WHITE PAPER

From Downgraded to Dead: Monitoring Application Performance

By Nick Cavalancia

IPSWITCH™

TABLE OF CONTENTS

IPSWITCH™

Your Microsoft applications act as the foundation for nearly all other services within the organization.

While organizations today focus on business continuity from a disaster standpoint, the likelihood of business-crippling disasters like fires, floods, and hurricanes is low. But the idea of working to keep your business continually running from an applications perspective is a far greater challenge. On any given day, hardware can fail, storage can run out, systems can reboot, and users can lose connectivity. The idea of continuity takes on a bit of a different medium when you are thinking about the continuity of your applications.

Continuity certainly can be about whether an application is running or not, but given the complexities of multi-tiered applications, the concept of the application becomes a bit muddled and generalized from, say, a user's perspective. And to add to the continuity concept, there's something to be said about an underperforming application and its' negative impact on the business.

Whether downgraded or dead, neither level of application performance is acceptable.

Loss of an application, even for a short period of time can have a material impact on the organization. According to a recent EMC survey, 49% of organizations had experienced an unplanned systems downtime within the last year, with the average time to recover ranging from 2 to 5 hours costing thousands of dollars per minute.

And it's not just downtime that impacts an organization; 73% of organizations saw decreased application performance have an impact on employee productivity and customer satisfaction—both impacting the bottom line.

Your Microsoft applications act as the foundation for nearly all other services within the organization. Active Directory, Exchange, SQL Server, and SharePoint, among others, represent the crux of services that keep most businesses running today, making the performance of these applications critical.

So what's the best way to approach monitoring your Microsoft applications to ensure business continuity, as well as application performance?

In this whitepaper, we'll address this issue of monitoring application performance by looking at the problem from three perspectives: why monitor, what to monitor, and how to monitor.

WHY MONITOR?

IT is not only expected to provide services and applications to their customers; they are also tasked with ensuring the availability of those same services and applications. Now, every company, even those without formal monitoring in place, actually does have the most rudimentary form of monitoring—their users. When an application is responding slowly or not at all, IT will receive a call. This aligns with the most fundamental thinking around monitoring—to simply to know whether applications are running or not.

IT is not only expected to provide services and applications to their customers; they are also tasked with ensuring the availability of those same services and applications.

But when you consider how using this "red light/green light" approach doesn't provide IT with any real actionable detail, and only does indicate an issue exists after users have been impacted, it's evident critical applications require a true form of application monitoring for a few IT-centric reasons:

- **IT Needs to Be Proactive**—Monitoring of critical applications can provide IT notifications of leading indicators that, if left unattended, could spell disaster for an application.

- **IT Needs Detail**—Identifying an issue before it becomes a problem impacting users is all well and fine, but without having details like which service failed, which drive has no remaining storage, or which network segment is solely impacted can make the difference between solving the problem in minutes or hours.

- **Applications Can Be Complex** – Multi-tiered applications (e.g. those with front-end services, back-end databases, and authentication services all tied to a central application service) can't simply be judged by whether they are up or down. IT needs to know what

specifically in the architecture has failed, causing (potentially) the entire application to fail.

The bottom line is an application is only valuable to the organization if it's running… and running well. Having every part of an application in view and knowing not just if it has a problem, but proactively being able to tell an application is going to have one is critical. This is certainly one reason to monitor, but is it the only reason to monitor applications?

More than just IT needs to know

In a multi-tired application, the line of business pros only care that they are in business.

The reality is it's not just IT that is concerned about application performance and availability. Many other roles within an organization are just as invested, if not more. Line of business professionals care about that the applications are running and supporting their respective businesses, as do business stakeholders and cross-functional teams—each having their own reasons for doing so, their own focus within the application, and their own needed levels of detail to react appropriately.

