Proactive monitoring for dynamic virtualized environments

By David Weber and Veronique Delarue

Virtualization can significantly increase monitoring complexity. By using BMC® ProactiveNet® Performance Management, IT departments can create an intelligent, automated approach to identifying and prioritizing issues in these dynamic environments.

Monitoring can be as simple as waiting for something to happen. Whether it's monitoring the availability, performance, or capacity of an endpoint, and whether that endpoint is a server, a storage device, or a piece of software running on both, monitoring is one of the simplest activities in IT.

Or at least that might have been true before virtualization came along. Now, not only can availability, performance, and capacity change from moment to moment, but the challenge of setting useful thresholds on those attributes has become dramatically more complex than it was in traditional physical environments.

For effective monitoring of a dynamic virtualized environment, flexibility, adaptability, and a deep understanding of that environment are critical. BMC ProactiveNet Performance Management (BPPM) is designed to address each of those facets. In BPPM, *flexibility* means the ability to collect key metrics and alerts from hardware, operating systems, and applications whether those components are instrumented for data collection or not. *Adaptability* means that BPPM gathers this critical data continually even as the environment changes. And BPPM's comprehensive and accurate data model enables administrators to make the most of the data to help them understand their environment. Through deep analytics, dynamic baseline capabilities, and—with the Hardware Monitoring component—the ability to retrieve hardware data directly from the hardware either in band or out of band, BPPM helps IT administrators to ensure the availability and performance of complex environments.

In-band and out-of-band monitoring

Hardware Monitoring is a modular element of BPPM available both as an agentless element and as a Knowledge Module® (KM) component. The KM loads rules and intelligence into the BMC PATROL® Agent to enable the collection of hardware data, including Dell™ PowerEdge™ server alerts and other performance and health status data. (For more information on KMs and agents, see the "What is a Knowledge Module?" and "Agent or agentless?" sidebars.)

In total, the Hardware Monitoring component offers three options for hardware monitoring: in band, out of band through the OS, and out of band through a server's Integrated Dell Remote Access Controller (iDRAC). *In-band* monitoring refers to the data collection from within the OS. In this approach, the PATROL Agent resides in memory and on disk within the OS running on a server, and collects OS performance metrics. The Hardware Sentry KM tells the PATROL Agent how to retrieve alerts and data from the

Reprinted from *Dell Power Solutions*, 2010 Issue 4. Copyright © 2010 Dell Inc. All rights reserved.

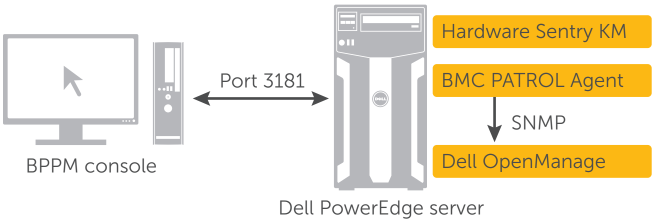


Figure 1. BMC PATROL Agent communicating locally with Dell OpenManage

Dell OpenManage™ Server Administrator agent and how to collect native Intelligent Platform Management Interface (IPMI) or RAID controller data directly from Microsoft® Windows Server®, Linux®, Solaris, or VMware® ESX and ESXi platforms (see Figure 1). The modular BPPM architecture enables administrators to choose their targets and monitor only the components of the infrastructure that need attention.

