PRACTICE MAKES PERFECT

How Event Management Best Practices Are Delivering Real Value to the Enterprise

By Mike Moser, BMC Product Management Director,
and Mark Rascoe, BMC MAINVIEW Product Manager

When you're driving down the highway, do you need to know precisely how much fuel remains in your car's gas tank? Not really. However, you do need to know when the fuel level is low and it's time to start looking for a gas station.

Likewise, when you're managing a mainframe environment, you don't need precise details on CPU utilization or how many bytes of storage are available at any given moment. What you need to know is when conditions exist that could bring business-critical systems to a screeching halt.

Event management tools alert the right people when certain conditions occur or when certain thresholds are exceeded, enabling IT staff to intervene and prevent performance degradation and outages. However, conditions are constantly fluctuating and events are constantly occurring. As a result, the IT staff is deluged with thousands of alerts and notifications, the majority of which don't represent a risk to effective operations and don't require human intervention.

The ability to cut through the noise quickly and pinpoint the events that really matter is vital. This ability is contingent upon more than simply having event management tools in place. It also involves implementing best practices around determining what you want to measure, identifying the metrics to use, establishing the thresholds that make sense, and taking the right steps — particularly automated steps — when action is required. Following these best practices allows IT to do a substantially better job of keeping critical systems running while reducing costs and boosting IT productivity. Furthermore, these best practices lay a foundation for continuous improvement and innovation that allows the shop to grow and scale as workload and environmental complexity increase over time.

Identifying Metrics that Matter

There are many metrics for event management, but when IT organizations try to monitor them all, the result is people frantically trying to respond to all the red alerts that show up on their consoles. Best practices call for identifying the metrics that deliver true value for your enterprise. For most organizations, this may be no more than 10 to 15 core metrics, such as those for thresholds, resource utilization, events, alerts, time-to-resolution, and diagnostics. By focusing on

those metrics that really matter, you can ensure that the events that appear on the operator consoles are the ones that really need immediate attention.

The IT organization of a large insurance company faced this challenge. Major outages occurred regularly, and the staff spent hours trying to pinpoint the problem. They collected data on a variety of metrics but not the metric that was required to zoom in on the cause of the outage that impacted their business.

By automating the process of identifying issues, taking corrective actions, and opening tickets for tracking, they reduced the hours of time-to-resolution (TTR) down to minutes.

To get on track, the staff began using a core set of metrics. Next, they analyzed historical data to establish valid thresholds for those metrics, identified who needed to be notified, and defined additional metrics required (diagnostic metrics) to resolve the issue. Help desk tickets were opened when the thresholds were exceeded and included the diagnostic metrics (which helped the staff resolve the issue faster), provided data on how often shortages were occurring, and allowed for examination of the impacts on the business processes. As a result of this insight, the staff was quickly able to avoid related outages in production systems. By automating the process of identifying issues, taking corrective actions, and opening tickets for tracking, they reduced the hours of time-to-resolution (TTR) down to minutes. The reductions in wasted CPU cycles and the labor to detect and document the issues were obvious compared to prior techniques of having people watch fields on a screen change colors.

A key insight driving best practices is that most metrics collected in typical monitors are simply "diagnostic" in nature and are not suitable for driving threshold violation events. We introduced the concept of "core" metrics earlier, and it is these metrics that should get the attention for threshold management. Any monitoring solution has major problems if the thresholds used are not correct. The number of core metrics varies from one organization to another and depends on such factors as the configuration of infrastructure components required to support the business processes.

For organizations that have an IBM DB2 data sharing group, for example, best practices call for monitoring timeouts and deadlocks so the staff can manage DB2 sharing. A good example of how core metrics are important is based on the experience of another major insurance company, which was suffering from DB2 lockouts that were impacting the business. The staff was not monitoring these core metrics and was not being notified when the condition was occurring; therefore, they couldn't identify who or what was causing the problem and negatively impacting service.

By monitoring for timeouts and deadlocks, the staff was able to collect data and provide detailed information to incident tickets that helped resolve the problem. Through monitoring the core metrics and using the diagnostic metrics when needed, the staff captured the SQL code in cache before it was flushed out. This information was passed on to application developers who then incorporated a fix that prevented future occurrences. By monitoring the correct metrics and using valid thresholds, the IT organization dramatically improved its business process response times, reduced CPU resources, and ensured that service level objectives were being met.

Setting Thresholds that Make Sense

Identifying the metrics that matter is just the starting point. Next you have to figure out the thresholds that

make sense for your environment. Best practices in this area include the analysis of historical data and the use of persistent checking.

