

A White Paper

**Using Enhanced Application Monitoring to Achieve Peak SQL Server Performance**

***HEROIX***

Heroix  
165 Bay State Drive, Braintree, MA 02184 USA  
[www.heroix.com](http://www.heroix.com), [info@heroix.com](mailto:info@heroix.com)

# Improving SQL Server Performance through Better Application Monitoring

SQL Server is at the heart of many enterprises' most critical business processes, ranging from line of business production applications, to internal CRM and decision support systems, to customer-facing e-commerce and self-service web applications. Consequently, SQL Server availability and performance are high on IT's list of priorities.

Many challenges, however, face the IT administrator in charge of supporting SQL Server resources. For starters, application complexity is on the rise and IT infrastructures are becoming far more diverse. With complex applications working across heterogeneous infrastructures, monitoring applications and providing fast, accurate problem resolution are more difficult.

Additionally, IT departments are being asked to do more with far less in terms of budget and personnel. Few companies have ample IT budgets and staff to manage their increasing workload. As a result, tasks like application monitoring are not given the attention that they really need to pay significant dividends.

Finally, enterprises are becoming increasingly distributed and the needs of the end-user population are far more diverse both in location and in the technology, they are using. An IT user may be a business partner or a customer, accessing your databases via web interfaces from anywhere in the world. This makes user demands far less predictable and harder to manage.

Fortunately, delivering high performance and availability for SQL Server can be achieved even in light of these challenges if you have the right monitoring strategy and tools in place.

In the first section of this paper, Monitoring the SQL Server Environment, you will learn about how you can provide higher performance and availability for your SQL Server by knowing the key drivers behind SQL Server performance and availability. Armed with this knowledge, you will be able to focus your monitoring strategy and get better returns from the time you dedicate to SQL Server monitoring and management.

In the second section of this paper, Choosing a Monitoring Solution, you will learn what to look for in an application monitoring tool. By choosing the right tool, you can dramatically improve your monitoring strategy while greatly reducing the amount of dedicated personnel needed for application monitoring and management.

## Monitoring the SQL Server Environment

A focused monitoring strategy is essential to creating and maintaining a high performance and highly available SQL Server and application infrastructure. By accurately monitoring and tracking performance of SQL Server resources, you will be able to

- Rapidly locate problems at the hardware, server and application level and reduce the time and resources needed for supporting your SQL Server infrastructure

### **Monitoring the SQL Server Environment**

**Monitoring Memory**

**Monitoring Disk I/O**

**Monitoring CPU Utilization**

**Monitor Key SQL Server Indicators**

**Monitor Proactively**

### **Choosing a Monitoring Solution**

**Make multiplatform a must**

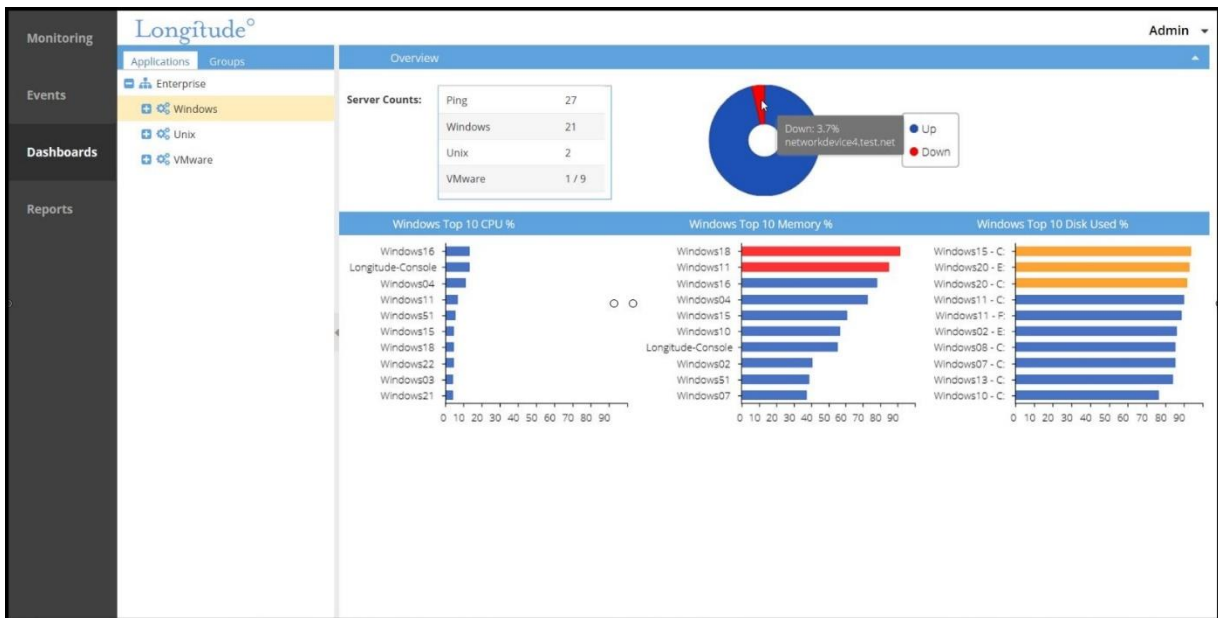
**Reduce the burden of proving SLA performance**

