Web Server Clustering with Single-IP Image: Design and Implementation

Yi-Min Wang
AT&T Labs, Research
Florham Park, New Jersey
Om P. Damani
Dept. of Computer Science
Univ. of Texas at Austin
P. Emerald Chung **Yennun Huang** **Chandra Kintala**
Bell Laboratories, Lucent Technologies
Murray Hill, New Jersey

November 1, 1997

---

1. Introduction

Web server clustering is an attractive approach to achieving scalability, load balancing, and high availability for Web services. ***Request dispatching mechanism*** is at the heart of any server clustering techniques. Figure 1 gives an overview of various request dispatching mechanisms that can be implemented at different layers. First, a Web client specifies a *service name*. The name can be translated by a Java applet into one or more URLs, each containing a particular host name. The applet can be either a HAWA (High-Availability Web Access) applet that provides user-customizable fault-tolerant parallel accesses [Wang 97], or a smart client applet [Yoshikawam 97] that acts as an client-side agent for a particular Web site. Each host name is then translated by the Domain Name Service (DNS) into an IP address. DNS round-robin [Kwan 95] is a popular technique for supporting server clustering in this translation process. Once an IP address is selected, the client's request can reach one of the server-side machines, either a router or directly a server machine. In the former case, the IP address is further mapped to a final IP address by either the Network Address Translation approach [Cisco WWW], the TCP router approach [Attanasio 92] [IBM WWW], or the ONE-IP approach [Damani 97], of which the technical details will be described in this paper. Finally, when the request reaches its final destination, the HTTP redirection [Anderson 96] mechanism allows the server to reject the request and ask the client to resend it to another host.