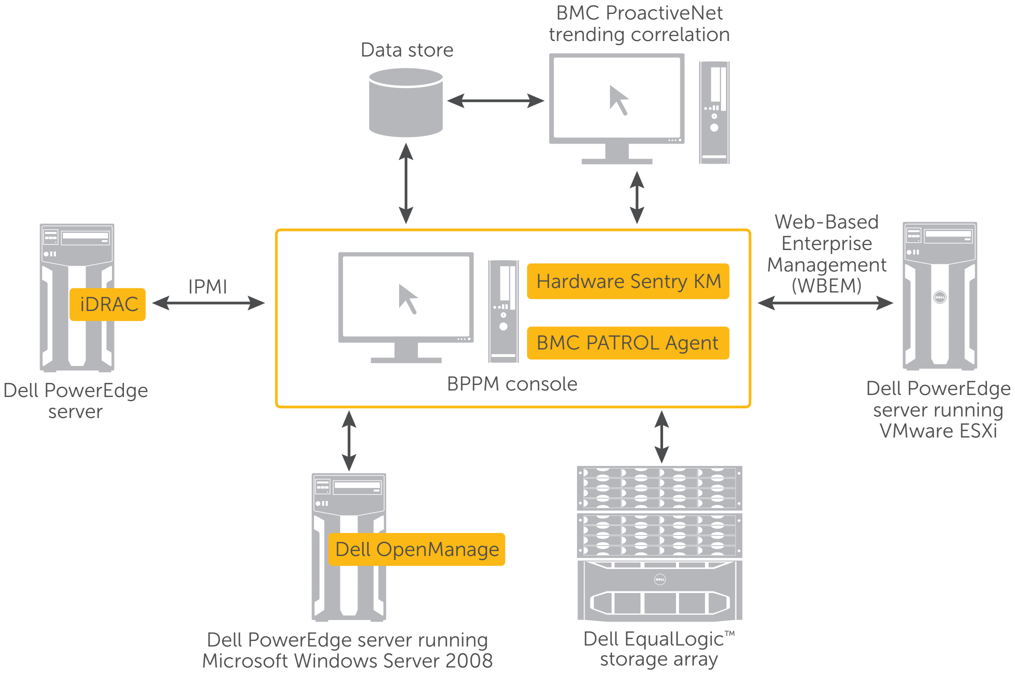
Out-of-band monitoring refers to the collection of the same data from an external vantage point. Administrators can use the PATROL Agent for either in-band or out-of-band data collection. For agentless operation, the PATROL Agent can act as a proxy (see Figure 2).

A single agent installed within a data center can handle out-of-band monitoring for many servers, enabling highly flexible monitoring architectures. In addition, BPPM administrators can choose to collect server alerts by communicating with the OS out of band or by direct communication with the hardware. The iDRAC in Dell servers presents IPMI, HTTP, and Simple Network Management Protocol (SNMP) interfaces, all of which are accessible with the Hardware Monitoring component. Administrators can therefore choose from a number of event collection options depending on the environment.

Comprehensive automation

Flexibility is critical, but with so much data coming in from servers, operating systems, and applications, what happens when virtualization enters the mix? The rapid and often unpredictable changes in configurations that can occur in virtualized environments can significantly challenge administrators' ability to keep up: when virtual machines migrate from one server to another, for example, static monitoring may leave them monitoring the wrong server. BPPM combines

Figure 2. Example architecture using BMC ProactiveNet Performance Management for agentless monitoring



What is a Knowledge Module?

As the name implies, a Knowledge Module (KM) adds capabilities, or “knowledge,” to a BMC agent and console in the form of additional routines that execute on the agent, as well as filtering and reporting insight at the console. KMs are available for hardware, operating systems, hypervisors, databases, and application platforms.

The BMC PATROL Agent was designed to accept KMs as flexible, integrated extensions. A KM may contain simple data filtering rules, or—as in the case of the Hardware Sentry KM—may contain complex code that enriches the PATROL Agent, enabling it to query hardware interfaces like Intelligent Platform Management Interface (IPMI). Other KMs available from BMC allow the PATROL Agent to execute SQL queries, or to communicate directly with application-specific interfaces like Open Database Connectivity (ODBC), Java Management Extensions (JMX), and Microsoft .NET.

a comprehensive impact model with dynamic thresholds and proactive analytics to help address the challenges of monitoring in these environments.

