The Critical Role of an Application Delivery Controller

Ashton, Metzler & Associates

Leverage Technology & Talent
for Success

Friday, October 30, 2009

Introduction

In any economic environment a company's senior management expects that their IT organization will continually look for ways to cut cost, get better control over the company's data assets, and show a high rate of return on the investments that they make. The current economic environment has significantly increased those pressures. One of the initiatives that many IT organizations have taken in order to respond to senior management's expectations is to consolidate resources, such as applications, servers and storage into centralized data centers. Due to the economy of scale that occurs upon consolidating resources, this initiative reduces cost. This initiative also enables an IT organization to have better control over who has access to a company's data and to be better able to enforce security policies.

However, before these resources were consolidated, most data traffic transited a high-speed, low latency LAN. After resource consolidation, most data traffic transits a relatively low-speed, high latency WAN. In addition, in many cases a chatty protocol such as Common Internet File System (CIFS), which was designed to run over a LAN, now runs over a WAN. A chatty protocol requires tens, if not hundreds of round trips, to complete a single transaction. The combination of factors such as these often results in unacceptable application performance.

Another initiative that many IT organizations have taken in order to respond to senior management's expectations is to deploy new Web-based applications and to implement Web-based interfaces on existing applications. The resulting applications are accessed via a Web browser over a network such as the Internet or an intranet. Among the advantages of Web-based applications is the fact that it eliminates the need to install and manage clients on each access device. Web-based applications utilize protocols such as HTTP. The good news is that HTTP is not as chatty as is CIFS. The bad news is that is common for a Web application to use fifty or more objects to create a single page. It is also common to have each object require ten or more packets be transmitted. The result of having to transmit hundreds of packets in order to down load a single Web page is unacceptable delay and user frustration.

Resource consolidation and the deployment of Web-based applications are just two examples of IT initiatives that provide significant value, but which often result in unacceptable performance. As will be shown in this brief, in order to overcome these performance challenges, IT organizations need to implement an Application Delivery Controller (ADC). This brief will also discuss criteria that IT organizations should use in order to choose an appropriate ADC.

The Role of an ADC

While an ADC is a relatively new class of product, similar product categories have existed for forty years. In particular, the genesis of the ADC dates back to the IBM mainframe-computing model of the late 1960s and early 1970s. Like today's servers, the IBM mainframe of that time period was a general-purpose computer. That means that it was designed to perform virtually any task. However, the downside of being designed to perform virtually any task is that there were certain tasks that the IBM mainframe did not perform well. For example, the IBM mainframe was ineffective at performing computationally intensive communications processing tasks, such as terminating WAN links and doing the associated protocol conversions. It is not the case that the IBM mainframe could not perform these tasks. However, performing these tasks consumed an inordinate amount of compute resources, which meant that there was few, if any, compute resources left to process business transactions.

In order to improve performance and maximize their IT investment, most large IT organizations of that era implemented a computing model that included having a Front End Processor (FEP) reside in front of the IBM mainframe. The primary role of the FEP was to free up processing power on the general purpose mainframe computer by performing communications processing tasks in a device that was purpose-built for these tasks.

While some IT organizations still use mainframe computers, most IT organizations have implemented server farms. A server farm is a group of servers that are networked together with the goal of meeting requirements that are beyond the capability of a single server. One of the challenges associated with implementing a server farm is to ensure that a request for service is delivered to the most appropriate server. There are many ways to define what the phrase *most appropriate server* means. Certainly the server has to be available. Ideally, the most appropriate server is the server that is processing the lightest load of any member of the server farm.

In order to ensure that a client's service requests are delivered to the most appropriate server, a new product category was developed: the server load balancer (SLB). As shown in Figure 1, an incoming request is directed to an SLB. Based on parameters such as availability and the current server load, the load balancer decides which server should handle the request and forwards the request to the selected server.

