

## WEBWORKING

### Networking with the Web in mind

- **SERVERS: THE PRESSURE IS ON**
- **CREATING WEB SERVER FARMS**
- **BACK TO THE FUTURE WITH WEB DATA CENTERS**
- **WEB TRAFFIC CONTROL SYSTEMS - OUT OF CONTROL**
- **WEB SWITCHING: THE NEXT STEP IN NETWORKING FOR THE WEB**
- **DISSECTING THE WEB SWITCH**
- **CONSIDERING WEB SWITCH DESIGN**

**Alteon WebSystems, Inc.**

50 Great Oaks Boulevard  
San Jose, California 95119  
408-360-5500  
408-360-5501 fax

<http://www.alteon.com>

The need to create large-scale websites is no longer isolated to a few large portals and ISPs. Every corporation must now build and maintain a major Web presence, not just to market and sell products but to support internal operations, communicate with partners, and conduct real-time business transactions. This far-reaching "Webification" trend has a profound impact on today's application architecture and the underlying server and network infrastructure requirements.

### SERVERS: THE PRESSURE IS ON

The "thinness" of the browser function, which enabled its widespread use, has driven Web servers to take on increased processing burden. The Web's ubiquity also forces servers to support more users than ever and be available at all times. As a result, scaling servers while maintaining zero-downtime has become a key-determining factor of a website's performance. Server clustering is the simplest solution to boost server capacity and minimize the impact of individual server failures. To facilitate server clustering, application functions can be partitioned into a multi-tiered architecture (see Figure 1).

In this model, the data server provides the functions associated with querying, managing, and recording data. The application server uses the data, processes client requests and formulates responses and so on. The Web server offloads user communications from the application server.

With this architecture, servers optimized for each task can be deployed in the appropriate tier to maximize overall performance. Bottlenecks are more easily identified and fixed by scaling the particular tier that is slow. Such scaling is most commonly achieved by grouping a number of physical servers to create a single, large virtual server using a server load balancer. For example, as shown in Figure 2, if the Web server layer is the bottleneck, multiple Web servers can be used with an appropriate Web server load balancing solution to distribute traffic across the Web servers.

### CREATING WEB SERVER FARMS

With clustering, applications become much more scalable, but management of the websites becomes more complex. Large Internet sites may have tens or hundreds of servers connected together in a hierarchical fashion, with the numbers, types and mix of servers changing over time due to requirements to offer new services, support new content, or adapt to changing site usage patterns.

FIGURE ONE

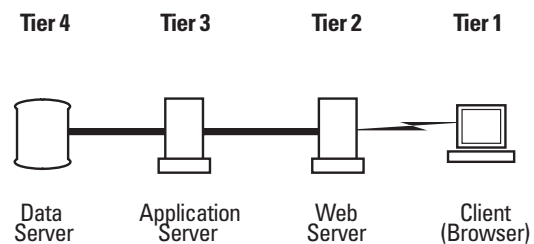
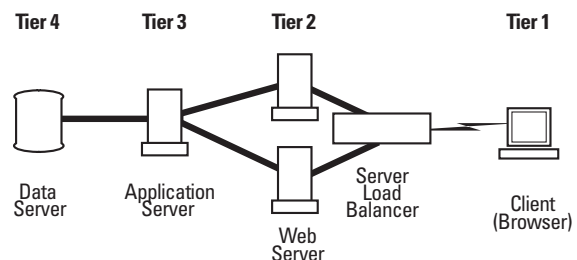


FIGURE TWO



The operational aspects of managing and securing these servers and related applications have resulted in physical co-location of servers into Web server farms. Relative to considerations like mission-criticality, traffic flow convergence, security and operations, the Web server farm can be viewed as today's mainframe. As it was with mainframes, it makes sense to protect the Web server farms by consolidating them into one or a number of "Web data centers" with the proper environmental support, physical access control, etc.

