

A question of

Scale

It takes a new breed of Application Delivery Controller to keep up with the demands of the Web 2.0 age.

As a senior analyst with industry analyst firm Enterprise Strategy Group, JON OLTSIK witnesses first-hand the struggles enterprises face trying to keep up with burgeoning traffic demands. Oltsik spoke with F5World Editorial Director Paul Desmond about how VIPRION, a new breed of Application Delivery Controller (ADC) from F5, can help stem the tide.

Q You've said web application use is "exploding" due to collaborative Web 2.0 technologies such as wikis, blogs, and content-rich documents. Do you expect this level of growth to continue?

A It's likely to continue and grow at an escalating pace. What we've seen is that the enterprise is intrigued but really hasn't completely bought into the Web 2.0 application model yet; that's going to happen in the next few years. From a performance perspective, this means providing the best experience for users will be that much more difficult, because applications are coming from everywhere with very little central coordination, except at the network layer.

Q Many ADCs claim to be able to help with the kind of performance issues that Web 2.0 and other web applications create. Where do they fall short in practice?

A It's a question of scale. A lot of the early ADCs were built for finite uses. The more we design network-based applications, the more content they use, the more locations they hit, the bigger the problem becomes, and the more scale you need. The solution today is to string together a lot of small ADCs. But anytime you try to solve an IT problem by piecing



together a bunch of point products in hopes of scaling up, it becomes cumbersome from both an operational and performance management standpoint. There are just too many knobs to turn. It's really difficult to get an aggregate view of what's going on.

Q What do you see as the solution?

A The ADC functions—things like load balancing, caching, and application layer intelligence—all have to work together. So we don't just need a bigger box, but a bigger, smarter box that can be a central point of control for all this network intelligence.

Q Do you think F5 has achieved that with VIPRION?

A I think it has. It's a function of bigger, faster hardware combined with what they've already done with TMOS. TMOS is able to take different, discrete functions and use software to layer them together, so you don't need an independent box for each function. Consequently, F5 is able

to virtualize intelligence with TMOS and also apply TMOS to a lot more hardware. You get a supercomputer effect: really fast hardware that's able to segment different tasks over different hardware processors and then aggregate everything together for total performance.

Q How do the benefits of VIPRION play out in practice?

A Maybe around the holidays you need lots of transaction speed for your web site, and then later, you want to shift some resources to a video that you're hosting. Those are two completely different application behaviors, but in order to maximize the user experience, you want some central way to manage the services that each application needs. Pooling also helps you eliminate the classic IT phenomenon of over-provisioning. If I can't manage my bandwidth, I buy more bandwidth. That gets expensive after a while. With VIPRION, because you're pooling functionality, you can adjust the box and add horsepower wherever you need it. You don't have to over-provision each function.

Q What other advantages do you see from a product like VIPRION?

A It enables you to create a flatter network configuration. If you have to route traffic through multiple boxes for the specific services each one provides, it really disrupts the way your traffic flows. With VIPRION, you have all the traffic going through one place. This makes the network flatter and easier to manage. It's also a classic example of consolidation. If you can consolidate IT functions, you end up saving on operations and capital costs. *