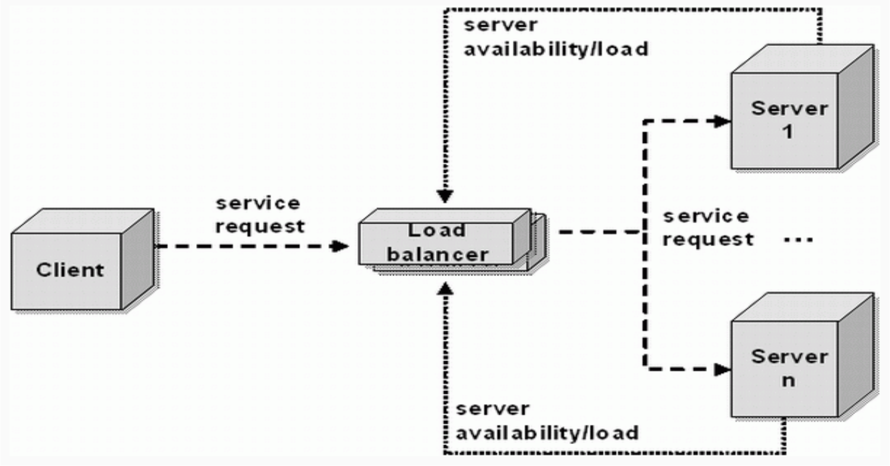

Figure 1: The Role of an SLB

The value of an SLB is that it enables an IT organization to add servers to a server farm and be confident that the expanded server farm will be able to process a proportionate number of additional transactions. As is explained below, this is why an SLB is referred to as a linear solution.

This confidence that IT organizations have in an SLB stems from the fact that as previously mentioned, the SLB ensures that a given service request is handed off to a server that is relatively lightly loaded and hence has the resources to respond to the request. Unlike a FEP, however, an SLB does not offload any processing from the servers and hence does not increase the overall ability of an individual server to process transactions. As such, an SLB can be considered to be a linear solution in that it enables an IT organization to get a *linear benefit* out of its servers. By *linear benefit* is meant that if an IT organization increases the number of servers in a server farm that is front-ended by an SLB by a given percentage, that the number of transactions that the server farm can support should increase by roughly the same percentage. For example, if an IT organization doubles the number of servers in a server farm that is front-ended by an SLB, then the server farm should be able to support roughly double the number of transactions.

It is helpful to look at an ADC as a combination of an SLB and a FEP. It is like an SLB in that as shown in Figure 2, it sits in front of a server farm and receives service requests from clients and delivers the requests for service to the most appropriate servers.

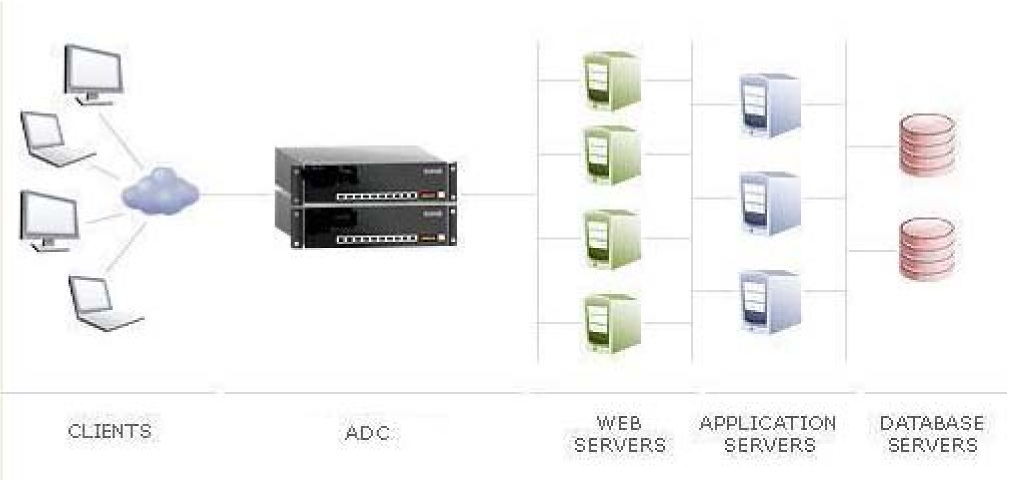

Figure 2: The Role of an ADC

However, an ADC is also like an FEP in that it was purpose-built to be able to offload computationally-intensive, communications processing off of the servers and hence and ADC can significantly increases the overall ability of a server to process transactions. Because it combines the functionality of a FEP and an SLB, the ADC accelerates application performance, and increases both security and server efficiency.