**Aid in threshold adjustment**

**Consider agentless monitoring**

- Improve end-user performance by determining the configurations that are most appropriate for your unique operating environment
- Evaluate usage patterns and plan for the best times to perform system upgrades or maintenance
- Improve capacity planning by looking beneath the surface of performance and availability problems to determine whether issues can be resolved by changing database and server configurations rather than purchasing new hardware
- Proactively troubleshoot performance problems before they reach end-users

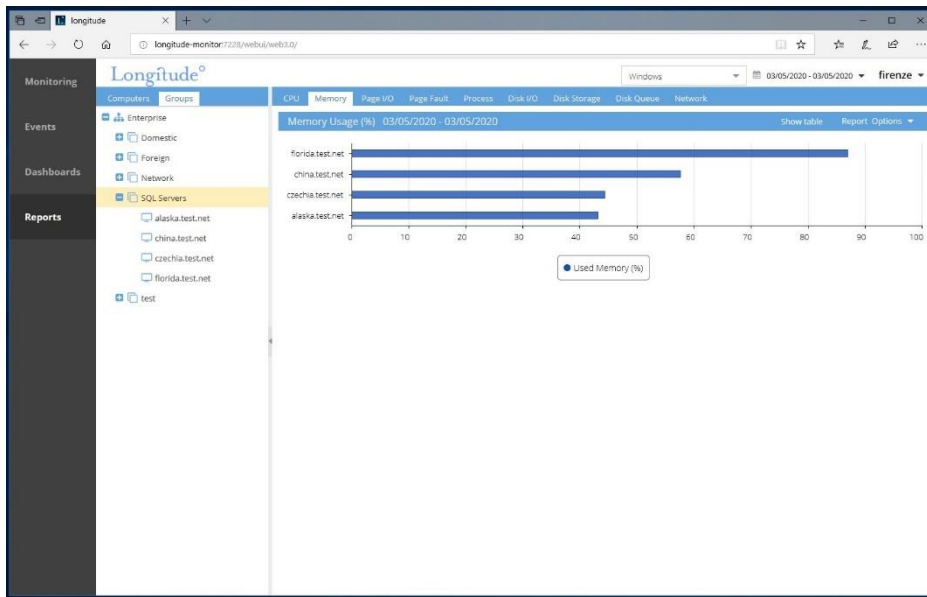
When it comes to monitoring SQL Server, there is an abundance of different sources for data. While it may be tempting to monitor everything, many administrators find themselves overwhelmed by the sheer volume of performance data available to them. Fortunately, most SQL Server performance issues can be detected based on just a handful of metrics, starting with memory, I/O and CPU, in that order. This is because a performance issue in memory can contribute to I/O and CPU bottlenecks, and so on. If you spot trouble in one area, you can then delve in to investigate further.