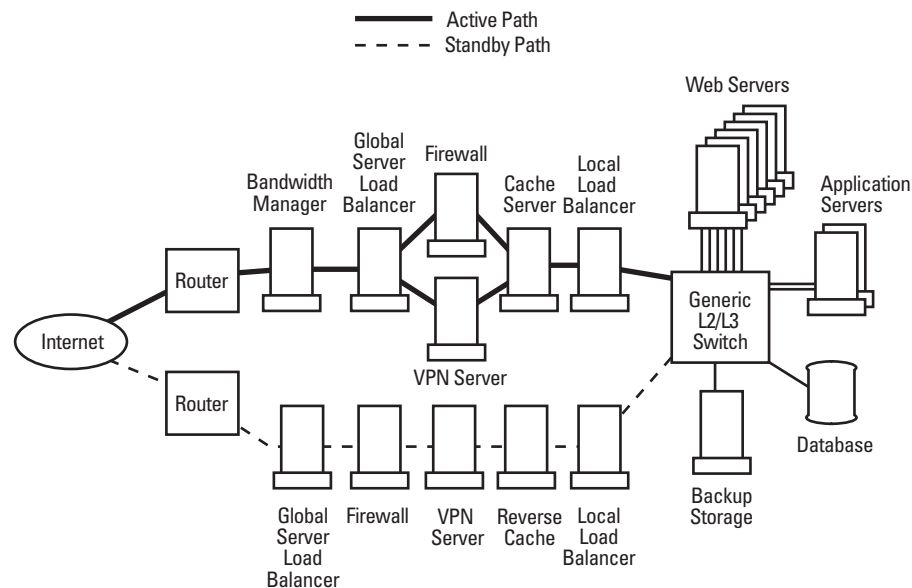
### BACK TO THE FUTURE WITH WEB DATA CENTERS

With mainframes, the responsibility for networking, traffic management, access privilege control, application-to-user session management, collecting accounting data, and enforcing quality-of-service policies all reside at the Front End Processors. The system worked but was slow and inflexible.

Web-based application environments are more flexible and scalable. But many of the control mechanisms of the mainframe environment must be re-established using a new set of functions that can collectively be called Web traffic management. Because the need for these functions emerged at different times and the technology matured at a difference pace, Web traffic management functions today reside on a number of discrete platforms.

For example, all or most of these devices: firewalls, proxy caches, local load balancers, distributed load balancers and bandwidth managers are typically found in a Web data center between the WAN routers and Web servers. (see Figure 3)

FIGURE THREE



### WEB TRAFFIC CONTROL SYSTEMS - OUT OF CONTROL

With server clustering, the scalability challenge of a Web data center has now fallen on the discrete Web traffic management devices. The majority of these devices are so called "IP appliances," a term that describes single function devices for which specialized software was written using a standard operating system (e.g. BSD UNIX),

then pre-loaded onto a standard Intel platform before delivery to customers. These IP appliances were attractive at first because features could be rapidly developed in a ready-made operating system and they could be built using low-cost, commodity Intel-based hardware.

However, as traffic has grown in size and complexity, IP appliances have shown serious performance limitations. When more and more features are added, each requiring a slice of the single CPU in the device, performance degrades dramatically. At the same time, traffic loads is exploding. The only defense against these performance stresses is to upgrade the CPU/motherboard to the latest Intel technology. But even with upgrades, performance slips over time as a function of load and new feature complexity.

Furthermore, being no more than PCs, IP appliances reduce infrastructure resilience and scalability, as they must be located directly in the data path. Building full-meshed topologies in the Web data center is virtually impossible as IP appliances can only support single input and output ports. While some devices have added bridging or routing capabilities to support multiple interfaces, this is merely a bandage to the problem—performance degrades faster with the added networking overhead. Host-based routers were obsolete years ago for the same reasons!

#### **WEB SWITCHING: THE NEXT STEP IN NETWORKING FOR THE WEB**

The ability to scale the entire Web data center infrastructure smoothly and easily is crucial to keeping in-step with Web growth. A revolutionary solution is now available in the form of a Web data center switch (Web switch). In the most basic form, the Web switch is a super-fast LAN switch that integrates the Web traffic management and control functions currently running on separate IP appliances. These include:

- *Local server load balancing*
- *Global server load balancing*
- *Server security protection*
- *Bandwidth management*
- *Traffic steering/redirection to remove inline server-based devices from the traffic path*
- *Ability to offload functions from expensive resources such as WAN routers and firewalls*

The LAN switch represents the ideal platform for such integration. Already providing the common connectivity fabric for all devices in a Web data center while front-ending all servers and their applications, this device is clearly in the best spot for administrators to exert traffic classification and control functions (see Figure 3).

