REDUCE MTTR: MACHINE LEARNING TO THE RESCUE

It's not uncommon for performance monitoring strategies to leverage thresholds to alert operators when certain metrics for key performance indicators (KPIs) go outside what is generally considered acceptable for IT apps and services to run smoothly.

To better understand this, consider when an app on your computer starts consuming a lot of memory and everything else seems to slow down.

While most of us don't have thresholds and alerts configured to mitigate our slow computer issues, enterprise-level computing requires this exact kind of notification *and much more* to keep IT operations at optimal performance. Think of it this way, instead of just *you* having a bad experience while an app is using up memory from your computer, imagine *thousands of users* accessing an app that is consuming memory from a bunch of servers. When IT is alerted with an issue from a threshold violation, they want the fastest mean time to repair (MTTR).

Helpline scenario

We call our bank's or healthcare provider's helpline and the customer service professional says, "My computer is slow today." He or she is accessing one or more of the mission-critical apps within their enterprise system. They are being affected one way or another by some element in their IT environment that causes your request for help to slow down. Slow service coupled by a bad experience might make you consider other options. What if hundreds or thousands of customers are having this same bad experience?

This customer service professional enters a service ticket for the slow app and the system also generates alerts based on one or more conditions that led to this bad user experience on the helpline.

Complexity

What makes these issues particularly complex is that enterprise-level app performance requires tip-top performance across all tiers of web servers, app servers, and database servers in traditional architectures as well as in more modern, highly-distributed environments with microservices. For most enterprises, the complexity behind mission-critical apps span data centers, private and public clouds, and even brokered services from other providers. If the apps are not running smoothly, chances are the business is losing revenue, productivity, and even brand loyalty.

If IT operations staff get an alert every time a performance measurement reaches a threshold for even a brief period, managing the volume of these alerts can be a nightmare. The issue might not even have a significant or sustained impact on end-user experience. This situation can be very inefficient to address.

That's why modern IT Ops solutions provide ways to go beyond basic monitoring to reduce the noise associated with event storms.

How machine learning and analytics can help

Instead of relying on static thresholds, here are several machine learning and analytics approaches to fix issues based on priority and impact so that IT operations can reduce MTTR:

Behavioral Learning—Use a combination of baseline data from machine learning and a threshold value to determine when an issue is more likely to be a problem. You only get notified when a measurement is outside of what is learned as normal for the time of the baseline data *and* is above the critical performance value of a threshold that you specify. A typical scenario for this would be during expected high demand usage at the beginning of the day or work week. CPU utilization for servers supporting apps could spike for an expected length of time every Monday morning between 9 AM and 10 AM.

For example, you have set an absolute threshold of 80 percent utilization for CPU—*generally considered as a good value to start being concerned about performance of your servers*. If utilization spikes to 90 percent against a learned baseline for this period between 85 - 95 percent, you would not get an alert. If, however, there's another time during the week when usage is at 90 percent with the baseline between 65 – 75 percent, you would get an alert that you need to address. In the first case, the actual usage does not go above the threshold *and* outside the learned baseline. In the second case, the actual usage is above what is considered normal *and* is above the absolute threshold.

This helps you prioritize the most important issues first.

Predictive Events—Use a combination of hourly baseline data and a critical threshold value to receive an alert or event notification when a measurement is about to reach the critical threshold (typically about three hours). Machine learning provides the hourly baseline data and you set the critical threshold. When a measurement starts to go outside the baseline value but before it reaches the critical threshold, you essentially get a warning. If you don't do something in time, performance

can be significantly impacted.

Someone on your IT team might change a configuration that could unexpectedly flood a certain resource during peak times. A predictive event would give you plenty of time to address the issue before users even notice.

Probable Cause Analysis—Use a process that analyzes data and displays all relevant events and anomalies when troubleshooting an issue. In complex IT environments, there are typically many factors that can impact the performance of an application or service. By having the system narrow down what is the likely cause of an issue, IT operations can pinpoint the root cause of the problem faster. Relevant events to consider are ranked and displayed based on their relationship to the initial event, the timeframe of consideration, and abnormalities captured by behavioral learning.

You could be alerted that a server in your environment has processor time that has spiked above what is considered normal and is related to a memory issue. Because of the spike in memory, the server does not respond to requests for data as quickly. By addressing the memory issue, the processor time goes back down to normal.

Pattern Matching—With a little upfront planning, you can identify probable cause scenarios that typically happen in your environment as *knowledge patterns* so that troubleshooting issues can go even faster over time. In the case of processor time and memory in the previous example, you could create a knowledge pattern that identifies critical sets of infrastructure that you need to address when both processor time and memory are an issue. You would get notified of the probable cause event rather than having to run a new probable cause analysis.

If the processor time is unusually high but there's no memory issue, then you could do further investigation for other issues.

Log Analytics—With machine learning applied to volumes of log files, the system can baseline what is considered normal for like log entries. That means what is considered "normal" is learned by the system. At any time when the number of log entries that match the pattern change by an unusual amount, you can then quickly identify an issue in your IT environment.

This is what is considered as an "out-of-bound" anomaly. The issue is occurring far more or far less frequently than what is normal. This would be particularly useful if you suddenly have a lot of users with access to a machine who did not before or in the opposite case, a lot of users don't have access when they did before. There were configuration changes related to access lists (ACLs) that you need to address.

A long history at BMC and more to come

These machine learning and analytics techniques have been a part of the TrueSight portfolio for years and it's just getting better. Innovating with cross-platform and 3rd party data support, TrueSight now offers service ticket analytics and correlation with time-series data from almost anywhere. You can quickly ingest volumes of real-time and historical data in minutes to uncover patterns and find root causes for what used to take many days and hours for IT operations staff to address.

Going beyond basic monitoring with TrueSight to reduce the mean time to repair—respond, remediate, rescue (MTTR) any IT operations issue.