For example, in a multi-tired application, the line of business pros only care that they are in business, so for them, it's more along the lines of the "red/green light" previously mentioned. They wouldn't be concerned that a database server processor is at 100%—that is, until it affects the overall application. However, the system admins and database admins are focused on only those components of the application they are tasked with managing. So, they will be proactively concerned about the performance of the database server and act appropriately.

Because of so many varying needs around knowing whether, how, and what parts of an application are running, application monitoring is necessary to quickly provide information to the affected or interested parties as necessary on everything from application availability, to state changes on specific applications or components, to precursors to major application issues.

In addition to information necessary parties about whether and how an application is performing, application monitoring also helps establish that IT is meeting agreed upon service levels.

Proactively approach service levels

The main purpose of service level agreement (SLA) reporting is to clearly document the levels of service you are providing to your customers, stakeholders and/or management. An SLA is really just an established expectation around application availability. An while the higher the availability expected, the more likely the application will be that much more complex (requiring monitoring beyond that of just up/down methodologies), the basic applications can require just as much monitoring.

The SLA on even the simplest application running on a single server needs to be well-defined so IT understands what is and isn't acceptable performance to the impacted members of the organization. Monitoring can actually help define which components of an application stack are critical by establishing the performance relationship between them. For example, if one or more discrete applications—like SQL Server or IIS—can be down without a parent/complex application going down, there may be a different SLA than where the parent application also fails.

In addition to establishing the SLA definitions, application monitoring can also provide the means to measure those SLAs.

In addition to establishing the SLA definitions, application monitoring can also provide the means to both measure those SLAs, as well as provide a much more accurate picture of the level of service you are providing.

To achieve awareness of whether your applications are running well, or at all, and to meet the service level agreements established, it's critical that you properly determine what should be monitored.

WHAT TO MONITOR?

It's simple enough to build a list of applications you need to make certain are running. You can easily find the list of services related to, say, Exchange or SQL Server fairly easily, tack on a few performance monitors around resource usage and you have yourself application monitoring, right?

In reality, application monitoring is far more than that.

When it comes to monitoring, you have an opportunity to comprehensively put everything under the monitoring "microscope" and have every part of an application scrutinized. You also have an obligation to meet the monitoring and reporting needs of those organization members with a stake in the performance of given applications as well.

So, how do you determine what to monitor?

Since no two environments are the same, it's impossible for this paper to outline exactly what to monitor. But to attempt to assist in guiding you to the right answer for your organization, follow these three steps.

A modern application is more than just a set of services on a server.

Step 1 – Define what you support

It's impossible to manage the unknown. Therefore, the fist step in providing optimal application performance in your Microsoft environment starts by establishing and maintaining an up-to-date inventory of both your network and server hardware and software assets, physical connectivity, and configurations to truly understand what you are responsible for supporting.

A modern application is more than just a set of services on a server; it's everything that makes up the delivery of those services—the network infrastructure, interfaces, web servers, application servers, physical and virtual servers hosting the application, OSes, authentication services, databases, and then the application itself. Because each part of the application delivery has the potential to impact the availability and performance of the application, you need to consider all of them when defining what should be monitored.

Step 2 – Use an end-to-end perspective

With the shift to a mobile workforce, the use of multi-tiered applications has grown. Monitoring requires incorporating more than one system to have a complete view of what's happening and where the problem lies that the user is experiencing. Thinking about how an application works—from the user to the data—and incorporating each facet of that process into your monitoring strategy will allow you to better identify issues leading up to impactful problems, as well as the root cause of problems within an otherwise complex mix of servers, services, and networks.

Step 3 – Be aware of dependencies

It's one thing to monitor a virtual server hosting a clustered application and have every aspect of it covered—from the virtual server, to the clustering services, to the application itself and the related storage components. But what about when that application needs to email something… and it's using Exchange to do so?

In addition to having knowledge of all that you support, and how your applications work end-to-end, you lastly need to be aware of and correlate any other systems on which the application in question depends upon. If you leave out dependencies as part of your monitoring strategy, you're in essence excluding a part of the application, despite its' use of a separate system.