#### **DISSECTING THE WEB SWITCH**

Terms that have been used to describe switches that provide the capabilities described above include session switches, content-intelligent switches, Layer-4 switches, even Layer-7 switches. None of these terms accurately describe the multi-faceted intelligence that a Web switch must possess. This intelligence includes the following:

##### **1. Server-Awareness**

The Web switch intercepts user requests destined for virtual servers (server clusters) and distributes them to the real servers associated with the requested virtual servers. While doing this, the switch must pay attention to optimizing response time for each user request—ensuring best performance and total service availability. As a traffic hub, the Web switch is also in the best spot to collect statistics on the virtual and real servers for use in planning and reporting. Specifically, the Web switch executes the following server-related functions:

- **Optimum Load Distribution Across Server Farms.** The Web switch keeps track of all potential resources at the Web farm that can service a particular application request. It possesses dynamic knowledge of each server's health, performance, and available capacity in order to select the "best available" server for each client request. Lastly, the Web switch performs the proper network address translation and redirects traffic addressed to the virtual servers to real servers selected and vice versa. In this way, the Web switch makes server load balancing completely transparent to the users and the real servers.
- **Web Farm Availability Management.** The Web switch must bypass failed or overloaded servers when load-balancing user requests. This involves monitoring not only the Web servers themselves, but also the entire path from the Web servers to the content. In fact, some Web administrators may have custom server monitoring requirements that include checking each server's disk I/O, memory capacity, CPU utilization, etc. The Web switch needs to provide a mechanism to accept inputs from custom-built, host-based monitoring agents or external scripts and take proper actions. The Web switch must also log failure statistics and other information which is useful for longer term reliability planning. It's important to note that the Web switch must never endanger service availability itself.
- **Web Farm Performance Management.** Another key responsibility of the Web switch is to maximize performance. By spreading user requests among the available servers and balancing the load, uniformly high performance is achieved. Further, the Web switch itself should impose minimal latency, particularly in connecting user requests to servers. This is important since response time is a key attribute that influences user experience on a website. The Web switch is well positioned to provide feedback on site quality by logging performance statistics such as application response time, hits per second, individual server response times and connection statistics for accounting, service level agreement management and capacity planning purposes.

## 2. Application-Awareness

The Web switch must manage state information of each application session in order to deliver packets for each session to the correct server and to ensure that open sessions are always completed. Knowledge of each application being load balanced also allows the Web switch to provide more in-depth server health checks, increasing application availability.

- **Application-Intelligent Health Checking.** To maximize service availability, server monitoring should include facilities to test proper operation of the applications being load balanced. Web switches support this by setting up an application session, retrieving a piece of content for each application being load balanced, and verifying successful completion.
- **Web Session And Application State Management.** A session is defined as the entire interaction between a user and a server to complete an application transaction. All packets within a session should be forwarded to the same server to ensure data integrity. The difficulty is that what constitutes a session is application dependent. For example, an HTTP session maps to a TCP connection while an FTP session maps to two TCP connections, one for control information and one for data. The following are a few examples of different session state management tasks required on Web switches:
  - TCP session management. At minimum, when a TCP connection is established between a client and a particular Web server, all ensuing traffic associated with that TCP session must be forwarded to the same server. If not, the client will RESET the connection and start all over.

- UDP session management. Some UDP applications, such as NFS, transmit large UDP datagrams which are fragmented by IP into smaller packets for transmission. In load balancing UDP applications, the Web switch must take care to forward all IP fragments that constitute a UDP datagram to the same server. In addition, while the UDP protocol is connectionless, a "UDP session" can still be defined as the duration over which the same user exchanges UDP packets continually with the same application. By forwarding all packets within the "UDP session" to the same server, the Web switch increases server efficiency by leveraging data pre-fetching and memory caching on servers.
- Persistent sessions. Applications such as Internet searches, forms, etc., use persistent HTTP sessions which require consecutive TCP connections from a user to be directed to the same server. The Web switch must recognize persistent applications and handle server selection in a manner consistent with the intent of the application.
- SSL sessions for eCommerce. eCommerce sessions are often encrypted using the SSL protocol. When these sessions are sent through a proxy firewall, the only way to distinguish real sessions from different clients is to examine the "SSL Session ID", which is generated when a client and a specific SSL server execute handshaking to begin an encrypted SSL session. To support granular load balancing and persistency on a client-by-client basis, the Web switch must decode SSL information and load balance by SSL Session ID instead of the source IP address. This requires a significant amount of processing on the Web switch.
- "Shopping Cart" session completion. Each eShopping session from a user may consist of two connections, a persistent HTTP connection for shopping-cart tracking and a SSL connection that is opened when the user selects items from the shopping cart for purchase. The Web switch must be able to associate both sessions with the same server.