*A dashboard display can provide a quick snapshot of overall performance and indicate areas that need further investigation.*

## Monitoring Memory

The primary factor impacting most SQL Server performance problems is memory. It's important to monitor memory to ensure that the percentage of available memory stays above 20%. If this number falls below 20%, and users are experiencing performance problems, you can be sure that insufficient memory allocation is the culprit. High paging rates, the rate at which pages are read or written to the disk when the system can't find the page in memory, can also contribute to an I/O bottleneck and impact performance. Be sure to monitor paging rates and tune your server if they are regularly above 1,000 pages/second.



*Investigate memory issues first, as a memory shortage can cause secondary I/O or CPU bottlenecks.*

SQL Server gives you several options for allocating memory. You can configure SQL Server to use a fixed amount of the memory available on the server, dynamically allocate memory to itself according to demand, or dynamically allocate memory to itself within limits that you configure.

If adjusting how memory is allocated to SQL Server does not free up enough memory for the SQL Server instance to function properly, the physical memory for the server may need to be increased, or the concurrent application load on the server may need to be decreased. Look at which processes are consuming memory to identify potential application problems such as memory leaks, or processes that can be offloaded to a different server.

## Monitoring Disk I/O

Another important area to monitor is disk utilization to see whether the I/O subsystem is the bottleneck. Disk utilization over 50% will result in sluggish response time and poor application performance. When monitoring the I/O subsystem, be sure to monitor both the physical disk times and the disk queue length. If the physical disk is consistently busy servicing I/O then the disk may be the problem. Performance can also degrade if the disk queue length exceeds two. Disk queue length is the number of outstanding requests that need to be serviced by the disk.

As with memory, the way to solve a disk I/O bottleneck is to reduce the concurrent application load or upgrade your hardware (in this case, upgrade the computer with faster disks and/or more disks). Faster disks will cope with the I/O load better, whereas additional disks will allow for the I/O load to be spread across more spindles.

## Monitor CPU Utilization

When monitoring CPU utilization, remember that occasional spikes of 100% are nothing to worry about and are normal. Consistently high CPU utilization rates, over 90%, may indicate a poorly-tuned or designed application, or an underpowered system.

A good auxiliary metric to look at is the size of the processor queue. If you see high CPU usage and a processor queue that exceeds two over time, it means that processes consistently have to wait for the CPU – a sure sign of a bottleneck.

If that is the case, check for blocked SQL Server processes or processes using large amounts of CPU time. If you have one or a few CPU-intensive processes on the system, you will want to investigate possible bugs or efficiency problems with the application; or, it's possible that high CPU utilization is simply a signature of this particular application.

## Monitor CPU Ready Utilization when operating on a VMware Host

When operating SQL Servers in a virtualized infrastructure it is important to understand the makeup of the host and the accompanying VMs on the host. A high CPU Ready % means that SQL servers are “ready” to run but are waiting for physical CPUs to be available. High CPU Ready % is the result of an over-commitment of the number of virtual CPUs on the host, values exceeding 5% per vCPU warrant a closer look and over 10% and it is time to start rationalizing the number of VMs and vCPUs configured on the host.

## Monitor Key SQL Server Indicators

By keeping your monitoring strategy focused on memory, disk I/O and CPU utilization, plus a few key SQL Server indicators, you will decrease the time and energy associated with ensuring high performance and availability for SQL Server. Some good indicators are as follows:

- Make sure SQL Server data caching has at least a 90% hit rate
- Don't let the transaction log get more than 85% full. In general, you want to make sure you have at least 20% free disk space
- Check to make sure all SQL Server services are running. If a SQL Server service is not running, restart it
- Verify that your queries are performing at adequate levels, particularly watching for queries that are using excessive amounts of CPU time
- Don't let the rate of unsafe auto-parameterization attempts get too high
- Watch to ensure that data throughput is at a sufficient level
- Check the number of user connections and ensure that there are not too many
- Look at average wait time for locks to verify that they are not too high. This may be common in OLTP (online transaction processing) and other applications characterized by high volumes of similar small transactions

In addition to performance metrics, it is a good idea to monitor the SQL Server Error logs for the presence of errors that could affect the performance of the server. Some errors to watch for include:

Error 1204	Not enough locks
Error 1205	Deadlock victim
Error 1105	Can't allocate space for object
Error 17809	Unable to connect
Error 605	Database corrupted

If you experience any of these errors, consult Microsoft documentation regarding potential consequences and advice on how to respond to them.

