



# APPLICATION PERFORMANCE MONITORING

## PRACTICAL WAYS TO MONITOR THE END USER EXPERIENCE

WHITE PAPER

Performance of key applications is a critical item to monitor in many IT environments where users depend on those applications to do their jobs efficiently. Application environments can be large, diverse, and geographically dispersed even within a single organization, thus resulting in challenges to an IT team trying to monitor them in a holistic manner. Furthermore, performance problems with mission-critical applications can directly impact the health of a business and its bottom line. This paper describes these challenges and provides some practical methods of efficiently monitoring application performance.

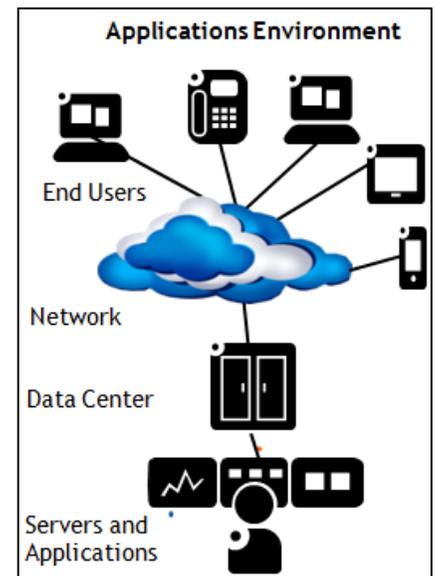
# TABLE OF CONTENTS

- Characteristics of Modern Application Environments..... 3
- Performance Monitoring Challenges..... 4
- Application Performance Monitoring Methods ..... 5
  - Continuous, Comprehensive Performance Monitoring is Key..... 5
  - Monitor Actual End User Application Experience ..... 5
  - Use Synthetic Transactions to Provide a More Complete View ..... 5
  - Monitor Servers and Networks that Support the Key Applications ..... 6
  - Get a Comprehensive End to End View of Performance ..... 6
  - Monitor Different Groups with Different SLAs ..... 6
  - Use a Robust Performance Management System to Make your Job Easier ..... 7
- Alcatel-Lucent VitalSuite® Applications Performance Management System ..... 7
  - A Comprehensive Performance Management Suite ..... 7
  - VitalSuite® Apps - Powerful Application Performance Management ..... 8
  - Data Collection from VitalAgent Software ..... 8
  - Analysis and Reporting from the VitalSuite® Apps Server ..... 10
  - Minimal Footprint ..... 10
  - Summary of VitalSuite Components for Monitoring Application Performance ..... 10
- Putting it all Together ..... 11
- Acronyms..... 11
- For More Information ..... 11

## CHARACTERISTICS OF MODERN APPLICATION ENVIRONMENTS

Consider an organization, where the environment for key applications may have the following attributes:

- Different user groups within the organization, with different sets of key applications. While a few applications may be common to everyone (IM and data sharing, email, time tracking, etc.) others are more specialized and used heavily by specific groups (order entry systems, sales tracking, accounting, source code control, etc.)
- Diverse application delivery types. Applications may range from traditional client/server model, to web based, to cloud based, using traditional servers or virtual servers, and various hybrids of these types.
- Diverse user locations and access types. Some users may be in large groups in single locations on a LAN or WLAN, while others may be widely distributed in smaller locations, accessing the application via xDSL, WAN or VPN. Global organizations may sometimes have users in various countries accessing the same applications.
- Large numbers of users. Application user communities may range from a handful of users for specialized applications, to thousands of users for more general applications.
- Different end user devices. Some users may be accessing applications from a modern desktop, laptop, tablet, or smartphone, while others are still using older models with different platform specifications. Some organizations may be experimenting with Bring Your Own Device (BYOD) which brings further diversity to this issue.
- Different client web browsers and OS platforms. Even for common end user devices, differences in OS and Browser release can create major differences in Application Performance.
- Diverse Application versions. While it is often a goal to update applications to common versions throughout an organization, in practice upgrades can be slow. Therefore multiple application versions may be deployed at any given time.
- Prime Time usage periods. Many organizations experience vast differences in application usage between high volume periods (such as during a work day) and low volume periods (such as at night or on weekends). This load behavior can be further complicated by large backup jobs that may run during the night.
- Critical and non-critical applications. Applications may range from those that are business-critical, to those that are useful but not critical. Sometimes these attributes can be completely different for different user groups, depending on their job functions.