### 3. Content-Awareness

By parsing session content at high speeds, a Web switch can support a wide range of applications and provide more efficient and granular load distribution. Information obtained from session content is useful in the following scenarios:

- **Supporting Applications With Dynamic Ports.** Many Web-centric applications including ICQ and VoIP communicate dynamic session information across TCP connections established to the applications' well known ports. This dynamic session information instructs the server and/or client to create one or more separate TCP connections that will be used during most of the data transfer or server interaction. To provide traffic management services for these applications, the Web switch needs to parse traffic communicated to the applications' well known ports to decode the necessary information for managing the subsequent TCP connections set up between the client and server.
- **Web Traffic Control Based On URL.** A Web switch can simplify IP address management by allowing a Web administrator to advertise only the public IP address for the top-level domain. When the Web switch intercepts a user request addressed to that IP address, it can then parse the subsequent packets to determine the requested URL which points to the desired destination server or virtual server. With the ability to extract URLs from user sessions, the Web switch lets users store content on different servers—each optimized by content type—and rely on the switch to direct traffic to the right server based on the content

type indicated by the requested URL. Awareness of content types also allows the Web switch to distinguish between static and dynamic object requests and intelligently route static object requests to cache servers and dynamic content requests to the designated destination server.

- **Cookie Cutting.** The Web switch can parse user sessions for cookies that provide specific information on the user. The information can then be used to drive policy selection for the user session, support accurate persistent server bindings based on user cookies, etc. Content parsing is not a simple pattern-matching task. In order to get to the TCP packets that carry content information, the Web switch must complete the TCP handshake (also called TCP connection termination) with a requesting user. After it obtains the desired content from subsequent packets and identifies the appropriate server to which to pass on the user request, the Web switch opens a TCP connection to the target server and "splices" the user-to-switch and switch-to-server connections. That involves fixing up the mis-matched TCP parameters including window size, sequence number and TCP checksum on every packet that flows between the user and the server, until the session terminates.

#### 4. Network-Awareness

To avoid a proliferation of boxes in the Web data center and some severe topology restrictions while delivering maximum scalability, Web switches must support a full-range of network services including layer 2 and 3 functions (bridging and routing) at high speeds. In addition, detailed topology knowledge is required to distribute traffic among servers located in different geographical sites by forwarding user sessions to the "closest" site as well as to route sessions around network failures. Specifically, the minimum networking requirement for a Web switch includes:

- Full wire speed layer 2 and layer 3 switching on every port (150K pps on Fast Ethernet and 1.5M pps on Gigabit Ethernet)
- Standards-based layer 2 functions: spanning tree, 802.1Q VLANs, 802.1P multicast groups and prioritized services
- Standards-based layer 3 functions: ARP, ICMP, RIP v2, OSPF, VRRP, IP Multicast.
- Support for network-oriented QoS functions: 802.1P, IP TOS, DiffServ
- Support for mesh and redundant network configurations to eliminate all single points of failure.

#### 5. Advance Web Switch Functions

Besides server load balancing, the Web Switch must support the following capabilities to effectively manage all aspects of the Web data center infrastructure.

- **Multi-Site Web Farm Coordination.** Websites are often "virtualized" by pooling resources at different geographic locations; all able to provide the same service. All the different sites require access to the same content and—if a dynamic application is supported—the same data. The Web switch controls the flow of user requests to the different load balancing sites, taking into account site health and load, "nearness" to the client, and each site's Web farm performance. Integration of both local and distributed load balancing on the Web switch minimizes administration. The combined functions also provide a higher level of availability and more accurate measurement of site performance.

