

F5 Friday: The Rules for the Game of Application Performance Tag



Lori MacVittie, 2010-04-06

It's an integration thing.



One of the advantages of deploying an application delivery controller (ADC) instead of a regular old [Load balancer](#) is that it is programmable – or at least it is if it's an [F5 BIG-IP](#). That means you have some measure of control over application data as it's being delivered to end-users and can manipulate that data in various ways

depending on the context of the request and the response.

While an ADC has insight into the end-user environment – from network connection type and conditions to platform and location – and can therefore make adjustments to delivery policies dynamically based on that information, it can't gather the kind of end-to-end application performance metrics both business stakeholders and developers need. This data, the end-user view of application performance, is increasingly important. Back in April [Google](#) noted that “[page speed](#)” [would be incorporated into its ranking algorithm](#), with faster loading pages being ranked higher than slower pages. And we all know that the increasingly digital generation joining the ranks of corporate end-users have grown accustomed to fast, responsive web applications no matter what device they're using or from what location they're coming.

Application performance is a big deal.

The trick, however, is that you have to know how fast (or slow) your pages are loading before you can do something about them. That generally means employing the services of an application performance monitoring solution like [Keynote](#) or [Gomez](#) (now a part of [Compuware](#)). The way in which these solutions collect application performance data is through instrumentation of web applications, i.e. every page for which you want a measurement must be modified to include a small piece of Javascript that enables the reporting of performance back to the central system. Administrators and business stakeholders can then generate reports (and/or alerts) based on that data.

IT'S an INVESTMENT

The time investment required to instrument every page for which you (or the business) desire metrics is significant, especially if the instrumentation is occurring after deployment rather than as a part of the development life cycle process. Instrumentation at any time incurs the risk of error, too, as its generally done manually in a programmatic way. With any code-based solution – whether operational script or part of an application – there's always the chance of making a mistake. In the case of web applications that mistake can be more costly as its visible to the end-user (customer) and may cause the application to be “unavailable”. It's no surprise, then, that while most agree on the importance of understanding and measuring application performance, many organizations do not place a high priority on actually implementing a solution.

dynaTrace recently conducted a study on performance management in large and small companies. The quick facts paint a horrible picture. 60 percent of the companies admit that they do not have any performance management processes installed or what they have is ineffective. Half of the companies who answered that they have performance management processes admitted that they are doing it only in a reactive way when problems occur. One third of all companies said that management is not supporting performance management properly. From this data we can obviously conclude that performance management is not a primary interest in most companies.

Week 22 – Is There a Business Case for Application Performance?

Can only instrument external web performance management for those applications where you can modify code and tag.

For those applications you can modify, maintenance and management can explode with virtualization.

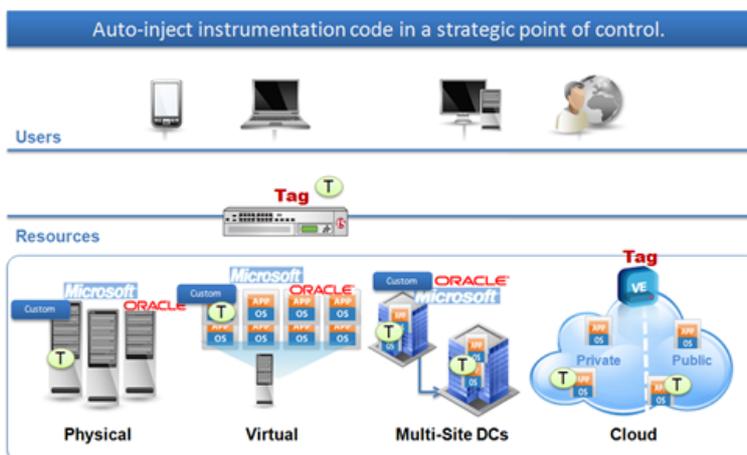
With multi-site and inter-cloud architectures it becomes exponential impacting flexibility and simplicity.

For those applications you can't instrument, you lose all context and control to measure real-time performance SLAs.

While the data from the dynaTrace study is interesting, it ignores how the reality of implementing APM solutions impact the ability and/or desire to deploy such solutions. One of the reasons many companies have no performance management processes is because they are unable to instrument the application in the first place. For some, it's because the application is "packaged"; it's a closed source, third-party application that can't be instrumented via traditional code-based extension. For others it may be the case that instrumentation is possible, but the application is frequently updated and instrumentation must be done with every update and necessarily extends the application development lifecycle.

AUTOMATED INSTRUMENTATION for ALL APPLICATIONS

It is at this point that the game of application performance "tag" comes in handy and alleviates the risk associated with manually instrumenting pages and enables the monitoring of packaged and other closed-source applications.



Even though we often refer to BIG-IP as a "load balancer" it really is an application delivery controller. It's a platform, imbued with the ability to programmatically modify application data on-demand. Using iRules as a network-side scripting solution, architects and developers and administrators can control and manage application data in real-time without requiring modification to applications. It is by leveraging this capability that we are able to automatically inject the instrumentation code required by Gomez to monitor application performance.

The BIG-IP "tags" application response data with the appropriate end-user monitoring code which enables the ability for APM providers like Gomez to monitor *any* application.

This is particularly interesting (and useful) when you consider the inherent difficulties of measuring performance not only from packaged applications but from an off-premise cloud computing environment. By deploying a virtual BIG-IP (a virtual network appliance) the same network-side script used to inject the appropriate client-side script to enable application performance monitoring can be used to instrument off-premise cloud computing deployed applications on-demand. This means consistent application performance measurements of applications regardless of location. Even if the application "moves" from the local data center to an off-premise cloud, the functionality can remain simply by including a virtual BIG-IP with the appropriate iRules deployed.

The GOMEZ SOLUTION

F5 has partnered with Gomez to make the process of implementing this instrumentation for their service almost trivial. The nature of iRules, however, makes it possible to duplicate this effort with any application performance monitoring solution which relies upon a client-side script to collect performance metrics from end-users. The programmatic nature of iRules is such that organizations will enjoy complete control over the instrumentation, including the ability to extend the functionality such that the injection of the Javascript happens only in certain situations. For example, it might be done based on the type of client or client network conditions. It might be only injected for a new end-user as determined by the existence of a cookie. It might be injected only for certain pages of an application. It can, essentially, be based on any **contextual** variable to which the ADC, the BIG-IP, has access.

You can read more details (including the code and the specific “rules” for implementation) in this tech tip from Joe Pruitt, “Automated Gomez Performance Monitoring” and you can download the full implementation in the [iRules CodeShare](#) under [GomezInjection](#).

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