This paper focuses on Application Deployment Environments. Other environments (such as Application Software Development) may have other more specialized performance monitoring needs that do not necessarily function well in production deployments.

# PERFORMANCE MONITORING CHALLENGES

IT personnel tasked with monitoring such an environment face additional challenges

- Limited resources for monitoring. IT teams are often thinly staffed and are expected to monitor very large environments. Hence they need to find ways to enhance productivity and efficiency.
- Limited resources for infrastructure improvement. IT teams may be constrained in improvements they can make in network and application infrastructure items.
- Different expectations from user groups. Some user groups may require different levels of service compared to other groups, even for the same applications. Even the tasks of identifying different user groups, understanding their application performance needs, and communicating with those groups can be time consuming.
- The need to be proactive. Organizations that are constantly in reactive mode, responding to user complaints and troubleshooting, may find it difficult to get ahead of the situation and identify and resolve problems before they affect users.
- Monitoring the actual end user experience. In many cases, IT teams have access to information about certain infrastructure components, but do not have a good view of what end users are actually experiencing.
- Getting the right information. IT teams need systems that provide the right kind of application performance information, in the right context, to support better decision making.
- Avoiding the blame game. IT organizations often split responsibilities among different support teams, such as Network, Server, Desktop, and Applications. Alternatively, they may split responsibilities according to geographical teams. However modern IT environments involve dependencies among the various components in order to provide Application Service. Hence there is a need to identify both end-to-end performance issues, and performance issues with specific components (such as desktop, network, or servers). This task is crucial in order to keep the focus on problem resolution and improvement, rather than on discussions of which group should be responsible.
- Efficient performance monitoring. Organizations cannot afford to have performance monitoring systems and activities that are so burdensome or data intensive that they slow down the very applications whose performance they are monitoring.
- Seeing the forest as well as the trees. IT teams often deal with information that may describe part of an issue but not all of it. Consider the following examples
  - Users A and B are experiencing poor response time for application A
    - IT personnel starting with this limited amount of information can potentially waste a lot of valuable time trying to resolve the problem.
  - Users A and B, as well as all other users at location L are experiencing poor response time for application A, which is provided on server S, and this has performance has occurred from 10 AM to noon for the last week, gradually getting worse over that time. Location M users who also access application A from server S have experienced good response times.
    - IT personnel starting with this information can more quickly focus their efforts on infrastructure items that are common to location L (such as Network connections and bandwidth, desktops deployed there, etc) rather than wasting time trying various solutions that are unlikely to resolve the issue.

As the above example shows, the right kind of information, in the right context, can help IT teams make decisions better and faster on where to spend their limited resources to resolve the application performance problem.

## APPLICATION PERFORMANCE MONITORING METHODS

### Continuous, Comprehensive Performance Monitoring is Key

Some organizations rely solely on a troubleshooting approach, and only deploy tools that support this mode of operation. Other organizations may monitor faults but not performance. However this instantly puts them in react mode – when a customer calls and complains (if they complain; many will not) then they investigate and resolve. There are many situations where faults may not be present, but the application is nevertheless totally unusable because of poor performance.

A better approach is to deploy systems that \*continuously monitor\* application performance throughout the organization.