- **Transparent Traffic Redirection.** This function provides flexibility for site administrators to implement unique traffic management schemes that have not been integrated into the Web switch. Traffic redirection allows user traffic to be steered transparently to a proxy for the intended destination server. This is useful for caching or "pre-processing" of traffic. For example, email traffic can be steered to a spam filtering server before being forwarded to the designated mail server.
- **Non-Server Load Balancing.** Besides the Web switch, traffic flowing through the Web data center may traverse many traffic management devices such as routers, firewalls, VPN servers, etc. Each of the devices, being on the data path, is a potential bottleneck and single point of failure. Web switches can bring the same scalability and high availability benefits of load balancing to these devices as they bring to Web servers. For example, users can have the Web switch distribute traffic sessions across multiple active firewalls to increase site throughput. Likewise, instead of forwarding outbound traffic through a single WAN router acting as the universal default gateway, the Web switch can transparently load balance outbound traffic across multiple WAN routers (e.g., hot-standby routers), increasing performance and productivity of expensive router resources.
- **Web Farm Bandwidth Management.** Performance management also includes policing a set of usage policies for different websites supported on a Web server farm. For example, the amount of traffic that is allowed to a specific application or website can be measured and if necessary, limited. This requirement is seen routinely at Web hosting environments to ensure that one site does not hog all resources and degrade other sites' performance. Bandwidth management is applicable in corporate Web data centers as well. If the traffic from an application starts to exceed its pre-set threshold, a variety of traffic management options can be initiated. For example, the Web switch can hard-limit the traffic allowed to the application, slow down communications for all users so that everyone's performance smoothly degrades but no packets are dropped, or it can temporarily allocate more bandwidth for the busy application. The Web switch must provide "knobs" to allow bandwidth management options to be tuned and refined to optimize site operation, over a wide variety of load and application deployment conditions.
- **Security Measures.** Packet filtering, stateful inspection and firewalling, VPN tunnel termination, and SYN attack protection are all functions that could potentially be provided by the Web switch.

## CONSIDERING WEB SWITCH DESIGN

The combination of server, application, content and network intelligence is not found in conventional layer 2/3 switches and routers which forward each packet as an individual unit - making no distinction between packets associated with different TCP or higher level sessions. Layer 2/3 devices also have no awareness of the health or load of the ultimate end-system to which the packets are destined, nor do they have the processing capacity to handle packet content parsing at high speed.

Web switches are a new breed of product, uniquely designed to meet the needs of Web data centers. In addition to the wirespeed packet forwarding rates, the modularity, port density and redundancy that is expected of state-of-the-art, Layer 2/3 backbone switches, a Web switch must be designed from the silicon up for blazingly fast session switching. At the same time, the Web switch also needs a high degree of software flexibility to perform a variety of Layers 4-7 services.

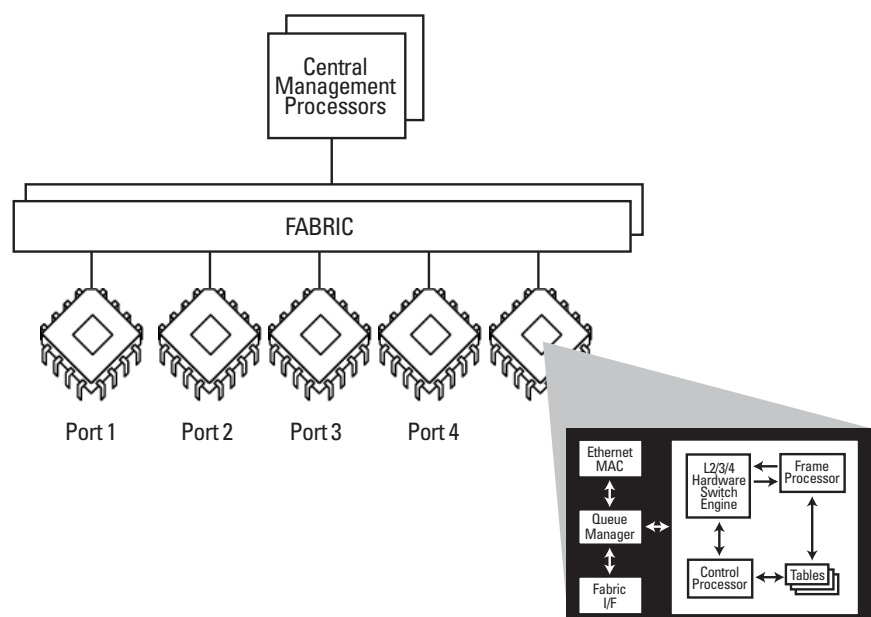


## 1. Designing For Performance And Flexibility

The following architecture blocks are key to Web switch design:

- **State-of-the art switching ASICs.** These ASICs integrate repetitive networking tasks in hardware, resulting in wire-speed throughput for Layer 2/3 packet switching and wire-speed session switching at higher layers. These tasks include:
  - Layers 2 and 3 filtering and forwarding
  - Layer 4 address filtering and session mapping
  - Session forwarding
  - TCP, UDP and IP session state tracking, session table management, look-ups, network address translation and session termination processing
  - Content-parsing assist (e.g., string matching, regular expression matching)
  - TCP connection splicing with automatic TCP header modification (required for content parsing)
  - Queuing and resource arbitration enabling bandwidth management services at full wire speed
  - Statistics, RMON and other per-port management tasks
  - Application flow accounting
- **Distributed processing on every data path.** Processors are needed for functions that require software flexibility. By distributing processors on every port, software processing can take place in parallel on both traffic ingress and egress ports, thereby maximizing performance (see Figure 4).

FIGURE FOUR: ALTEON 700 WEB SWITCH ARCHITECTURE



Examples of tasks that require processing include:

- *Content snooping for applications that use dynamic ports and URL or cookie inspection and server selection*
- *Session-to-server binding*
- *Exceptional session state tracking (e.g. SSL session)*
- *Bandwidth allocation, metering and control on a per-application flow basis*
- *TCP flow control*
- *Flexible policy enforcement for security, QoS, and preferential services features*
- *Session failover*

• **Central CPU for non real-time control functions.** This includes:

- *Layer 2 and 3 control functions (e.g. routing protocols, spanning tree)*
- *Exception packet processing*
- *Server and application monitoring*
- *Site-to-site communications*
- *Network management*

## 2. Is All This Performance Necessary?

Though it may seem that such massive throughput is an overkill for a Web data center attached to the Internet by only a few DS-3 links or less, several factors drive the need for such performance now and in the future.

- In a multi-tiered server architecture, each packet may make several hops over the Web Farm network: from router to firewall, firewall to Web server, Web server to application server, application server to data server, and all again in reverse. There can also be a large number of procedure calls between peer servers. The bottom line is—significantly more LAN bandwidth is needed per unit of WAN bandwidth.
- Web switches must also support non-Web-related server applications, including back-ups, database synchronization, software distribution, local management traffic, and other types of traffic which may never go over the WAN link.
- Web switching performance should be evaluated using different metrics compared to Layer 2/3 switches. The key performance measurement for a Web switch is Session Setup and Teardown Rate, which measures how fast a Web switch can process TCP connections between users and servers, and session-to-server binding latency.
- Analogous to the standard packet-per-second throughput measure using minimum sized frames, the standard session setup/tear-down rate should be measured using minimum sized sessions. A minimum sized session is comprised of four 64-byte packets from client to server and three 64-byte packets from server to client. On this basis, wire rates for session setup and teardown are approximately 16,700 sessions per second on a DS-3 (45 Mbps) link, and 37,200 sessions per second on a Fast Ethernet connection, and 372,000 sessions per second on a Gigabit Ethernet link.
- Consider a Web data center that is attached to dual DS-3 WAN links. The Web switch needs to have the session processing capacity to keep up with 32,000 new sessions per second. Central CPU-based Web traffic management devices will have a hard time keeping up with this many TCP connections per sec-