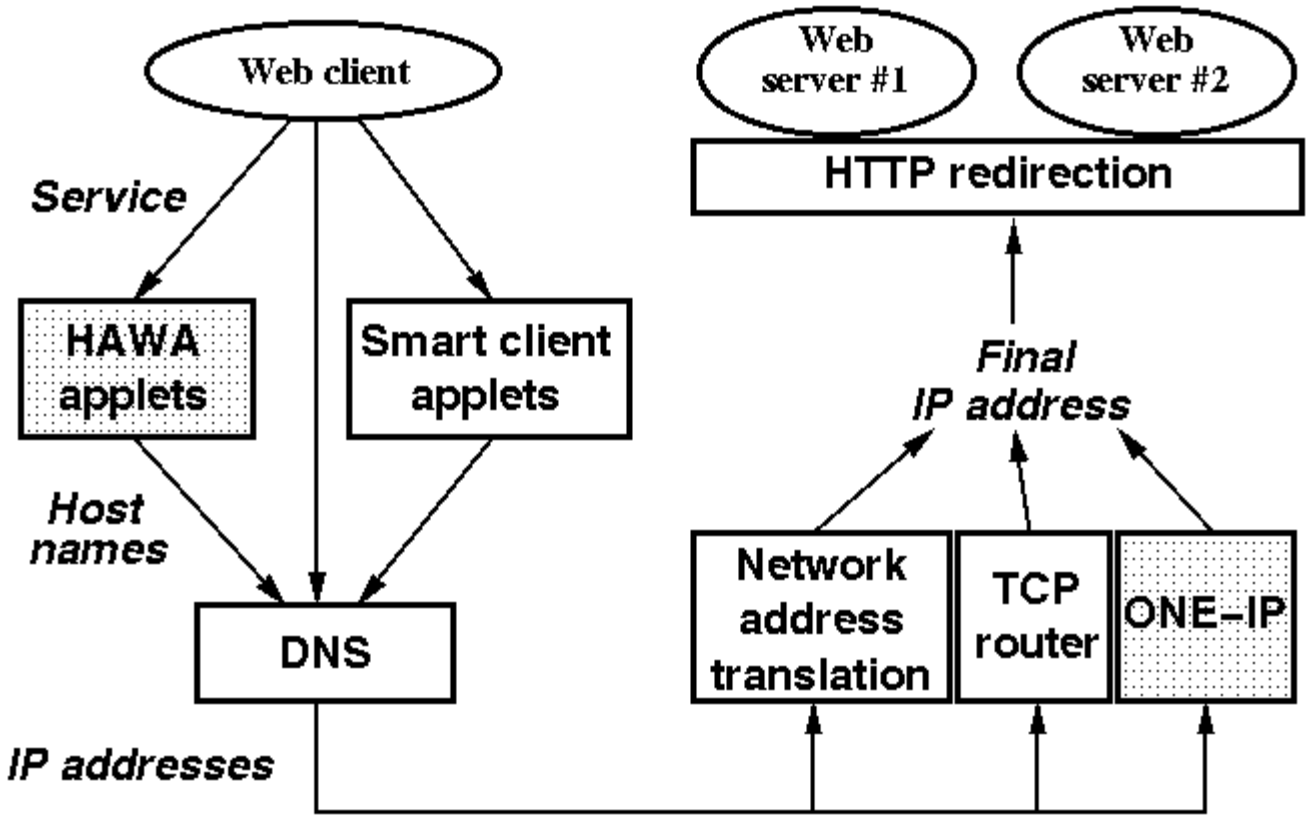

Figure 1: Overview of request dispatching mechanisms

---

2. The ONE-IP Project

The goal of the ONE-IP project is to investigate the issues involved when the networking protocol stacks are modified to support a single-IP image for a cluster of machines. Two approaches have been studied and implemented in the NetBSD kernel. The ***routing-based dispatching*** approach modifies the IP forwarding layer at a centralized dispatcher to route each request based on a hash function of the client machine's IP address. The ***broadcast-based dispatching*** approach uses the same hashing technique but applies it at a local filter at the device driver layer of each server machine. In both approaches, a single cluster IP address is publicized and all clients' requests are addressed to that IP address. In this paper, we use the name *ghostIP* for the cluster IP address to stress the fact that none of the server machines owns that IP as its primary address. To eliminate the need to modify any source or destination address in the request packet, however, all server machines share *ghostIP* as their secondary address through the `ifconfig alias` command.

2.1. Routing-based dispatching

Figure 2 gives an overview of routing-based dispatching. Every client request packet with *ghostIP* as its destination address is routed to a normal server-side router, which in turn routes the packet to a designated *dispatcher*. The dispatcher runs our modified kernel and is configured to run in the routing mode ([Wright 95] p.157). It selects one of the servers based on a hashed value of the source IP address, and performs the final

routing to the chosen one, Server #2 in this example. Server #2 then receives the packet, processes the request, and sends the reply back to the client. Since the dispatching is only based on IP hashing and the boundary of TCP messages is not maintained, ONE-IP can only provide static load balancing. The advantages are fast and stateless dispatching, which facilitates the failover of the dispatcher itself. Also, since the size of responses is usually much larger than that of requests in the Web setting, it is a very desirable feature to allow the responses to be sent directly back to the client, without passing through the dispatcher.

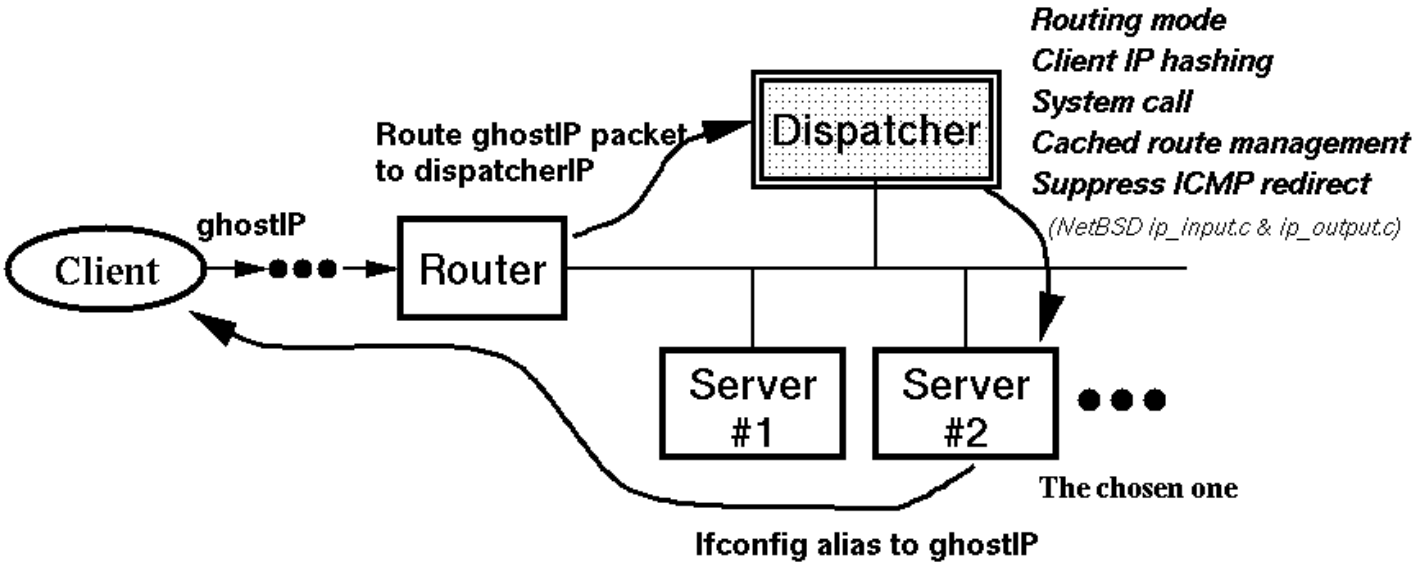

Figure 2: Routing-based dispatching

Technical details

- **Router**: ONE-IP assumes that the cluster owner does not have control over the kernel code running on the router (otherwise, the router can serve as the dispatcher). To allow the dispatcher to have access to every packet addressed to *ghostIP*, a routing entry with `Destination = ghostIP` and `Gateway = dispatcher IP` is added to the router's routing table ([Stevens 94] p.116).

