the operational organizations that are tasked with managing and ensuring that the applications are performing. Offering a 360° APM solution provides businesses quick insight into the exact problem, and can allow the incident management teams to troubleshoot issues before they affect the customers. The greatest business value of 360° APM will be for applications that directly impact revenue such as online product ordering, financial transactions or mission critical services like voice or video application services where productivity and performance is critical. 360° APM can also help ensure that service level agreements (SLAs) are not violated and help reduce or eliminate the penalties associated with SLA violations.

Architectural Requirements for 360° APM

The following are some of the architectural requirements to consider when searching for a 360° APM solution.



- **Access to All Data**

While in most cases, an enterprise will have packaged applications that can benefit from specialized agents or passive metrics specifically designed to monitor those applications, the 360° APM solution should have access to all standard real-time and persistent data sources (see below) so that the more complex custom applications can be monitored.

- o Software Components – Custom applications often rely on software components like databases, application servers, ESBs, and message brokers. Visibility into key metrics from these components often highlights critical junctures in application process flow.
- o Log Files – Log files are often the main source of information for legacy applications. To truly understand critical application metrics, it is necessary to be able to read and parse log files on a real-time basis.
- o SNMP – This is the richest source of information for devices on which the application depends for optimal performance. This information may also come from a standard enterprise monitoring system, but often it is desirable to gather these metrics separately for application monitoring purposes. Support

should be available for querying SNMP (gets), subscribing to SNMP traps, as well as setting SNMP traps.

- o JMX – JMX is becoming widely used for instrumenting custom Java applications. This creates a much more flexible and standardized way of reporting custom application status than can be derived from log files, but it also provides an interface to allow remedial measures to be injected into the application at run-time. JMX is also important because most of the commercial Java-based software components like application servers and messaging middleware have a JMX interface to determine component status.
- o WMI – WMI is the equivalent of JMX for .NET applications. Access to this information provides a standard way to monitor and manage custom .NET applications as well as other Microsoft applications like SQL Server, Exchange and SharePoint.
- o JMS – Some custom Java applications as well as software components use JMS to send point-to-point or multi-cast messages as a transport for application metrics.
- o SQL – Many custom applications store metrics into a relational database like SQL. SQL is also the main information gateway for many software components like BPM platforms, as well as for many APM point solutions like packet analyzers or web analytics.
- o CEP – In some cases, effective APM may require performing time-based correlations of streaming data to determine if there is an issue. If this is the case, the APM solution should provide a Complex Event Processing (CEP) engine or interface to third-party CEP vendors.
- **Data Calculations and Service Model Definition**

The ultimate goal is to be able to accurately describe how underlying dependent software and hardware components affect application performance. This requires a hierarchical data model – named here the “Service Model” – that describes low-level metrics, calculations that need to be performed on those metrics, and

component dependencies. In most custom applications, this data model cannot be automatically discovered, so the APM solution must provide an easy way to create and maintain this Service Model.

- **History**

The APM solution must be able to archive any necessary raw and aggregated data which would describe the history of application performance. In this way, analysis can be done that shows current status in context with how applications and components performed in the past. The history also allows for the creation of rules that signal when the application is performing outside of normal bounds when compared to the average performance during a particular time period in the past.

- **Rules Engine**

A rules engine is necessary to provide the definition of an automated response to a detected anomaly. The rules engine should be able to access any metric in the Service Model as well as Historical data for that metric, and initiate automated behaviors based on threshold values. Thresholds should be variable at runtime without having to re-deploy rules definitions. This helps give control to reduce noise at peak times of incident activity, or when rules are no longer valid because of deployment or testing activities. Automated behaviors should address both incident notification such as email or input to an incident management system, and self-healing commands such as activating system scripts, sending JMS messages, or invoking WMI or JMX methods to alter application behavior.

- **Alert Management**

When problems happen that cannot be automatically detected and repaired, it is critical to be able to understand the history of alerts. Answers to questions such as 'when did the event occur?,' 'how long did it last?,' 'how many times did it occur?,' and 'did anyone else see or act on this event?' are all key to the resolution. This information, combined with drilldown access to current component-level metrics, is critical to rapid analysis. The APM solution should maintain persistent alert state, and allow for presentation and management of this information in an efficient and customizable way.

- **Scalability and Support for Distributed Architectures**

There are four main areas which should be analyzed to determine whether an APM solution can scale as needed. They include Data Access, Data Processing, Data History, and Data Presentation.

Data Access – In many cases, the performance metrics need to be retrieved from distributed locations in a WAN. The APM solution should be able to gather the appropriate metrics and provide data reduction where possible so that the monitoring information delivery doesn't become an unmanageable consumer of network bandwidth.

Data Processing – The APM solution should have ways to scale out the processing load of a large quantity of data and the corresponding data calculations, rules execution and data aggregations that are necessary to determine performance and detect anomalies.

History – APM systems can have a large volume of time-stamped, raw data and aggregated performance metrics that must be stored to do comparative analysis. The history systems of an APM solution must scale for the necessary quantity of data and any other data management needs.

Data Presentation – The APM solution needs to be scalable to the full number of users who require access to the performance data.