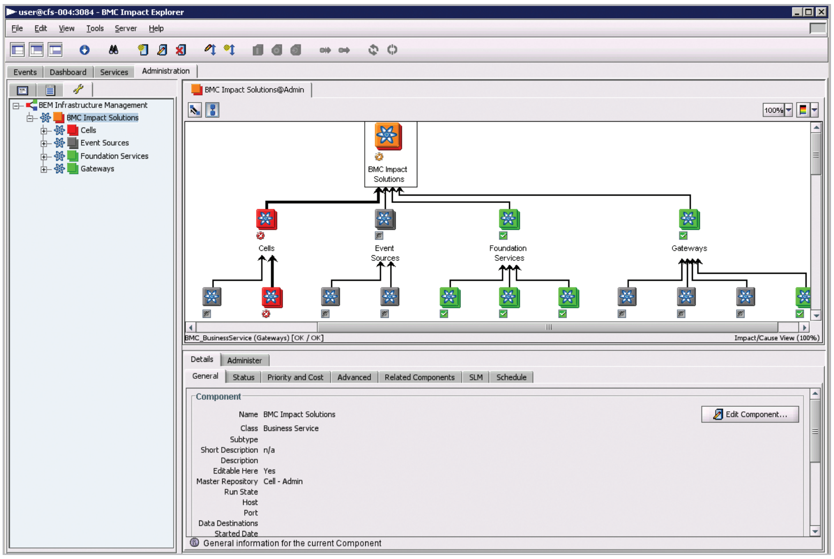
Impact model

A comprehensive data model is a good step toward improved modeling—if it can be maintained and trusted to always represent the current state of the environment. BPPM relies on an impact model not only to represent the configurations of the servers, storage, operating systems, hypervisors, and applications, but also to map those discrete components against their value to the enterprise (see Figure 3). In other words, this impact model describes the functional IT components as well as their relationships to business services. Services like order processing and shipping include not only IT resources like servers and applications, but also the person-to-person processes that depend on those resources to be successful.

When a fault occurs within the infrastructure, the impact model enables a BPPM operator to see the source of the fault as well as the relationships that the faulty component has to the rest of the infrastructure. Because the model also describes how the infrastructure components support business services, operators can quickly prioritize remediation. For example, imagine a scenario in which two alerts come in: one from a Web server in a hung state, and another from a database server running out of memory. Without an impact model, the operator would likely prioritize the resolution of these problems on a first-come, first-served basis. Using the impact model, however, the operator can quickly determine that the Web server supports customer-facing order entry systems, while the database is a secondary replication target for payroll—which is not due for another 10 days. In this situation, it is easy to see how critical the impact model is to the effective evaluation and resolution of IT problems.

Dynamic thresholds and proactive analytics

Although the impact model enables operators to prioritize remediation activities and target the



Name BMC Impact Solutions
Class Business Service
Subtype
Short Description n/a
Description
Editable Here Yes
Master Repository Cell - Admin
Run State
Host
Port
Data Destinations
Started Date

General information for the current Component

Figure 3. Infrastructure impact model in the BMC Impact Explorer® interface

infrastructure components that need immediate attention, it does not reduce the volume of alerts that come in. In a complex environment with events coming in from a multitude of sources, even basic prioritization can become challenging for console operators. To help effectively monitor applications, BPPM collects alerts from the server, storage, and network hardware; the operating systems and hypervisors (if applicable); and the services on which the applications depend, such as databases, application servers, and middleware.

In typical monitoring scenarios, operators set thresholds to determine when discrete components like servers or applications need attention. When a server's processor utilization becomes too high or an application consumes too much memory, a threshold is crossed and an alert is generated. In virtualized environments, however, the goal is generally to maximize processor utilization, and what constitutes "too much" memory for a given application may change based on the application's requirements from day to day. So although thresholds can provide guidelines, they still require constant attention and readjustment, and simply setting and adjusting monitoring thresholds can quickly become a full-time job.

Thresholds represent generalized rules for how a given application or infrastructure component is expected to perform. The problem with static thresholds is that they fail to take into account the dynamic nature of today's systems. For example, when setting performance thresholds for a database server, it is important to closely monitor the memory consumed by and available to the database application. The accepted process for determining the correct low and high ends for these metrics is to turn on logging for some period of time and then examine the logs to determine the minimum and maximum memory thresholds for the database.

The appropriate amount of time to capture these logs depends on the application, but it should at least represent a full duty cycle. A payroll application, for example, should be observed for at least one full payroll cycle. While the system is idle between payroll periods, the database is likely to use a relatively small amount of memory, and in the days leading up to payday, memory utilization would typically increase. The thresholds identified from these observations are then set within the monitoring system, and if memory utilization goes below the minimum or above the maximum, an alert is generated. An administrator must then try to determine the cause of the alert.

Static thresholds are not only time-intensive to establish; they also tend to lead to a situation in which alerts are generated only after a failure has occurred. Because static thresholds do not take into account variances in time, an operator could miss important indicators that lead up to a failure. These indicators represent an opportunity to prevent the failure rather than simply respond to it after it has occurred.

BPPM's proactive analytics feature is designed to bring automation to the task of monitoring. BPPM begins by analyzing the stream of events coming from the infrastructure. In a short time, it can determine what normal performance looks like for an application or server, and over time it generates a baseline that represents normal performance at a given time (see Figure 4). Both the baseline and the value of that baseline at a given point in time represent the work that goes into creating static thresholds. BPPM can determine rough thresholds within hours; within days, it can build a highly accurate model of expected behavior at all times. If a server reaches

Agent or agentless?