Keep in mind that performance monitoring is an iterative process. A single snapshot of a few key metrics may help diagnose a current performance problem, but it takes an ongoing monitoring effort to ensure performance and availability over the long haul.

The next section, —Monitoring Proactively talks about the merits and tactics of an ongoing monitoring strategy which will allow you to sustain high performance and availability and diagnose and address problems before they reach the end user.

## **Monitor Proactively**

By capturing, analyzing and acting on performance data over a long period of time, you will develop a clear picture of exactly what constitutes normal behavior on your system, and you will be better equipped to detect and deal with the subtle symptoms that spell trouble ahead. A proactive monitoring strategy will alleviate the time and resources needed to sustain high performance and availability for SQL Server. This section will layout a blueprint for a reliable proactive monitoring strategy.

In any performance monitoring exercise, remember that you will need to make adjustments based on the characteristics of your various expected database workloads – which you can only do if you know what’s normal on your system. Also, remember that sound SQL Server performance and high availability may require several adjustments, or a combination of hardware and software remedies, in order to obtain the results that you’re looking for.

### **Establish a baseline**

One of the biggest pitfalls of performance monitoring is waiting for a problem to happen, then not having any sort of baseline for comparison. To establish a baseline, capture a week or two’s worth of performance data in log files. This data will serve as your benchmark moving forward. Capture as much data as possible. Disk space is cheap, so it’s OK to collect a large amount of data. It’s important to be exhaustive up front so you don’t have to revisit this step down the road when more granular or obscure metrics may be needed.

### **Refine the baseline**

With an effective baseline in place, the next step will be to monitor key metrics like memory, disk I/O and CPU usage over long periods of time, keeping logs and comparing performance over the long term. As you begin to get a sense of how user demands shift over time, you will be able to identify usage patterns and correlate actual processing load with anticipated processing load. In time, your perception of what is typical behavior for your SQL Server environment will become more accurate, allowing you to make smarter decisions about when to perform maintenance and upgrades. You will also be in a much better place to diagnose performance problems as they crop up, knowing how individual hardware and software components are performing as compared to your benchmarked data.

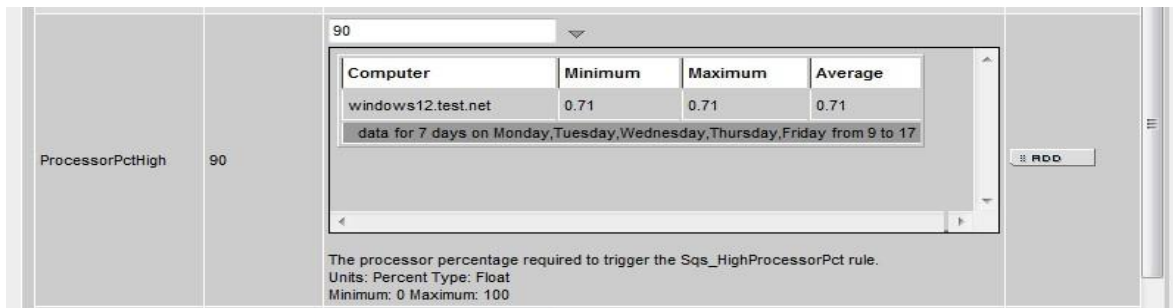
Be sure to monitor the response time of commonly run queries over time to gauge the relative performance of your SQL Server. If the response time for the same queries increases to unacceptable levels, the resources on the server may need to be increased. Performance patterns may emerge that also suggest that the application code needs to be refined.

## Troubleshoot

Once you have refined your baseline, you will be in a position to automate a good deal of the monitoring and problem diagnosis process by establishing thresholds for key performance metrics and setting up automated notifications – alerts and alarms - when thresholds are violated.

Having an accurate performance baseline for your SQL Server environment will allow you to create accurate thresholds. Preset, manufacturer-recommended monitoring thresholds are useful and necessary. However, you need to tune thresholds that are meaningful in context of your environment and user demands. Keep in mind that over time, as demands change, the thresholds should change as well.