- **Flexible Real-time Dashboards and Reporting**

The APM solution should have a completely flexible and customizable way to present data. This involves the typical tables, charts and indicators that are common in data analysis. But there are more complex requirements with APM that involve, for example, showing application performance with a geographic reference or sub-components in a logical reference to process flow. These interfaces need to be very interactive so that data can be efficiently and progressively disclosed. There also needs to be support for role-based visualizations both for security reasons and to provide relevant content to the right people.

- **Ability to Connect With Incident Management Systems**

APM solutions are typically problem management systems. They can provide a means for generating top-level “incidents,” yet they also maintain the level of detailed information which would lead to the analysis of the actual cause of the incident (problem management), initiate repair, and prevent it from reoccurring. In most large organizations, there is some existing incident management system that the APM solution should be able to communicate with to initiate the workflow necessary for a particular incident reported from the APM solution.

Best Practices

The Reality:

The difficult part about properly managing complex distributed applications – including custom applications – is that it often requires a development expert to determine the key metrics which define availability and performance levels. The reality is that the key metrics cannot be automatically discovered. When the application is changed and thus modifies critical business process flow or uses new process components, the application service model needs to be updated to reflect those changes. Advancements in SOA and monitoring capabilities have started to allow the automatic discovery of business process flow and key performance metrics. However many initiatives need to come together to make it possible to fully automate a 360° view through a SOA.

- Evolution of available SOA with built-in performance metrics and standardized means to discover process flow
- Enterprise adoption of those architectures
- Migration of legacy applications
- Adoption of performance metric delivery standards by complex packaged applications

These initiatives may eventually come about, but in the meantime, it is not an option for many organizations to simply ignore the performance management of critical applications just because change is difficult to manage. The key is to standardize on a delivery platform for application performance management to reduce the tremendous costs and waste involved with re-creating and managing different solutions within every application development and support team. Such disparate solutions further complicate

the goal of a 360° view because they cannot integrate and provide a single interface to the end-to-end performance of applications.

In addition to getting pressure to reduce the number of trouble tickets and MTTR, application support teams are also getting pressure to provide line-of-business stakeholders with visibility into performance management as it relates to business objectives. They also have pressure from the CIO to manage toward ITIL standards for Incident Management, Availability, Capacity and Service Level Management.

Where to Start:

Create a team. First a team must be created that is responsible for application performance management. In many organizations, this team is derived from the development departments and in others it comes out of IT operations. In either case, the team must be capable of gathering information from the various application development and support teams, and understanding what sort of tools and practices they currently use and how to capture performance management requirements for the applications they are responsible for.

Discover sources of metrics. Research critical applications and discover the source of metrics necessary to capture application performance. Be sure your APM solution can handle these sources of data in an efficient manner.

Discover current tool sets. There can be a lot of pushback from various application support teams who are vested in their current solutions. Be sure the APM solution can access data from these current solutions to allay fears of change. Also check that the APM solution can integrate or drill down to application analysis tools provided by those solutions to give these teams the best of both worlds.

Start small. Once you are sure that the APM solution can scale to the 360° requirements, determine a fixed set of high visibility, critical applications to construct a proof of concept. Determine your audience and key critical analysis paths they use for problem analysis. For application support users, make sure that the top-level views navigate to detailed analysis data in the same way they would actually solve relevant problems. For line-of-business users, present the data in ways that help them determine

how application performance is related to business objectives. Help them answer questions such as “Are these performance issues related to high-risk applications?”, “Have we delivered on promised SLAs?”, and “Where does it look like we need to invest in better performance?”.

Build in application lifecycle procedures. Development should have standard built-in procedures for instrumenting custom applications and providing new definitions of service models. If the procedures are not built into the lifecycle, the quality of performance management support will deteriorate and become unreliable.

The Result:

Many organizations are successfully delivering APM solutions that were not feasible even a few years ago. This is because of advancements in technologies such as standardized application instrumentation, standardized instrumentation of packaged software components and middleware architectures, real-time analytic engines, real-time data archival, and Web 2.0 presentation technologies. APM platforms that take advantage of these technology advancements are leading the way in providing robust, cost-effective ways of solving application performance problems, problems that can no longer be ignored without great risk.

About SL Corporation

Over the past 24 years, SL Corporation has become the most knowledgeable and responsive provider of real-time monitoring, analytics, and visibility solutions. SL’s flagship product, RTView, addresses a broad spectrum of operational visibility challenges spanning 360° application performance management (APM), business activity monitoring (BAM), and component-level infrastructure monitoring. RTView also has become the de facto standard for extending the visualization of complex event processing (CEP) engines, TIBCO messaging middleware, Oracle Coherence data grids, and custom applications. SL’s exclusive focus on real-time visibility solutions, commitment to customer success, and partner-centric culture are why thousands of industry leaders have chosen to work with SL to support their most critical applications and businesses. SL Corporation can be reached at +1 415-927-8400 or on the web at www.sl.com.



Contact Information

SL Corporation
240 Tamal Vista Blvd.
Corte Madera, CA 94925
+1 415-927-8400
info@sl.com

For more information regarding RTView for TIBCO, please visit:

www.sl.com