About load balancing in Wowza Streaming Engine

When two or more instances of Wowza Streaming Engine™ media server software are used together in a cluster, a load balancer distributes the load between each server. Load balancing is an essential part of any streaming solution that uses more than one Wowza Streaming Engine server for playback. There are various types of load balancing solutions that can be used individually or together, depending on how large and distributed the server cluster is. And depending on the type of streaming, some load balancing solutions are better than others. Your workflow and type of streaming will dictate which is the best choice for you.

Load balancing requirements

A load balancer for a cluster of Wowza Streaming Engine servers should meet as many of the following requirements as possible.

- You can add or remove Wowza Streaming Engine servers easily from the cluster. Ideally, this would be automatic so that server failures could be detected easily.
- The load balancer can handle persistent and non-persistent connections. Persistent connections include connections to other Wowza Streaming Engine servers or to the Wowza GoCoder™ app) over WOWZ™ and RTMP protocols. Non-persistent connections include most HTTP connections and RTMPT connections.
- The system can determine the load on all Wowza Streaming Engine servers in the cluster so that accurate balancing decisions can be made.
- The system can handle server failures quickly and efficiently.
- The system doesn’t disrupt or delay the stream traffic.

Load balancing options

Wowza Load Balancer

Wowza Load Balancer is a free AddOn that can be run on any Wowza Streaming Engine server. It consists of two parts. A Listener runs on a Wowza Streaming Engine server that’s designated as the Load Balancer, which receives the initial player requests and automatic status updates from the other servers in the cluster. A Sender runs on each of the servers in
the cluster and sends current load information to the Listener.

When a player connection is made, it initially connects to an application on the Listener. The connection is automatically redirected using a standard 302 redirect to the correct server in the cluster. The player then disconnects from the Listener application and connects to the streaming application on the server to which it is redirected. In order for this to work, the player must be able to handle the 302 redirect. For players that can't handle the redirect, there's an HTTP interface that can be called to get the current redirect address.

Wowza Load Balancer won't work with encoders that can't handle the redirect, so if balancing encoder connections is needed, a different solution may have to be used.

Because Wowza Load Balancer is built on top of Wowza Streaming Engine, you can extend it by using custom Java classes to allow for very fine control over how the connections are balanced.

**Round-robin DNS and geo-based DNS load balancers**

Domain Name System (DNS) servers that support round-robin DNS or geo-based DNS can be used in some cases to load balance Wowza Streaming Engine servers. They usually work adequately for persistent connections, but sometimes cause problems for non-persistent connections. This is because a non-persistent connection performs a new DNS lookup for every request and may receive an address for a different server when it reconnects. DNS-based load balancing only provides rough balancing to the server cluster because there isn't any way for the DNS server to know the current load on the clustered servers. Also, the built-in caching mechanism associated with DNS can cause some servers to receive more connections than others. If a failure occurs, it may not be detected until the cached DNS request expires.

**Third-party load balancers**

Most third-party load balancers work by forwarding the TCP or UDP packets from client-side connections to the correct Wowza Streaming Engine server in the cluster, similar to how a router forwards packets on a network. All of the network traffic from the client runs through the load balancer. Some load balancers have different options for the traffic returning from the server cluster to the client. In most cases, this traffic consists of large media packets, so care must be taken to select an option that doesn't cause the load balancer to delay the returning traffic.

Because all traffic from the client side passes through the load balancer, the load balancer can monitor the load on the server cluster fairly accurately. Separate monitoring of the cluster is used to detect failures.
Normally, the load balancer is completely transparent to the client application; however, care must be taken when configuring the load balancer for non-persistent connections to ensure that the connections are always redirected to the correct server. In most cases, there’s a setting to handle these.

Notes

- In some cases, it may be beneficial to use multiple types of load balancing. For example, round-robin DNS can be used to direct connections to a pair of Wowza Streaming Engine load balancers or third-party load balancers that are then used to balance the server cluster.
- Geo-based DNS load balancers can be used to send connections to a specific data center where Wowza Streaming Engine load balancers or third-party load balancers are used.
- Either DNS-based load balancing method can also be used to distribute encoder connections to multiple Wowza Streaming Engine origin servers.

More resources

- Get the Dynamic Load Balancing AddOn for Wowza Streaming Engine