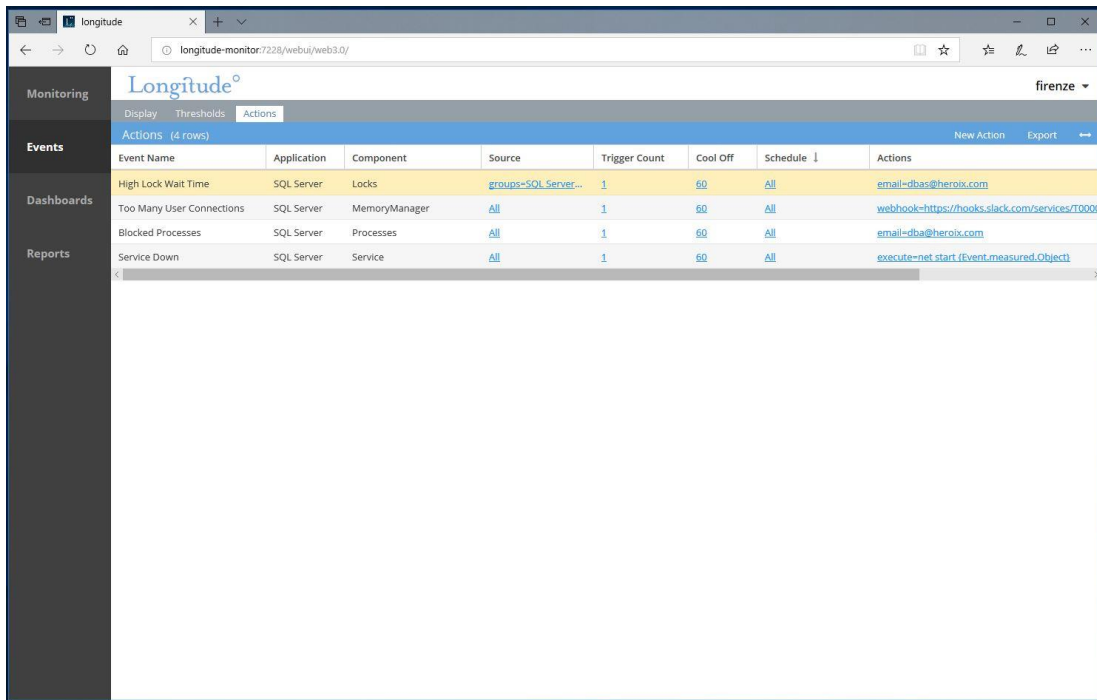
*Manufacturer-recommended thresholds should be adjusted to the usage patterns of each workload. Here, actual workload data makes it easy to fine tune thresholds for individual servers or groups of servers.*



With accurate thresholds in place, you can arrange to have alerts and alarms set up to notify the correct person. With the right monitoring solution, you can build in rules for escalating problems—issuing wider alerts, triggering phone calls or e-mails to people progressively higher in the IT department hierarchy—until a problem is solved. And last, you can build in automated corrective action that is triggered when predefined (or dynamically defined) thresholds are reached or other conditions occur. In all of these cases, you're not limited merely to responding after the fact.

With a proactive monitoring strategy, you will have the data you need to fix minor SQL Server performance issues before they mushroom into major problems and impact your service levels. Start with a baseline and a disciplined monitoring regimen which helps you refine your baseline and establish an accurate picture of usage patterns and the systems requirements needed to accommodate the current performance demands of your users. Once an accurate baseline is in place, you can automate portions of the monitoring function by creating predetermined thresholds that trigger alerts and alarms when human intervention is necessary to address performance problems.

*Email notification and corrective action help solve problems before users are affected.*



The screenshot shows the Longitude monitoring tool interface. The browser address bar indicates the URL is longitude-monitor:7228/webui/web3.0/. The interface has a sidebar with 'Monitoring', 'Events', 'Dashboards', and 'Reports'. The main area displays a table of events with columns for Event Name, Application, Component, Source, Trigger Count, Cool Off, Schedule, and Actions. The table contains four rows of data.