The choice between loading an agent on a managed server and collecting data from an external vantage point often comes down to a balance between the criticality of the monitored server and the performance impact of polling. Agent-based in-band monitoring runs in the OS, in parallel with critical applications. In the event of a hardware fault, the OS—and, by extension, the agent—may go down, preventing a critical alert from being sent. However, software issues like application performance problems may be detected immediately, with detailed alerts then forwarded to an administrator.

Because out-of-band agentless monitoring depends on polling, administrators using this option must decide how often to poll the monitored server. Even if the hardware or OS were to crash, agentless monitoring would alert the administrator—but only the next time the server is polled.

100 percent processor utilization or a database's memory footprint increases during a heavy traffic period, BPPM can determine whether these events represent a significant change from the expected behavior. Distinct from static thresholds, BPPM's dynamic thresholds determine an application or server's low and high ends at any given point in time, taking the manual labor and guesswork out of setting thresholds.

For example, by applying dynamic thresholds to the scenario of the payroll application, BPPM could determine whether the database server's memory usage was higher than normal during the time between paydays. Even though the memory usage might not be as high as the static maximum, BPPM would notice that the database was consuming more memory than it otherwise would at a time between paydays. Long before the database reached a point where it would have to be shut down and restarted because it exceeded a static threshold, BPPM would alert an operator to the anomalous condition and allow IT staff to address the issue without affecting a critical resource during a production time period.

Another advantage of proactive analytics is root cause identification. As noted earlier in this article, comprehensive monitoring involves the collection of data from all the components that make up the impact model, from servers and storage to operating systems and hypervisors, applications, and middleware. Because BPPM collects data from all of these sources and analyzes the entire event stream, it can automatically correlate events that come in from different sources. When an application stops responding, multiple alerts are often generated—BPPM may receive alerts from the application server, the OS, the server, the storage device, and the network, to name a few. When these alerts come in, BPPM refers to the impact model to determine the relationship between these components and identify the single event that most likely precipitated the entire stream of alerts.

This scenario provides another key example of automated monitoring. In the earlier example of the hung Web server and the database server running out of memory, the console operator would examine the impact model to determine the highest-priority fix. Proactive analytics enables that same operator to sort through multiple alerts and prioritize a fix for exactly the right problem.

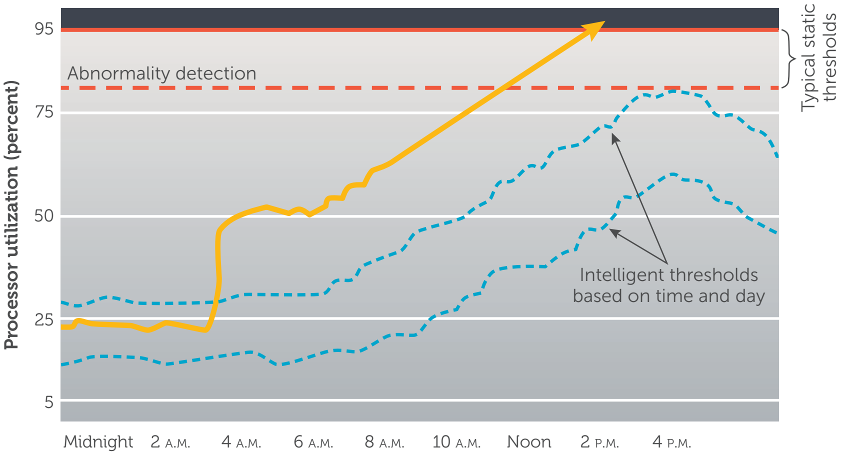


Figure 4. Example dynamic thresholds in BMC ProactiveNet Performance Management

Deep insight into complex virtualized environments

Effective monitoring in virtualized environments requires an approach designed for the dynamic nature of virtual systems. On top of a solid foundation of hardware monitoring enabled by the Hardware Monitoring component, BPPM offers deep insight into IT infrastructure—as well as advanced capabilities like the impact model and proactive analytics—to help IT administrators get the most out of these complex environments. PS

David Weber is lead integration manager at BMC Software and a 25-year veteran of the IT industry, having spent most of that time in systems and service management.

Veronique Delarue is a technical and marketing writer at Sentry Software with over 15 years of experience in the computer systems, software, and telecom industries.

Learn more

BMC Software:
bmc.com

Dell systems management:
dell.com/openmanage