You need to be aware of and correlate any other systems on which the application in question depends upon.

While for some it may be a simple issue like using Exchange for email, but for most of you the challenge will be how to identify these separate, but interrelated application components and assets, and to determine how to build a comprehensive monitoring view.

HOW TO MONITOR?

Even when you take the reasons why you need to monitor applications within your organization, and marry them with the specifics of what needs to be monitored, it won't always be evident how to join the two into an action plan. Should you monitor just the core Active Directory services to report an SLA to the AD admins? How should you define your CRM's uptime to your CEO?

It's not always going to be straightforward. So, this last section is about how to approach monitoring in an effort to protect the application, those invested in its success, and its users

Start with the user's perspective

Because some many applications today are too complex to monitor as individual systems, don't try (at least, not at first). Start by monitoring whether the application as a whole is up or down. You can determine an application's state based on monitoring only the critical components upon which its' performance depends. So, for example, if a database server, web server or domain controller is down, the application is also down.

Some third-party monitoring solutions even take on the role of the user and test beyond just counters and metrics and actually test out, say, a web shopping cart from start to finish as part of monitoring the application. But even if you only have native tools to accomplish your monitoring, you can still identify the equivalent critical components (so, the web server hosting the cart, the service that processes the credit card transactions, and the database back end that validates inventory) and monitor their availability to somewhat accomplish the same thing.

Generally, starting with the user's perspective is the correct starting point as applications get more complex. The only caveat to this is when the components are individually deployed applications themselves, such as Exchange. You can certainly still include is as part of the 10K foot view of your complex application, but it's important to also monitor the individual application and its' components to ensure it's performing according to its' own SLAs.

When a problem arises, only monitoring at the 10K foot view alone won't help identify root causes.

Monitor at multiple levels

When a problem arises, only monitoring at the 10K foot view alone won't help identify root causes, allow you to prove you are meeting application and component-specific SLAs, nor notify owners so problems can be quickly remedied.

Let's stick with the web-based shopping cart example to make the point. Monitoring at the 10K foot view, you'll be aware the SQL Server backend is no longer responding, but that's the level of detail you have. Those administering your SQL Server need a bit more information than just "it's down" and because the application is critical, they need that information as quickly as possible. Having multiple levels of monitoring, with an ability to logically drill down through your monitoring will speed up the remedial process.

Even if all you have is a set of PowerShell scripts pulling performance monitoring metrics and checking the current state of services for that 10K foot view, having a second script looking at lower level counters and metrics for just the SQL Server would provide its owners with more information, such as the processor pegging at 100% or the having no more disk space.

By building your monitoring strategy around the perspective of both the user, as well as the more detailed view of those supporting the applications, you'll not only be aware when something goes (or is about to go) wrong, as well as to provide the information needed to do something about it quickly.

KEEPING APPLICATIONS PERFORMING

We live in a world today where applications are all about integration, interoperation, and interdependency, increasing complexity. These intricacies have the potential to create a downtime domino effect. Monitoring application performance, if done correctly, does more than just proactively identify issues that would otherwise negatively impact the business; it can help define and establish meeting of SLAs, provide needed performance detail to business stakeholders, and when a problem arises, provide those addressing the issue with pertinent details.

We live in a world today where applications are all about integration, interoperation, and interdependency, increasing complexity.

Whether you take on this challenge using native tools or using a third-party application, by putting some level of application monitoring in place, you'll minimize your risk and maximize your application performance.

With nearly 20 years of enterprise IT experience, Nick Cavalancia is an accomplished consultant, speaker, trainer, writer, and columnist and has achieved certifications including MCSE, MCT, MCNE and MCNI. He has authored, co-authored and contributed to over a dozen books on Windows, Active Directory, Exchange and other Microsoft technologies. He has spoken at conferences such as the Microsoft Exchange Conference, TechEd, Exchange Connections, and on countless webinars and at tradeshows around the world.