### Monitor Actual End User Application Experience

By deploying a system that can efficiently monitor the actual end user application experience of dozens, hundreds, or thousands of end users, then IT teams get a more complete picture about what end users are actually seeing. Typical Key Performance Indicators (KPIs) should include

- Number of transactions the user started for each monitored application
- Response time of those transactions
- Successful versus failed or timed out transactions
- What group or domain the user belongs to
- Which servers were being accessed
- Application availability over time

Some organizations may choose to deploy monitoring software on every end user desktop in the organization, which provides the most complete view of end user performance. Other organizations may instead choose to deploy it on selected desktops (several at each location, or in each user group, for example). This approach provides a more limited, but still useful, view of end user application performance.

### Use Synthetic Transactions to Provide a More Complete View

End user experience monitoring is necessary but not always sufficient. Its primary limitation is that during periods when no users are active, there is no performance data about that application.

Consider the following scenario: Application A becomes extremely slow at night while no users are accessing it due to file system low on space. No alarms are raised because the IT team relies exclusively on user transaction monitoring. At 8 AM the next morning, dozens of users call the help desk to complain that they cannot access Application A.

Synthetic transaction monitoring provides a method of periodically testing Application A automatically, regardless of the level of actual end user activity. A performance management system with this feature could have provided early warning to the IT team that Application A response time was too slow, and they would have time to correct the problem before the user community needed their access to it during prime time hours.

## **Monitor Servers and Networks that Support the Key Applications**

Since the servers and networks typically have a key role in Application Performance, monitoring those components is important. Furthermore, certain applications may produce useful performance monitoring data that can only be obtained via monitoring the Application server.

Network performance monitoring is also important, particularly in situations where VLANs or QoS features are configured to segregate and treat certain Applications differently from other network traffic.

## **Get a Comprehensive End to End View of Performance**

The previous discussion indicates that a complete understanding the performance of Key Applications requires performance data from many different sources.

A truly useful performance management system will provide comprehensive integrated views of Application, Server, and Network Performance, obtained from a variety of data sources.

Furthermore, IT personnel need performance data from different timeframes in order to understand the problem. To illustrate this point, consider a performance problem where the IT person has access to very detailed \*current\* performance data, but no information about performance of the same applications in previous days. Without the historical context, it would be more likely to come up with a workaround or fix that doesn't solve the root cause of the performance problem; for example, reconfiguring a router when the actual problem is servers that are periodically overloaded due to too much user load at key times.

## **Monitor Different Groups with Different SLAs**

While it may be a noble goal to monitor all users with the same Service Level Agreement (SLA), in practice this is not always feasible or even desirable. Consider for example a key application that used daily by a large group in USA, with specific SLA requirements (1500 msec response time for example). This same application is used infrequently by a few users in France, who commonly experience 4000 msec and higher response times. If you take the simplistic approach of lumping both sets of users together, the IT team members will often see SLA violations when the French team uses the application. Over time they are likely to get used to this performance, and become complacent if SLA violations started to occur in the USA group.

A better approach is to create two user groups, and manage their SLAs separately. For example, establish an SLA for the USA group at 1500 msec for 90<sup>th</sup> percentile response time, and a French group SLA at 4000 msec for 90<sup>th</sup> percentile response time. This can reflect the reality that the organization is not yet willing to invest in upgraded WAN infrastructure to

bring the French SLA in line with the SLA for the USA group. But by monitoring as two separate SLAs the IT team can have the data it needs both for dealing with real End User performance issues, and for planning future enhancements (such as justifying a WAN upgrade, or another Application Server deployment, or other performance solution). A key point in this discussion is the need for tools that automate these SLA monitoring tasks.