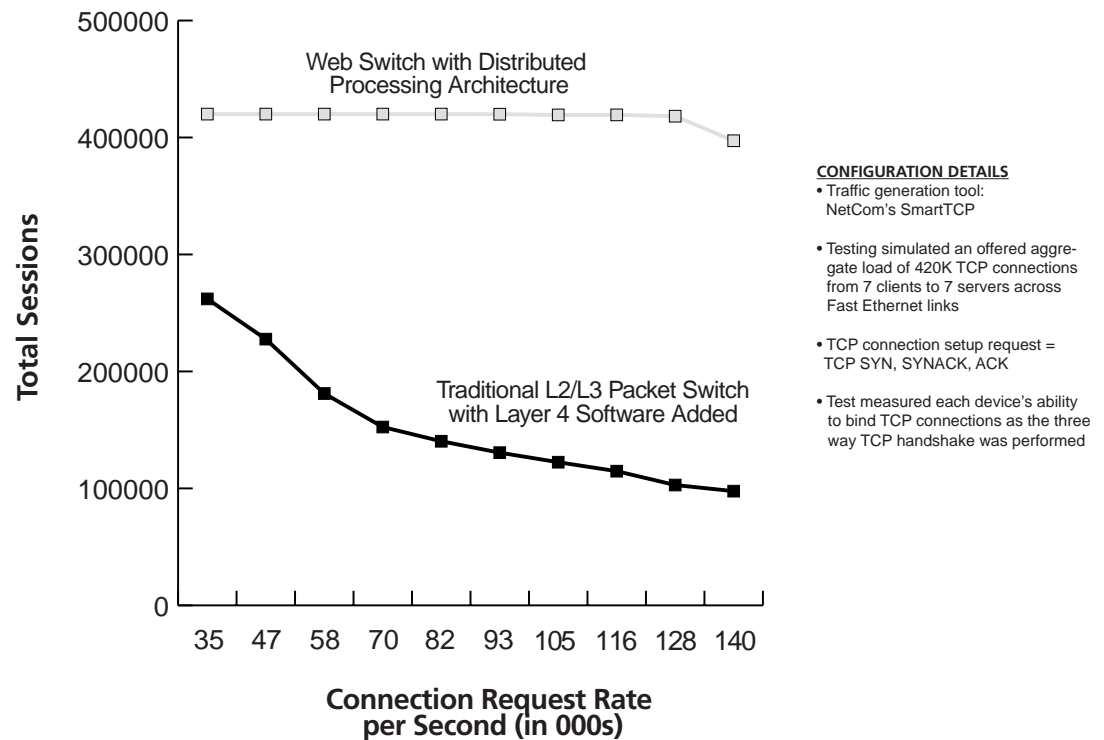
ond, not to mention other tasks such as network management, server health monitoring, etc., that the CPU must process in parallel.

- To deliver packets from point A to B, layer 2/3 switches do as little as possible as fast as possible. On the other hand, Web traffic management systems do as much analysis and processing as required to assure that server resources and/or available WAN bandwidth are used optimally. Web switches must do both.
- WAN bandwidth is rapidly becoming more plentiful and less expensive. The Web Farm infrastructure should have some headroom to support higher and higher levels of WAN traffic and at the same time have the capacity to support new features and functions.

### 3. Don't Be Fooled

Some vendors with ASIC-assisted Layer2/3 switches have added server load balancing capabilities to their products but have done so by loading a software-based implementation on to their central (management) CPU. This is nothing more than an IP appliance bolted onto a switch. When traffic is heavy, this approach not only reduces server load balancing performance but also concurrent Layer 2/3 services and management performance.

FIGURE FIVE: Centralized vs. Distributed Processing Architecture



**SUMMARY**

Service providers, content providers and corporations are quickly finding themselves on the path to supporting large-scale Web-oriented applications. Meanwhile computing technology, needed to create faster servers, is not advancing fast enough to keep up with the growing demands on websites. To support unpredictable loads and optimize Web applications, server functions are being partitioned to create multi-tiered application architectures with server clustering implemented where possible to provide sufficient server bandwidth.

This results in large, complex Web server farms that have unique and potentially conflicting requirements: massive and rapid scalability, combined with the flexibility for quick change, combined with the requirement for granular tuning of service delivery parameters combined with the need for control.

These pressures have led to the requirement for a new type of value-added infrastructure device - these devices, dubbed Web switches or Layer 4 switches, inherit the requirements of high speed Layer 2/3 switches, but also must rapidly perform networking services vastly different from those that Layer 2/3 switches perform. These services include server and site resource monitoring, server load balancing to maintain application availability and scalability, multi-site synchronization and granular bandwidth management.

Web or Layer 4 switches must achieve the performance of Layer 2/3 switches with the flexibility and expeditious feature additions of software-based Internet traffic appliances.

Alteon WebSystems' Web switch architecture is a unique combination of ASIC support for repetitive Layer 2/3/4 tasks and per-port embedded packet processing for complex Layer 4 – 7 operations at high speed. The unique Alteon Web switches are intersecting the need of those companies supporting large scale Web server farms over ultra-high speed data center LANs. They are fundamental to scaling Web data centers to support the hyper-growth of the Internet.