Primary Functionality Provided by an Application Delivery Controller

Ashton, Metzler
& Associates

Leverage Technology & Talent
for Success

Background

While ADCs were initially positioned primarily as a Server Load Balancer (SLB), the ADC has assumed, and will most likely continue to assume, a wider range of more sophisticated roles that enhance server efficiency and provide asymmetrical functionality to accelerate the delivery of applications from the data center to individual remote users. In particular, the ADC can allow a number of compute-intensive functions, such as SSL processing and TCP session processing, to be offloaded from the server. Server offload can increase the transaction capacity of each server and hence can reduce the number of servers that are required for a given level of business activity.

ADC Functionality

Among the functions users can expect from a modern ADC are the following:

- ***Traditional SLB***

  ADCs can provide traditional load balancing across local servers or among geographically dispersed data centers based on Layer 4 through Layer 7 intelligence. SLB functionality maximizes the efficiency and availability of servers through intelligent allocation of application requests to the most appropriate server.

- ***SSL Offload***

  One of the primary roles played by an ADC is to offload CPU-intensive tasks from data center servers. A prime example of this is SSL offload, where the ADC terminates the SSL session by assuming the role of an SSL Proxy for the servers. SSL offload can provide a significant increase in the performance of secure intranet or Internet Web sites. SSL offload frees up server resources which allows existing servers to process more requests for content and handle more transactions.

- ***XML Offload***

  XML is a verbose protocol that is CPU-intensive. Hence, another function that can be provided by the ADC is to offload XML processing from the servers by serving as an XML gateway.

- ***Application Firewalls***

  ADCs may provide an additional layer of security for Web applications by incorporating application firewall functionality. Application firewalls are focused on blocking the increasingly prevalent application-level attacks. Application firewalls are typically based on Deep Packet Inspection (DPI), coupled with session awareness and behavioral models of normal application interchange. For example, an application firewall would be able to detect and block Web sessions that violate rules defining the normal behavior of HTTP applications and HTML programming.

- ***Denial of Service (DOS) Attack Prevention***

  ADCs can provide an additional line of defense against DOS attacks, isolating servers from a range of Layer 3 and Layer 4 attacks that are aimed at disrupting data center operations.

- ***Asymmetrical Application Acceleration***

  ADCs can accelerate the performance of applications delivered over the WAN by implementing optimization techniques such as reverse caching, asymmetrical TCP optimization, and compression. With reverse caching, new user requests for static or dynamic Web objects can often be delivered from a cache in the ADC rather than having to be regenerated by the servers. Reverse caching therefore improves user response time and minimizes the loading on Web servers, application servers, and database servers.

  Asymmetrical TCP optimization is based on the ADC serving as a proxy for TCP processing, minimizing the server overhead for fine-grained TCP session management. TCP proxy functionality is designed to deal with the complexity associated with the fact that each object on a Web page requires its own short-lived TCP connection. Processing all of these connections can consume an inordinate amount of the server's CPU resources. Acting as a proxy, the ADC offloads the server TCP session processing by terminating the client-side TCP sessions and multiplexing numerous short-lived network sessions initiated as client-side object requests into a single longer-lived session between the ADC and the Web servers. Within a virtualized server environment the importance of TCP offload is amplified significantly because of the higher levels of physical server utilization that virtualization enables. Physical servers with high levels of utilization will typically support significantly more TCP sessions and therefore more TCP processing overhead.

  The ADC can also offload Web servers by performing compute-intensive HTTP compression operations. HTTP compression is a capability built into both Web servers and Web browsers. Moving HTTP compression from the Web server to the ADC is transparent to the client and so requires no client modifications. HTTP compression is asymmetrical in the sense that there is no requirement for additional client-side appliances or technology.

- ***Response Time Monitoring***

  The application and session intelligence of the ADC also presents an opportunity to provide real-time and historical monitoring and reporting of the response time experienced by end users accessing Web applications. The ADC can provide the granularity to track performance for individual Web pages and to decompose overall response time into client-side delay, network delay, ADC delay, and server-side delay.

- ***Support for Server Virtualization***

Once a server has been virtualized, there are two primary tasks associated with the dynamic creation of a new VM. The first task is the spawning of the new VM and the second task is ensuring that the network switches, firewalls and ADCs are properly configured to direct and control traffic destined for that VM. For the ADC (and other devices) the required configuration changes are typically communicated from an external agent via one of the control APIs that the device supports. These APIs are usually based on SOAP, a CLI script, or direct reconfiguration. The external agent could be a start-up script inside of the VM or it could be the provisioning or management agent that initiated the provisioning of the VM. The provisioning or management agent could be part of an external workflow orchestration system or it could be part of the orchestration function within the hypervisor management system. It is preferable if the process of configuring the network elements, including the ADCs, to support new VMs and the movement of VMs within a data center can readily be automated and integrated within the enterprise's overall architecture for managing the virtualized server environment.

When a server administrator adds a new VM to a load balanced cluster, the integration between the hypervisor management system and the ADC manager can modify the configuration of the ADC to accommodate the additional node and its characteristics. When a VM is de-commissioned a similar process is followed with the ADC manager taking steps to ensure that no new connections are made to the outgoing VM and that all existing sessions have been completed before the outgoing VM is shut down.

For a typical live VM migration, the VM remains within the same subnet/VLAN and keeps its IP address. A live migration can be performed between data centers as long as the VM's VLAN has been extended to include both the source and destination physical servers and other requirements regarding bandwidth and latency are met.

In the case of live migration, the ADC does not need to be reconfigured and the hypervisor manager ensures that sessions are not lost during the migration. Where a VM is moved to a new subnet, the result is not a live migration, but a static one involving the creation of a new VM and decommissioning the old VM. First, a replica of the VM being moved is created on the destination server and is given a new IP address in the destination subnet. This address is added to the ADC's server pool, and the old VM is shut down using the process described in the previous paragraph to ensure session continuity.