Event Name	Application	Component	Source	Trigger Count	Cool Off	Schedule	Actions
High Lock Wait Time	SQL Server	Locks	groups-SQL Server...	1	60	All	email=dba@heroix.com
Too Many User Connections	SQL Server	MemoryManager	All	1	60	All	webhook=https://hooks.slack.com/services/T00000
Blocked Processes	SQL Server	Processes	All	1	60	All	email=dba@heroix.com
Service Down	SQL Server	Service	All	1	60	All	execute-net start (Event.measured.Object)

No discussion of application monitoring is complete without spending some time on monitoring tools. While many applications and hardware components come with built-in monitoring programs, it just isn't feasible to use separate monitoring programs across your entire infrastructure. More complex enterprises require more sophisticated monitoring tools which take the complexity and diversity of your environment into account. This section will discuss some of the key things to think about when evaluating a monitoring tool.

## Make multiplatform a must

Most IT departments are charged with supporting an IT infrastructure that is multiplatform, and comprised of a diverse set of proprietary hardware. In choosing a monitoring tool, look beyond SQL Server and Windows and seek a solution that will support your company's other operating systems and applications directly out of the box. Operating system support may span Microsoft Windows®, Red Hat Linux®, SuSE® Linux®, Hewlett-Packard HP-UX®, IBM AIX®, Sun Solaris™, Hyper-V®, and VMware® platforms. Application support may extend to a variety of application servers, Web servers, databases, messaging, and infrastructure products from different vendors. Think Apache™, Oracle®, Microsoft Exchange, and the like. In addition, look for the ability to specify and track a range of user and business metrics, including transactions and service level agreements (SLAs).

You stand to get more from a multiplatform package designed to be truly platform-neutral from the start, as opposed to one developed for SQL Server / Windows environments and retrofitted for Unix or Linux platforms (or vice versa). The idea is to have consistent capabilities across the platforms you are monitoring.



## Reduce the burden of proving SLA performance

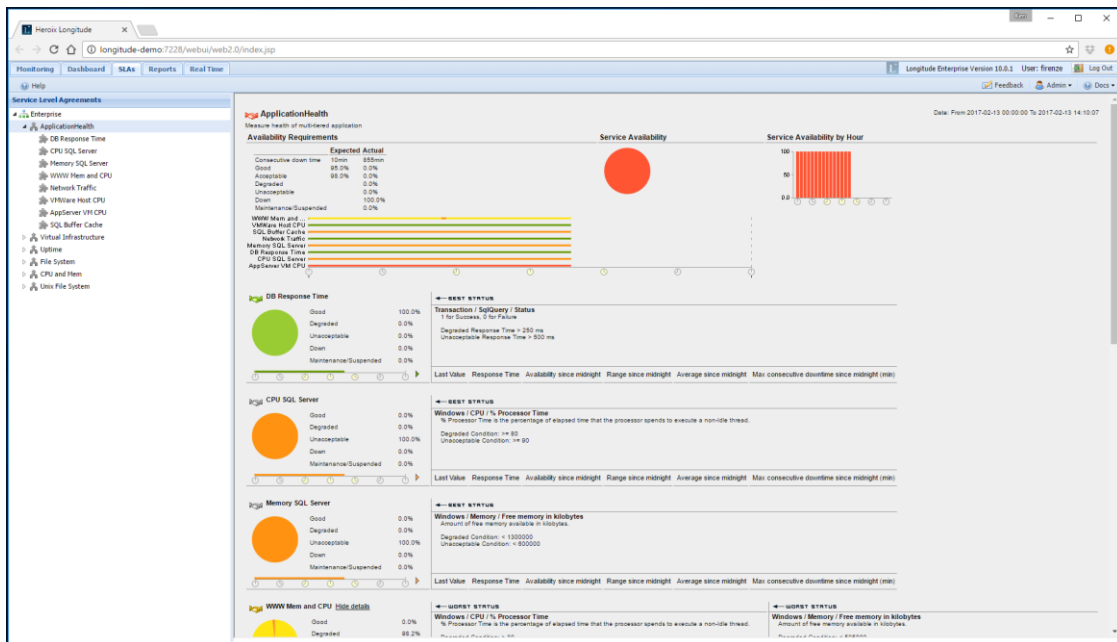
Even if you don't have formally written Service Level Agreements, you can benefit from SLA monitoring. SLAs enable you to look at delivery of specific business services and understand which problems are truly caused by SQL Server and which may be due to system, network, or application issues.

