

# Business-Layer Load Balancing



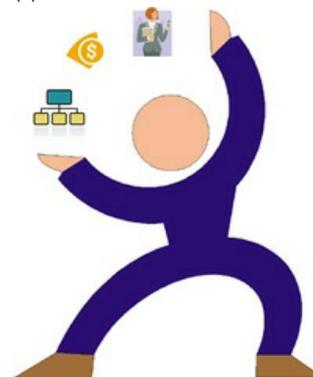
Lori MacVittie, 2009-15-07

One of the interesting points that discussions around [intercloud](#) brings up is the need for infrastructure to, if you'll pardon the use of marketing jargon, align with the business. What that really means is that applications and their supporting infrastructure need to be more *business-aware*.

Thing is you don't really need [intercloud](#) or even [cloud](#) or even [virtualization](#) for many of these business-aware capabilities. They are certainly a boon, but solutions that include application delivery functionality don't need to wait for a fully-baked [cloud](#) or [intercloud](#) implementation.

Consider, for example, the potential of business-layer [load balancing](#). What is that? The general premise is that application requests are routed based on business value, rather than only on technical parameters like capacity or response time or number of connections.

Rather the decision where to route a request is based on some business metrics: costs, value of a transaction, or something as esoteric as the number of people who are on vacation today. Assume that there are different business processes that need to execute based on the value of any given transaction. Perhaps there are more detailed logs that need to be written, or specific services that need to be executed as part of the process, or other "special handling" around high-value transactions. The result is that the "application" that handles high-value transactions may be a bit different than the one that handles transactions of nominal or low value, and there needs to be a way for such requests to be routed to that special application rather than the normal one.



## CONTENT-BASED ROUTING

What we really need in this situation is content-based routing (CBR). CBR is most often associated with [ESB](#) (enterprise service bus) implementations, as it is one of the primary value propositions of that messaging technology. But content-based routing is also a function of application delivery controllers, that is the advanced load balancer that is already responsible for assuring availability and scalability of your applications. In fact, this functionality at one time was often referred to as *content-switching*, but over time became more commonly known as [Layer 7 switching](#).

Regardless of what you call it, CBR or content-switching can be implemented in an [application delivery controller](#) as well as an [ESB](#). In fact, it might be *easier* to accomplish the implementation in an application delivery controller because the [network-side scripting](#) technology through which such capabilities are implemented is infinitely more elegant than the often times confusing service orchestration systems used by an [ESB](#) to produce the same effect. Removing an [ESB](#) from the data flow also eliminates the need to scale not only the applications but the messaging bus, which can result in a much simpler architecture.

Basically we're going to implement a [SOA Router Pattern](#) but we're going to add a bit more intelligence to the logic that determines how any given request is routed. Just to differentiate, let's call this the *Business-Value Router Pattern*. The actual logic isn't any more complex than a simple URI-based router pattern, but it does require that you be able to extract from the transaction a value upon which you can base routing decisions. Perhaps that's a simple form field called "total", or perhaps it will require some sort of calculation to determine. Either way, network-side scripting offers the flexibility to build this business rule with alacrity.

### SOA Router Pattern

```
when ( request_received ) {  
  value = HTTP URI  
  if (URI matches "/somevalue")
```

### Business Value Router Pattern

```
when ( request_received ) {  
  value = HTTP form field "total"  
  if (value > $1000)
```

```
    route to application_cluster_A
else
    route to application_cluster_B
}
```

```
    route to application_cluster_A
else
    route to application_cluster_B
}
```

Obviously this type of intelligence and business value doesn't require a cloud or intercloud, just the ability to extract the right information and use it to route requests. But it can also work in an intercloud or cloud environment, as the network server virtualization inherent in an application delivery platform abstracts the actual location (as well as whether it is virtual or traditional) of the application or service.

---

## THE INFRASTRUCTURE 2.0 ADVANTAGE

---

One of the core tenets of Infrastructure 2.0 is that it is *collaborative*. Integration via standards-based mechanisms ([service-enabled APIs](#), for example) provide the foundation for collaborative solutions that allow business layer metrics, measurements, and processes to direct infrastructure and applications to perform tasks that are focused on providing business value rather than just architectural value. That's the premise behind business-layer load balancing, too: providing business value above and beyond the architectural (and very IT-focused) benefits of a dynamic infrastructure.

The promise of cloud includes automation of IT processes, but many of those processes are in fact grounded in business metrics and rules that must be manually applied. Infrastructure solutions that are not enabled with collaborative capabilities end up being configured in a static and manual way that is operationally expensive. That is because every time a business rule or condition changes that would affect how the infrastructure should react, route, or otherwise manage application traffic, the configuration must be changed to match it. Manual configuration of such solutions is time consuming and inefficient. The automatic modifications and tweaks to configurations via integration capabilities makes such changes more efficient and can, when implemented to do so, even give the business owners control over how those rules are applied to the delivery of critical applications.

This kind of flexibility comes at a cost: it cannot be turn-key. It is nearly impossible for anyone to codify all the possible variables, systems, and processes that span business and IT organizations and thus any turn-key method of providing business-layer load balancing would be too narrow to be of real value. The time, effort, and money invested up-front to deploy and integrate infrastructure 2.0 solutions with IT and business layer systems pays off first in the reduction of operational costs as automation relieves the burden of manual configuration from the shoulders of overworked IT staff. The second payoff is, one hopes, in an increasing bottom line due to the ability of the business to adjust IT processes and systems based on changing needs.



- [Implementing SOA Patterns: The Router](#)
- [Cloud Balancing, Cloud Bursting, and Intercloud](#)
- [Not all application requests are created equal](#)
- [Layer 7 Switching + Load Balancing = Layer 7 Load Balancing](#)
- [You say "Late", I say "Delayed"](#)
- [Don't just balance the load, distribute it](#)

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

---

©2017 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113