Passive Monitoring—Maintaining Performance and Health

Introduction

One of the negative effects of the continued evolution of the load balancer into today's application delivery controller (ADC) is that it is often too easy to forget the basic problem for which these devices where originally created—creating highly available, scalable, and reliable application services. We get too lost in the realm of intelligent application routing, virtualized application services, and shared infrastructure deployments to remember that none of these things is possible without a firm basis in basic load balancing technology. One particular capability, a core foundation of load balancing technology, is the requirement to monitor the health of the application servers and to identify when there is a problem. Once one of the key differentiators between products, it is now rarely discussed; however, that doesn't lessen its importance or its possible negative impact on the application itself. Health monitoring, or the ability to verify that back-end systems are operational, is a basic tenant of load balancing and therefore critical to ADCs. Like almost all aspects of Application Delivery Networking, health monitoring has been plagued by "intelligence" versus "performance" issues since day one.

Health Monitoring: A Historical View

Without a basic understanding of the current state of the back-end application, none of the advanced features of today's ADCs are all that useful. If we are unable to know when an application is malfunctioning, or simply not there, we contribute nothing to the process. As we'll see, the health monitor of today has been a long time coming.

The original "health monitor" of back-end applications, still used by many products, was the simple ICMP PING of the server hosting the application. While this could certainly communicate that the application server was receiving network traffic—the absence of which was a definite sign that the application was unavailable—it didn't tell you anything about the actual application state on that server. In other words, you could definitively prove that the application was unavailable (no PING response), but you were never quite sure if the application itself was up or not. For example, Windows NT Server was notorious for responding to PING even though the system itself had "blue screened" and no real application was running or capable of processing those network packets.

The next iteration of the health monitor was the migration from network health monitor to TCP health monitor. Instead of relying on lower layer responses, they attempted to interact with the TCP port associated with the application and verify that a connection could be made, signifying that an application was running and listening for users. A typical example would be to attempt to attach to TCP port 80 of a web server. A successful connection to the appropriate port was a far better indicator than a simple network PING that an application was actually listening on the server. While this gave much greater confidence that the application was up, not to mention a tool that could differentiate between multiple applications on a single physical server by TCP port, a positive response was still not a definitive indication that the application was capable of handling end-user traffic.

The last major iteration of health monitoring was an "application health monitor," or a monitor that was capable of interacting with and interpreting the response from the back-end application. These monitors took another step beyond simply connecting to the application port, but interacted with the application itself. An FTP monitor, for instance, not only connected to the back-end FTP server, but attempted to download a known file or marker and then verify that the file was

correctly downloaded. These monitors also solved many other problems like being able to differentiate between multiple web sites hosted off of the same web server and knowing that some traffic (for the "bad" web site) shouldn't be forwarded to a specific physical server while other traffic (destined for the remaining "good" sites) could. By adding application intelligence to the monitor, it was now extremely confident that the back-end application receiving end-user traffic was capable of receiving, processing, and replying to that traffic.

Just like the basic load balancer has now become an application delivery controller, the health monitors used in ensuring high availability and reliability evolved from being network-centric to being application-centric, applying more intelligence along the way, as well as more processor, bandwidth, and time.

The Impact of Intelligent Health Monitoring

Much like the impact of adding intelligence to load balancing decisions— moving up the stack from layer 2/3 to layer 7—adding intelligence to health monitoring also negatively impacted the performance of both the ADC and the back-end applications themselves.

Using application layer health monitors, in effect, increased the number of connections and transactions that the back-end systems need to process. This not only replaced processor overhead previously alleviated by load balancing, but also complicated the capacity planning process by having to account for health monitoring overhead when calculating usage needs. It also increased the amount of network traffic on the intermediate networks between the load balancer and the application servers, further restricting the scalability of the application farm and driving the need for faster LAN connections in the data center.

Increasingly intelligent and application-specific monitors also negatively impacted the ADC itself. As the monitors continued to evolve, they took up more processor and bandwidth to execute as well as interpret the responses from the application. They also increased the utilization of connections between the load balancer and the application servers; again, limiting the scalability of the deployment and complicating capacity planning and utilization statistics.

Together, negatively impacting the applications as well as the load balancing device, health monitoring activity began to chew up significant amounts of resources that had been created by load balancing in the first place and decreased the scalability of the deployments—all contrary to the original decision to deploy load balancing. On more than one occasion, the customer's indiscriminant use of sufficiently advanced application health monitoring actually caused network and application crashes. The result was the need to manage health monitoring activities to balance between the benefits of application awareness with the side-effects of increased utilization. In many cases, this led organizations to abandon intelligent health monitoring altogether in lieu of less intelligent, but more optimized TCP or network health monitoring.

A New Way with F5 Networks

In keeping with the spirit of the F5 TMOS™ architecture, the first intelligent, full-proxy architecture capable of performing at network line speed, F5 announces the solution to the "intelligent" versus "performance" conundrum in health monitoring with intelligent, "passive" health monitors. Passive monitoring enables new levels of confidence without adding the additional overhead of traditional health monitoring.

Because BIG-IP® Local Traffic Manager™ (LTM) uses the full-proxy software architecture of TMOS, it is capable of not only using real-world data to intelligently route application traffic, but also of examining the real-world application responses for indication of application issues. For example, if a request made to connect to a specific service results in an error message, BIG-IP LTM can take that as a fairly definitive indication that the application is not capable of processing

the user request. BIG-IP LTM can then mark that specific service on that particular server as “down” and then initiate further active monitors to assess when the service is operational again.

In this way, BIG-IP LTM can intelligently determine the true state of application servers without injecting any further overhead in terms of server utilization, network utilization, or even ADC utilization from processing multiple health checks when the services are handling user-traffic just fine. Only when a service does something unexpected and fails to process real traffic are the active monitors processed—and then only for the malfunctioning system, not the dozens of systems that are operating as expected.

Passive monitors finally break the “intelligence” versus “performance” barrier giving BIG-IP customers the best of both worlds.