Even when users are satisfied with the SQL Server performance and availability levels you've achieved, you need to document that service level agreements are being fulfilled. Reduce the burden of proof with a monitoring solution that integrates the required reporting and graphing. Not only will you avoid developing reports and graphs from scratch, but also you will be able to produce documentation quickly and easily whenever your organization requires it.

What's more, a dashboard can present gauges, graphs, and statistics that let you evaluate SLA adherence in real time. You will then be positioned to spot developing trends and problems within the context of SLAs, allowing you and your team to act before users notice an adverse impact.

Whether supplying you with real-time or historical SLA information, a standout monitoring solution will correlate, analyze, and present key statistics in a graphical, easy-to-understand style. Summary views as well as in-depth details should be provided.

*Effective SLA monitoring will look across servers and applications to determine the overall health of the business service being delivered. Here, the service depends on the availability of a web page and the performance of the network, memory, and CPU, in addition to the availability of the SQL Server.*



Reporting and analysis should take into account that meeting an SLA normally depends on multiple factors. Aggregating data from monitored entities in this manner is beneficial in a number of ways. For instance, delivering a service to business users may require that a key SQL Server query, web page download time, CPU resources, and network traffic all perform at a — “good” level. If a problem affects one of these components - say that poor SQL Server caching slows database response time and impairs service - the monitoring solution should report the service level as degraded even though the condition of the other three components remains — “good”.

Another case where a monitoring tool can help by providing SLA information along multiple dimensions is server redundancy (whether the servers are clustered or not). Perhaps five different monitored servers are running a database application when one server goes down; that might still be — good for the particular service level. The condition would fall to — “degraded” if two servers went down, and so on. While the remaining three servers might all remain in good condition, the overall service level is the primary concern.

## **Employ a solution that uses workload characterization to aid in threshold adjustment**

One of the greatest challenges in SQL Server monitoring is setting and maintaining the right thresholds for memory, disk I/O, CPU utilization, and other functions. One size does not necessarily fit all.

What constitutes normal behavior for one database server may be very different for another. For instance, the SQL server for an OLTP application may carry much more volume than a similar server for another application. As a result, you would need a different, higher data throughput threshold before an alert for corrective action is triggered. Conversely, the thresholds on a lower volume application should be lower to avoid letting problems go unnoticed.

A monitoring solution that can provide you with workload data will be much easier to fine tune to your needs. In some cases, it may even be able to do the job for you by adjusting monitoring thresholds automatically. This not only saves valuable staff time and energy, it keeps your SQL Server monitoring and management at peak performance.