ADC Functionality

This section of the brief will describe TCP offload and some other key functionality that IT organizations should look for in an ADC.

TCP Offload and Multiplexing

In Web environments, supporting TCP session maintenance places a significant burden on servers. This burden increases as the setup/teardown rate and the number of simultaneous connections increases. For example, in a high throughput environment, a server can spend up to 90% of its CPU resources just dealing with TCP connections. While persistent connections improve the way HTTP and TCP work together, they end up benefiting the client more than they do the server.

In order to enable servers to spend the bulk of their resources doing useful work (e.g. serving up objects and processing transactions), an ADC must support a technology known as TCP offload and multiplexing. The role of TCP offload and multiplexing is to minimize the overhead associated with TCP session maintenance, both with regards to the overhead associated with setting up and tearing down sessions, as well as the overhead that is associated with maintaining a large number of sessions. By performing this role, the ADC maximizes the amount of useful work that can be performed by a server.

In order to perform TCP offload and multiplexing, the ADC functions as an intermediary between the client base and the server. In this role, the ADC acts as the server towards the clients, while simultaneously acting as a client to the server. The ADC assumes the responsibility of dealing with all client-side connections and it also consolidates the front-end object requests from the many client-side connections onto a few server-side connections. The connections on the server side are sessions with a long lifetime that are used repeatedly as new object requests are received.

The benefits of TCP offload and multiplexing include the fact that:

- Since the ADC handles all client side connections, the server is no longer tasked with rapidly setting up and tearing down connections.

- Because of connection consolidation, the number of simultaneous TCP sessions a server has to deal with is drastically reduced.

- The ADC shields the server from client-side connection weaknesses. This means the server resources are not affected by any delay, congestion, or packet loss in the client connection. Therefore, the few connections that the server does operate through can perform at minimum overhead, and hence maximum the performance of the server.

SLB and Global SLB

As noted, an ADC sits in front of a server farm and receives service requests from clients and delivers the requests for service to the most appropriate servers. As such, an ADC functions as a traditional SLB. In addition, an ADC can function as a global server load balancer (GSLB). In this role the ADC balances the load across geographically dispersed data centers by sending a service request to the data center that is exhibiting the best performance metrics.

Secure Sockets Layer (SSL) Offload

SSL is widely used to encrypt documents that are being sent over the Internet. Unfortunately the processing of the security mechanisms that are associated with SSL is computationally-intensive. As a result, a key function of an ADC is SSL offload. As part of SSL offload, the ADC terminates the SSL session and performs the computationally-intensive processing of the security mechanisms thus freeing up server resources.

Scripting

One of the characteristics of most IT environments is that the environment is comprised of a large and growing number of servers and applications. Another characteristic is that most IT organizations have very limited control as to which users access which applications and which servers. An ADC gives control to the IT organization through functionality sometimes referred to as scripting, and sometimes referred to as a rules engine. This functionality allows the IT organization to directly classify and modify the traffic of any IP-based application.

Compression

The basic role of compression is to reduce the amount of traffic that has to be transmitted over the WAN. One of the factors that impacts how effective compression will be is the type of traffic. For example, applications that transfer highly redundant data, such as text and html on web pages, will benefit significantly from compression. How the compression is implemented also impacts its effectiveness. For example, compression that is performed in hardware is likely to be more effective than compression that is implemented in software.

Caching

If an ADC supports caching, new user requests for static or dynamic Web objects can often be delivered from a cache inside of the ADC rather than having to be regenerated by the servers. Caching therefore improves user response time and minimizes the loading on Web servers, application servers, and database servers.