For the large insurance company that reduced outages by monitoring the Common Storage Area (CSA), historical data showed that storage areas were running at about 97 percent utilization in development systems and 93 percent in production systems. Based on best practices, 80 to 85 percent utilization would be a more acceptable threshold. However, due to a large number of IBM IMS databases running on their systems, the amount of CSA used was above the recommended thresholds. The company was able to analyze historical data to identify what had caused the CSA to exceed acceptable utilization thresholds and then take the necessary actions to maintain acceptable levels. Analyzing historical trends of the metric provided the understanding of how the thresholds should be set to monitor this critical resource effectively.

Persistent checking is important when the period of time that the threshold is exceeded is more important than actually exceeding the threshold. To continue the automobile analogy, if you're driving a diesel-powered car, you need to track not only the fuel level but also the presence of water in the fuel line. In today's diesel vehicles, when sensors detect water, they continue taking samples over time and alert the driver only when the level of water reaches a threshold and remains above that threshold, indicating the need for some kind of action, such as flushing out the fuel line or putting in an additive. This type of persistent checking eliminates false alarms.

Persistent checking works the same way in the mainframe environment. Just because a threshold is exceeded does not mean there is a problem. In many situations the metric that is being monitored may see spikes. Persistent checking is used to eliminate false alerts due to spikes. For example, the average response time of a critical business service may experience several spikes throughout the day. These spikes are normally driven by transaction volume and demands placed on the system from other workloads. If the average response time exceeds the threshold over a given interval, this does not necessarily mean a problem has occurred.

Checking the threshold over a defined number of intervals before creating an alert is important. Also, using a combination of different metrics over a given interval can reduce false alerts. To monitor the business service response times, using valid thresholds for the number of transaction and the response time metrics over a given number of intervals will drastically reduce false alerts.

With metrics identified and thresholds in place, you're now able to ensure that events coming into the consoles are real events that need attention to prevent performance degradation or outages.

Taking the Right Actions

With metrics identified and thresholds in place, you're now able to ensure that events coming into the consoles are real events that need attention to prevent performance degradation or outages. In line with proven best practices, you now need to figure out what actions to take when thresholds are exceeded and to prioritize events based on the business impact so that critical events receive top priority. Questions to answer include the following:

- Who needs to be notified?
- What additional data is required?
- What steps must be taken to resolve the problem?

Incorporating automation at this point can be highly beneficial. That automation might be as simple as generating a ticket for a CICS transaction that is starting to loop and then sending the ticket to the appropriate technical team. If the event is occurring on a critical system, and if business users might be affected, the event might be sent to an impact management tool or escalated in some other way to speed response.

The insurance company discussed earlier in this article has a complex CICS environment. Business processes were being affected due to misbehaving CICS transactions. The staff was having difficulty capturing information that would help application developers fix the problem to eliminate the bad transactions.

The staff was able to determine that any transaction that consumed 50 seconds or more of CP time represented a problem. So the staff created an automated process that could pinpoint any transaction that exceeded 50 seconds and purge it. Before the purge, however, the process collected detailed statistics on the transaction, performed some analysis behind the scenes, opened a ticket, and sent everything required for a fix to the administrative person responsible for the CICS region. This automated approach saved hours of operator time that had previously been required to collect the statistics, format them, open the ticket, and send the ticket to the right person.

Moving Ahead

Best practices for taking care of a car involve identifying metrics and setting thresholds — making sure that your gas tank never goes empty, your is car serviced at certain intervals, the warning systems works properly, and the car is performing optimally. It involves taking the right actions — such as rotating the tires — so that your drive is a safe one. By following this same approach for managing events — identifying metrics, setting thresholds, and taking the right actions — you can prevent conditions from slowing down your systems and can even move ahead at full throttle.

For information on BMC solutions for event management, visit www.bmc.com/solutions/msm-main/systems-management.html.

To learn more about how BMC can help activate your business, visit www.bmc.com or call (800) 841-2031.

ABOUT THE AUTHORS

Mike Moser, a product management director and program executive within BMC Software's Mainframe Service Management business unit, focuses on issues related to reducing costs while improving IT efficiency and service delivery. Moser's more than 20-year career has spanned a wide variety of engineering, IT management, technology consulting, and product management positions across both technology vendor and IT end-user organizations.

Mark Rascoe is a senior product manager in BMC Software's Mainframe Service Management business unit. He has been a senior software consultant for over nine years at BMC, implementing monitoring and automation solutions. Before coming to BMC, Rascoe was involved in IT capacity planning strategies and system and application performance studies. He has over 25 years of experience in IT performance management.

BUSINESS RUNS ON I.T. I.T. RUNS ON BMC SOFTWARE

Business thrives when IT runs smarter, faster, and stronger. That's why the most demanding IT organizations in the world rely on BMC Software across both distributed and mainframe environments. Recognized as the leader in Business Service Management, BMC offers a comprehensive approach and unified platform that helps IT organizations cut cost, reduce risk, and drive business profit. For the four fiscal quarters ended March 31, 2009, BMC revenue was approximately $1.87 billion. Visit **www.bmc.com** for more information.

103544