## **Use a Robust Performance Management System to Make your Job Easier**

IT Application teams don't have time to waste logging in to 8 different tools and multiple different server consoles to try to piece together a useful picture of application performance. They need a performance management system with key features to help with the task:

- Provide a complete view of Application performance across the organization
- Easily scale up to hundreds or thousands of end users and dozens of key applications
- Provide GUI screens, reports, and alarms with useful information about Application Performance –
  - Over multiple time periods – both current information and historical trends
  - For different groups of Users
  - For different Applications and Application Servers
  - Both high level and detailed reports
  - Both aggregated data about groups of users, servers, and applications; and specific data about individual users, servers, and applications.
- Provide flexible thresholds and alerts to manage SLAs efficiently
- Provide an easy-to-use web based GUI that makes it simple to deploy and use the system. Performance monitoring systems that are too difficult to use will rarely get used by the people who need the information most.
- Provide power and flexibility for adding new monitored application types, new user groups, new reports, etc.
- Provide easy installation and default configurations so IT teams can be quickly productive without investing in a massive configuration effort
- Provide a method for automatically deploying new Application definitions to end user desktops throughout the organization.

## **ALCATEL-LUCENT VITALSUITE® APPLICATIONS PERFORMANCE MANAGEMENT SYSTEM**

### **A Comprehensive Performance Management Suite**

The Alcatel-Lucent VitalSuite® Performance Management System provides comprehensive performance management of today's multi-technology, multi-organization, multi-application, multi-vendor network and application environments. Key components of the suite include:

- VitalSuite Apps – Application Performance Management component
- VitalSuite Net – Network Performance Management component
- VitalSuite Flow – advanced collection and analysis of Netflow records directly from flow-enabled network routers

- VitalSuite ART – Advanced Reporting Tool — wizard-based custom reporting. An easy-to-use yet powerful Web 2.0 GUI interface

Each of the VitalSuite components are designed to work together, sharing common components such as:

- An easy-to-use yet powerful Web 2.0 GUI interface featuring a truly unique 3D maps
- Network Discovery of both ipv4 and ipv6 devices, data collection, and analysis features to provide useful performance data when you need it.
- Powerful threshold and alarm features
- Built-in reports as well as tools for custom reporting ready to be emailed
- Common Server and DBMS platforms
- A scalable architecture allowing customers to start small and grow as their needs expand, up to the largest networks and application environments.
- Out-of-the-box application definitions and configurations
- Flexible tools for adding new users, new application definitions, and new device types

VitalSuite components are designed around the philosophy that “Easy things are easy, and hard things are possible.” This makes it easy to install the system and start getting useful results with minimal up-front configuration effort, while still providing the powerful configuration options and flexibility needed to adapt to different applications environments.

### **VitalSuite® Apps - Powerful Application Performance Management**

VitalSuite Apps is designed to monitor Application performance as seen by actual end users. Furthermore, because it shares the scalable VitalSuite architecture, it is designed to scale up to monitoring hundreds or thousands of end user devices via a single Web 2.0 GUI portal.