ADC Selection Criteria

As mentioned, the deployment of an SLB enables an IT organization to get a *linear benefit* out of its servers. Unfortunately, the traffic at most Web sites is not growing at a linear rate, but at an exponential rate. To exemplify the type of problem this creates, assume that the traffic at a hypothetical company's (Acme) Web site doubles every year. If Acme's IT organization has deployed a linear solution, such as an SLB, after three years it will have to deploy eight times as many servers as it originally had in order to support the increased traffic.

As was also mentioned, an ADC is like an FEP in that it offloads computationally-intensive, communications processing off of the servers and hence significantly increases the overall ability of a server to process transactions. If Acme's IT organization were to deploy an ADC then after three years it would still have to increase the number of servers it supports, but only by a factor of two or three – not a factor of eight.

The preceding section discussed some of the functionality that IT organizations should look for in an ADC. However, as is the case with most classes of networking products, over time there is not a significant difference in the features provided by competing products. This means that when choosing an ADC, IT organizations should pay the most attention to the ability of the ADC to have all features turned on and still support the peak traffic load. Referring back to the Acme example, it was the ability of Acme's ADC to offload numerous computationally-intensive tasks from the servers and still process the peak traffic load that enabled Acme to not have to deploy eight times as many servers in three years in order to support the increase in the peak traffic load.

If an ADC cannot support the peak traffic load with all features turned on, in addition to not maximizing the performance of the servers, the ADC becomes a bottleneck. While it is always important to try to avoid having bottlenecks, that is particularly true in the data center as a bottleneck here negatively impacts the experience that a large number of users have with multiple business critical applications. As part of the evaluation process, IT organizations need to ensure that they will not deploy an ADC that quickly becomes a bottleneck. To do this, IT organizations should stress test the ADCs that they are evaluating at traffic loads that are well beyond their current traffic peak traffic loads.

While it is critical that an ADC can support the peak traffic load with all features turned on, this is not an easy task to accomplish. The key-determining factor relative to the ability of an ADC to support peak load with all features turned on is the internal architecture of the ADC. In particular, the ADC must perform the most computationally-intensive tasks, such as TCP offload, in hardware that is itself purpose-built for the task.

Summary and Call to Action

A company's senior management expects that their IT organization will continually look for ways to cut cost, get better control over the company's data assets, and show a high rate of return on the investments that they make. Unfortunately, some of the steps that IT organizations have taken in order to achieve those goals, such as consolidating servers and implementing Web-based applications, have lead to unacceptable performance.

One of the ways that IT organizations improved performance in the traditional mainframe environment was to offload computationally intensive processing onto a device that was purpose-built to perform that processing: the FEP. More recently, IT organizations began to deploy an SLB in front of their server farms. An SLB can improve the performance of the server farm by ensuring that a service request is not handed off to a heavily loaded server if there is a more lightly loaded server available. However, since the SLB does not offload any processing from the servers, it provides only a linear benefit.

A solution that only provides a linear benefit will not be able to cost effectively support traffic that is growing exponentially. The solution to this problem is to deploy a device that is purpose-built to offload processing from each member of a server farm: the ADC. By offloading this processing onto a purpose-built device, an IT organization can significantly reduce the number of servers that they need. As a result, the IT organization saves the cost of the servers that would have had to be purchased, the associated software licenses and maintenance fees, as well as the cost of the related power and cooling.

When choosing an ADC, IT organizations need to understand the features that it supports. However, as this class of product continues to mature, the distinction between the features provided by competing products is lessening. This means that when choosing an ADC, IT organizations should pay the most attention to the ability of the ADC to have all features turned on and still support the peak traffic load, as this is what determines the ability of the ADC to cost effectively support exponential traffic growth. In order to support the peak traffic load with all features turned on, the ADC must perform the most computationally intensive tasks, such as TCP offload, in hardware that is itself purpose-built for the task.