Connection Multiplexing

Understanding Array TCP Offload Technology

White Paper

Overview

When the number of connections coming to a server increases, eventually the server runs out of network resources. This does not necessarily mean the server doesn't have enough horsepower, memory, or bandwidth available, it simply means that the server's TCP/IP implementation wasn't meant for such a high rate of connection turnover. Enter connection multiplexing. Learn about:

- ➔ How connection multiplexing converts a large number of short connections to a fewer number of high-throughput connections
- ➔ Using connection multiplexing to take advantage optimized bulk throughput server settings without changing configurations or content
- ➔ Doing more with less by implementing connection multiplexing technology

ArrayNetworks®
The Application Networking Company

254 E. Hacienda Avenue
Campbell, CA 95008
www.arraynetworks.net
(408) 378-6800

Connection Multiplexing

Understanding Array TCP Offload Technology

➔ Introduction

It was a Friday afternoon that the problem first came up. Array Networks had just moved into their first real office, a mere two months after being born when an engineer wanted to show off his latest performance improvements. Unfortunately, he found himself frustrated that the few servers he had were not enough to demonstrate the increases.

The engineer spoke with Lawrence Lu, Array Networks' founder and CTO, about purchasing additional servers. With tight budgets and no outside investors yet, Lawrence had to say no to the request for additional servers.

Thinking about the problem some more, Lawrence realized that his future customers would face similar situations as well. Going on an all-weekend hacking spree, the two designed and implemented a new feature: Connection Multiplexing.

With the new feature in place, Lawrence and his engineering team was able to demonstrate significant performance improvements using the same handful of servers built out of leftover parts.

Marketing was impressed. A new feature was born.

➔ How Connection Multiplexing Works

Typical network topologies put Array application front end appliances between web clients and web servers. In these situations, the Array appliance acts as an intermediary by accepting connections on behalf of the servers and then passing the connection onward if it meets the necessary security criteria. (i.e. the packet is not part of a DoS attack, etc.)

Connection Multiplexing

The result of such a configuration is a 1-to-1 correlation between connections sent by users and connections received by web servers. While this may work ok in low volume traffic, servers are often not designed to support a large number of concurrent connections with a high rate of connection turn over.

When the number of connections coming to a server increases, the server eventually runs out of network resources. This does not necessarily mean the server doesn't have enough horsepower, memory, or bandwidth available, it simply means that the servers' TCP/IP implementation wasn't meant for such a high rate of connection turnover.

This situation is made worse by the fact that most TCP/IP stacks are not tuned for a large number of short connections, but rather for bulk throughput.

Enter connection multiplexing.

The purpose of connection multiplexing is to convert a large number of short connections into a few connections needing more throughput. This allows us to take advantage of the server's optimized bulk throughput settings without changing any configurations or content.

Connection multiplexing works by taking advantage of a feature in HTTP/1.1 that allows for multiple HTTP requests to be made over the same TCP connection. So instead of passing each HTTP connection from the client to the server in a 1- to-1 manner, the Array appliance combines many separate HTTP requests from clients into relatively few HTTP connections to the server. The Array keeps the connections to the server open across multiple requests thus eliminating the high turnover typically encountered in high volume web sites.

Net result: higher performance out of the same servers *without* any changes or improvements to the server infrastructure.

➔ Technical Details of Connection Multiplexing

Array application front end appliances work as a full proxies thereby allowing them to perform connection multiplexing. With this feature, the Array pre-opens several connections to the origin server and keeps them open. As connections come from various clients, the Array attempts to reuse the pre-opened connections. The Array will keep the connection to the server open until it receives a "`Connection: close`" HTTP header or TCP FIN packet from the server.

Because of connection multiplexing, the Array appliance is able to take advantage of the Van Jacobson Header Prediction found in most TCP stacks (including Microsoft Windows 2000/.Net, Solaris, Linux, and the many flavors of BSD Unix). This feature of TCP stacks provides a fast path to move packets from the core operating system (kernel space) to the web server application itself (user space) so long as each packet meets the following requirements:

- The packet's associated connection is established
- The SYN, FIN, RST, or URG flags must not be set
- The packet must acknowledge previously sent data
- The packet is in order
- The TCP window size has not changed

Tests performed by W. Richard Stevens (author of the TCP/IP Illustrated Series, the now de-facto definition of TCP/IP) found that header prediction worked between 97-100% of the time in LAN environments, which is typically the case between Array appliances and servers.

Another important benefit of connection multiplexing is that it removes the burden placed on servers to turn over new connections very frequently – a process whose code path is often not as optimized as the path of a packet that meets the header prediction requirements. Furthermore, by not having to turn over new connections, the server is less likely to burn up its ephemeral ports by placing them into the TIME_WAIT state.

Because connection multiplexing removes the one to one correlation between client connections and server connections, the Array appliance inserts the X-Forwarded-For HTTP header into each request. This header has one parameter, the client IP's address. Modules for IIS and Apache web servers are available from Array Networks to set the client IP's address in the necessary data structures, logging systems, and CGI environment variables. These modules are available with source code to any Array customer.

If the administrator chooses not to have connection multiplexing enabled, the Array can transparently pass the client IP address in the IP header of each packet thereby allowing the origin server to work as if it were directly communicating with the client browser.

Note that connection multiplexing and client IP transparency are mutually exclusive. This is because it is not possible to spoof the client IP address from one client, keep the same connection open, and then spoof another client IP for a second request. The best alternative is the insertion of the X-Forwarded-For header with the true client IP on a per request basis.

➔ Summary

Connection multiplexing is an easy way for to reduce load on Web servers without needing to make any drastic infrastructure changes to either the network or the servers. Array Networks makes this technology easy to deploy and easy to administer in all application front end products.

About Array Networks

Array Networks is a world leader in secure application acceleration and deployment appliances for global enterprises. Built upon the Array SpeedStack(TM) technology, Array's unified secure content access solutions enable industry-leading performance, integration, scalability and ease of implementation and management. Headquartered in Campbell, California with sales offices in the U.S., Europe, Asia Pacific and Latin America, Array engineers and manufactures its products in the Silicon Valley and sells them through direct and indirect channels across the globe.

Array Networks, Inc.
254 East Hacienda Avenue
Campbell, CA 95008
Phone: (408) 378-6800
Toll Free: 1-866-MY-ARRAY
Fax: (408) 874-2753
Email: info@arraynetworks.net
www.arraynetworks.net