### **Data Collection from VitalAgent Software**

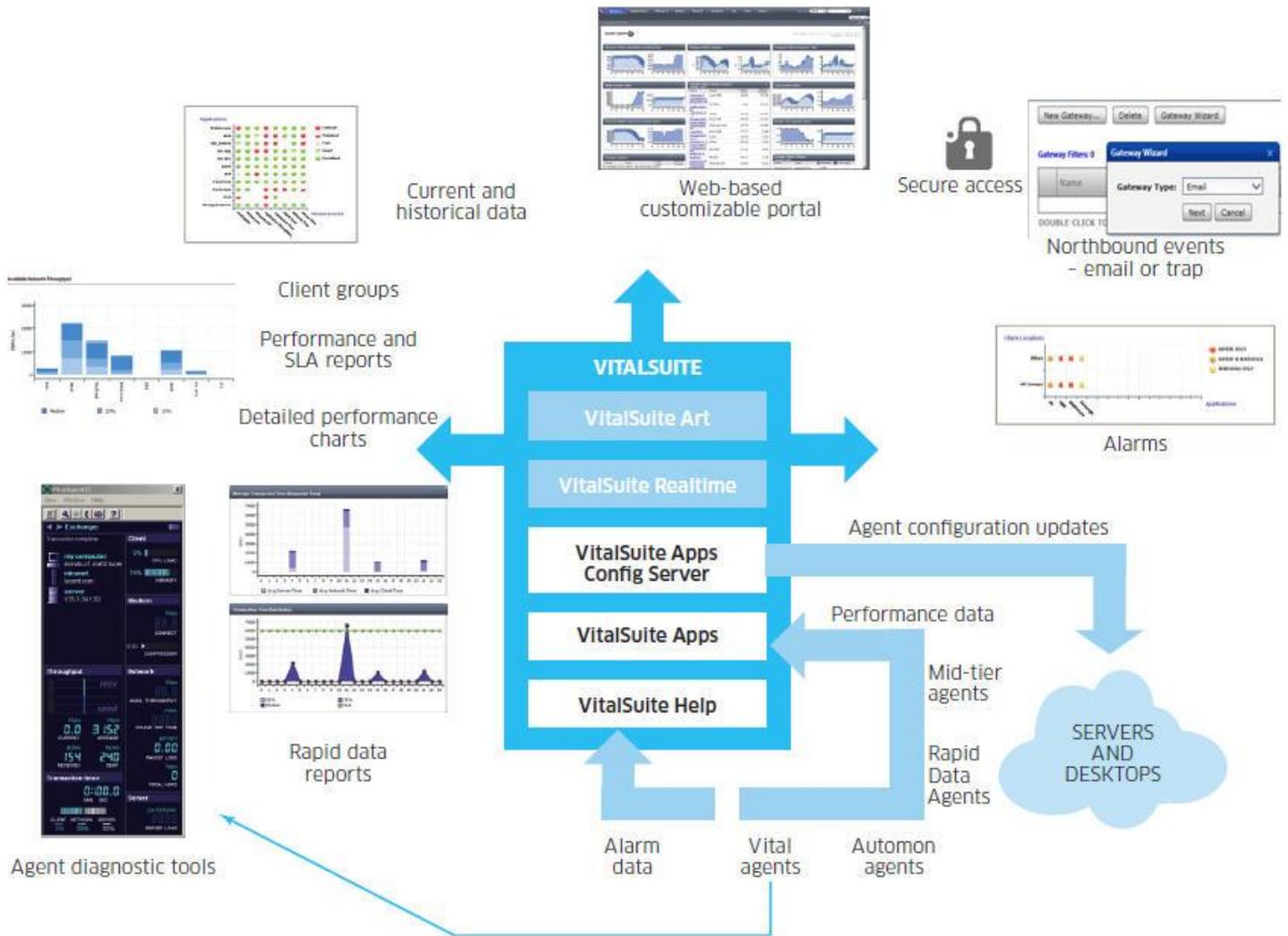
VitalSuite Apps collects performance data via a small light-weight software Agent that is installed on user desktops and selected Application Servers.

This software agent provides several key functions:

- Accepts new and updated Application definitions from the VitalSuite Apps Configuration Server.
- Analyzes TCP/IP packets to look for key Applications that are to be monitored for this client.
- When a key Application is recognized, the Agent monitors the transaction time, thereby providing statistics on
  - Number of transactions
  - Transaction response time due to Client, Network, and Server
  - Transactions failures and timeouts
- Periodically uploads this application performance data to the VitalSuite Apps server, which stores, analyzes, and aggregates Application performance data from all Agents throughout the network. Uploads can be hourly or on 5 minute basis (Rapid Data Agents).
- Monitors on-occurrence errors and application failures, and sends these events immediately to the VitalSuite Help module for immediate event analysis and alarming

- Provides a set of useful debugging tools so that a help desk operator can get useful performance troubleshooting data about the user's desktop, without having to take the time to login to that desktop directly
  - Provides handy debugging tools, including packet capture, and remote queries (ping, dns query, traceroute, etc.)

