How to instrument your Java EE applications for a virtualized environment

Lori MacVittie, 2008-30-09

If you're excited about the automation capabilities of cloud computing and virtualization, you are going to love this solution. In a virtualized environment where applications can ostensibly be popping up all over, and applications are no longer tied to specific servers, there is a need to automatically manage these application instances in a high-availability (load balanced) environment. What you need is the ability to automagically add and remove application instances from the application delivery controller (load balancer) so you don't have to worry about tying those applications down, which could reduce the benefits typically associated with virtualization.

If you aren't going to a fully virtualized and automated data center, you might be happy to know that you can still reap the benefits of this automated solution. Not only is this solution perfect for a virtualized environment, it's also just as great for a non-virtualized environment for automating availability of applications. Truth be told, the application and the solution doesn't care (nor should it) whether it's running in a virtual image or not; it merely "is".

In a nutshell, when an application initializes, it adds itself to the appropriate application pool on the application delivery controller. When the application is destroyed, it removes itself. This means no matter where the application instance is living - in a virtual image, in a different servlet container, on a new server - it will automatically be "discovered" and immediately become part of the high availability pool of servers.

iControl, F5's service-enabled API providing configuration and management control of its solutions, can be used from within your Java EE application to enable automation of pool management on a BIG-IP application delivery controller.

This solution uses the Java Servlet 2.3 ServletContextListener interface. The ServletContextListener interface can be used to listen and react to a variety of servlet events, including application lifecycle. In order to automate the addition and removal of an application from the appropriate BIG-IP pool, we'll be listening for two events: contextInitialized and contextDestroyed. In the former, we'll add the application to the appropriate pool and in the latter, we'll remove it automatically.

This proactive approach to managing applications managed by BIG-IP LTM (Local Traffic Manager) also ensures that requests are not caught in between a monitor's health check interval, which can result in either an error or a second connection as part of a retry event within an iRule. This improves performance by ensuring that only active applications receive requests, and reduces connection attempts that can improve the efficiency of high-volume applications.

This is also an excellent method of automating availability for applications for which synthetic monitors are problematic.

You can read about the full solution with code in this article. Yes, they actually let me code from time to time.

Happy coding!

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com