## **Consider agentless monitoring**

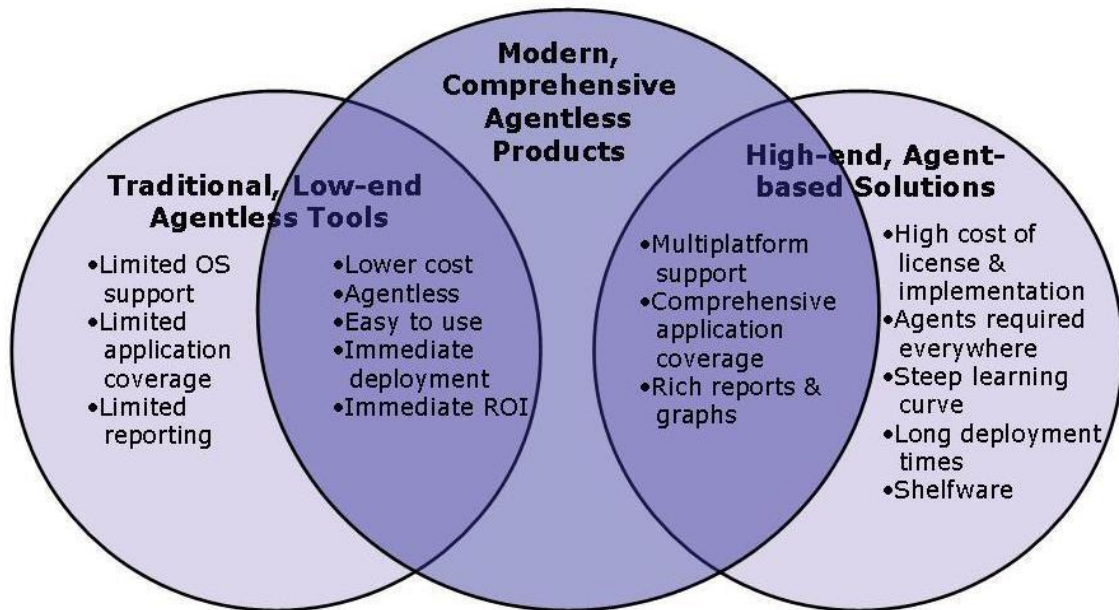
The distinguishing characteristic of agentless monitoring is that you don’t have to install software agents on the systems you are monitoring. In addition to the benefits that accrue from not having to introduce software on production SQL Servers, the primary upside of the agentless approach is the simplicity and savings that come from faster deployment, lower software license fees, and streamlined operation. Until recently, these benefits came with a steep downside: very limited capabilities. You usually could not expect an agentless product to do much more than collect a few statistics. The more recent trend is to build on industry-standard technologies, such as WMI, SSH, Telnet, rexec, HTTP, and JDBC, taking agentless monitoring to a higher level.

Choose well, and today it’s possible to find an agentless monitoring and reporting product that combines affordable agentless monitoring with strengths once provided exclusively by more expensive agent-based software. The best of the new breed of agentless solution will offer the most desirable features of high-end monitoring: multiplatform support, comprehensive application coverage, extensive reports and graphs right out of the box.

Determine whether the agentless product you are considering requires the purchase and installation of layered software. Some packages are not self-contained, which means they rely upon add-ons such as a database or report writer to be usable. Prerequisites like these diminish the cost savings and ease of use that you’re after.

A comprehensive, out of the box agentless monitoring solution, however, will reduce the complexity of implementing and administering a solution, and add some transparency to the overall cost of your monitoring investment.

## **Comprehensive Agentless Capabilities**



*Modern agentless products combine lower cost and ease of use with the comprehensive coverage previously provided only by higher end, agent-based solutions.*

## Summary

Delivering the performance and availability that end users demand from SQL Server begins with a sound monitoring strategy. The more focused you are on the metrics that matter most to performance and availability, the easier it will be to diagnose and troubleshoot problems. Focusing on memory usage, disk I/O and CPU usage will help you address a majority of performance related issues for SQL Server.

A sound monitoring strategy will allow you not only to quickly diagnose problems, but to sustain high performance and availability over the long haul and anticipate and address problems before they reach end users. Monitor your SQL Server environment proactively by establishing a performance baseline for your environment and using it as a benchmark for all of the data you collect in the future. By understanding typical usage patterns and performance demands of your environment, you will be able to make better judgments about performance related issues that crop up and you will also be able to better plan for upgrades and maintenance that may impact end-user performance during peak usage.

Finally, when choosing a monitoring tool, look beyond individual applications and seek out a monitoring tool that will provide a broad view of your entire infrastructure. IT environments today are a heterogeneous mix of hardware and software. By focusing monitoring efforts too narrowly, you will not be able to accurately consider all the many factors that contribute to application performance. Always keep the big picture in mind!

---

## About Heroix

Heroix has a 30+-year history of proven monitoring solutions, with products running on tens of thousands of critical servers. It offers fast, easy, affordable application and networking monitoring solution for physical and virtual environments. [Download Longitude Now](#) and you'll be monitoring and planning in just 10 minutes.

Heroix believes that the information in this document is accurate as of its publication date; such information is subject to change without notice. Heroix is not responsible for any inadvertent errors.

Heroix, Heroix Longitude and their corresponding logos are registered trademarks of Heroix. All other company and product names mentioned are used only for identification purposes and may be trademarks or registered trademarks of their respective companies.

Copyright © 2020 Heroix. All rights reserved.