**VitalSuite Apps Data Collection, Analysis, and Reporting**



In addition, there are several different types of Agents available for specific purposes:

- An Agent with a desktop GUI so advanced users can view realtime application performance data themselves (for example, clicking on a web page, and watching the resulting performance of the response in the VitalAgent GUI).
- A mid-tier server Agent, for analyzing specific server performance impact when an application is implemented using front-end and back-end servers (for example a front end GUI server talking to a back end DBMS server).
- An Automon agent, for performing synthetic transactions on a periodic basis, and reporting those results to the VitalSuite Apps server.

- Rapid Data Agents which report application performance results every few minutes for analysis and reporting.

## Analysis and Reporting from the VitalSuite® Apps Server

The VitalSuite Apps server collects performance data from the VitalAgents deployed throughout the network, including:

- Periodic performance data about end user application transactions and response times. This data is used to show reports including
  - SLA reports for Applications, covering Availability and Performance
  - Application usage reports showing how the volume of transactions varies over time
  - Percentile performance reports with 50<sup>th</sup>, 75<sup>th</sup>, and 90<sup>th</sup> percentile performance. This feature is particularly useful in large networks, where it gives a sense of whether outlier data are widespread or isolated, and hence how important those values are.
  - Both detailed reports (per Agent and per Server) and high level reports (many Agents and many Servers)
  - Current and Historical data views via the Web 2.0 GUI (no need to format a report each time).
- On-occurrence application errors, used for VitalHelp alarming and listing of application problems (such as web server not found, application timeout, etc.)

The system provides powerful grouping functions so IT personnel can easily monitor different SLAs for different groups of users, different Applications, and different Application Servers.

## Minimal Footprint

One important point to consider is the impact of the performance monitoring system on the network itself. A system that produces too much network traffic or client load can exacerbate the very performance that it is trying to identify. A system that requires large numbers of hardware probes deployed may require too much effort to cover the entire global network. This is one key reason that application monitoring approaches that may be useful in Lab or software development environments often don't scale well to production deployments. The Alcatel-Lucent VitalSuite® Performance Management System is specifically designed for efficient data collection to minimize network polling traffic, and to use minimal resources on end user desktops. The system is well suited for large scale production deployments of thousands of devices and end user desktops.

## Summary of VitalSuite Components for Monitoring Application Performance

### Examples of Monitored Objects and Metrics

VitalSuite Component	Monitored Object Example	Metric Examples
VitalSuite Apps	Application Performance seen by end user	Response Time, Failed Transactions
VitalSuite Automon Agent	Synthetic Application Transactions	Response Time, Failed transactions
VitalSuite Net	Apache Web Server Performance	Number of connections

VitalSuite Net	VMWare or Citrix Host Performance	CPU, Memory, Disk Utilization
VitalSuite Flow	Application Flows between endpoints	Volume of Application Traffic – octets and flows

## PUTTING IT ALL TOGETHER

Managing the performance of Application Environments is an increasingly difficult task. Relevant data may need to come from a variety of different sources, and there is often a need to reconcile what an end user is saying, against what is actually happening in the network or on their desktop. The task is further complicated by constraints of time, money, and talent that many organizations experience when doing this work.

Forward looking IT organizations also find it imperative to get ahead of the curve, becoming proactive about monitoring key applications so they can identify and correct performance problems in many cases before they impact users. Furthermore, these organizations carefully monitor performance against SLAs which reflect their business needs and goals.

A product such as the VitalSuite Applications Performance Management System can give IT organizations the tool they need to efficiently monitor their key applications, desktops, and servers, keeping their end users satisfied and productive.

## ACRONYMS

BYOD Bring Your Own Device  
IT Information Technology  
KPI Key Performance Indicator  
SLA Service Level Agreement

## FOR MORE INFORMATION

[www.alcatel-lucent.com/vital](http://www.alcatel-lucent.com/vital)



**VitalSuite**