- **Servers**: all server machines are `ifconfig alias` to *ghostIP* so that they can receive packets with *ghostIP* being the destination address. This does not cause a confusion because ONE-IP makes sure that neither the router nor the dispatcher uses *ghostIP* to find the link-layer address of the next hop.

- **Dispatcher**: this is the only machine that runs the modified kernel in the routing-based dispatching approach. The required kernel modifications include
  - Client IP hashing: in the original `ip_forward()` routine, the destination address is always used to find the next hop ([Wright 95] p.222). In the current implementation of ONE-IP, a simple `mod` operation on the source IP address is used to select a server, and the server's IP address is used to find the next hop. We also implemented a system call for changing the hashing function.
  - Cached route management: in both the `ip_forward()` and `ip_output()` routines, a *one-behind cache* is maintained to minimize the number of routing lookups ([Wright 95] p.223). A discrepancy between the current and the cached destination addresses always trigger a new lookup with the former. Since ONE-IP explicitly does not want to use the destination address *ghostIP* for the lookup, it needs to protect the obtained route throughout the routines.

- ICMP-redirect suppression: since the dispatcher receives and forwards the packet on the same network interface, an ICMP redirect message may be sent to the sending host in an attempt to bypass the dispatcher for subsequent requests ([Wright 95] pp.223-225). Since that is undesirable in ONE-IP, the ICMP-redirect messages need to be disabled for *ghostIP* packets.

2.2. Broadcast-based dispatching

Figure 3 gives an overview of broadcast-based dispatching. Every client request packet with *ghostIP* as its destination address is broadcast to all servers. Every server machine receives the packet and decides whether it should process or discard it based on a hashed value of the source IP address. Only one server will eventually process the request and send a response back to the client.

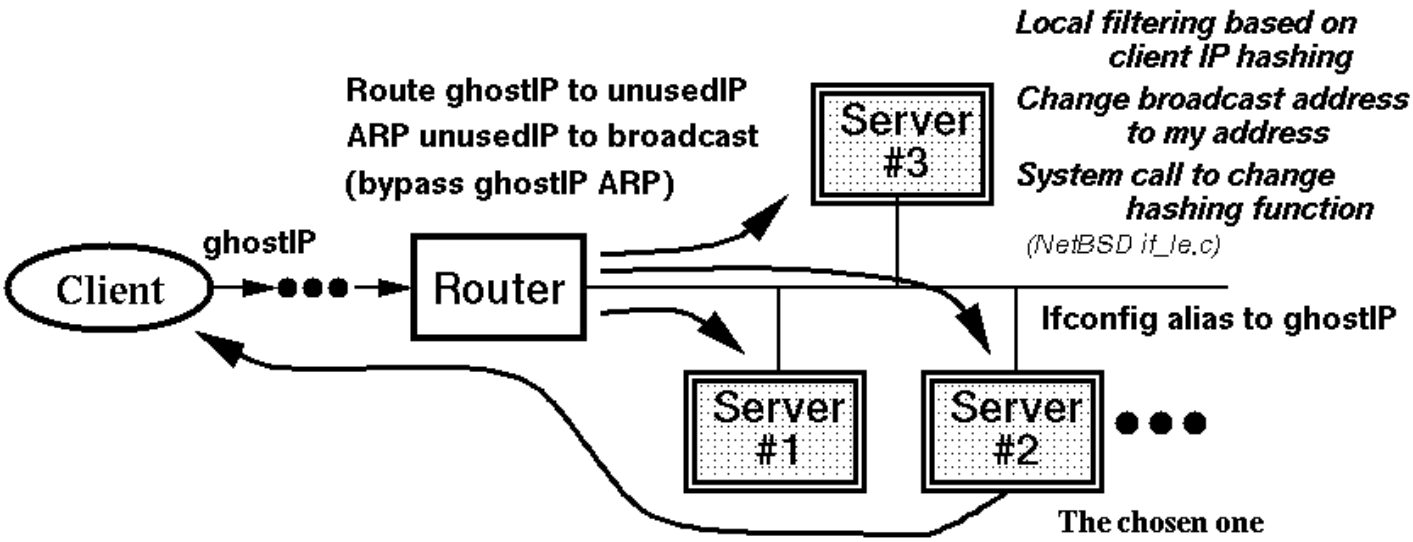

Figure 3: Broadcast-based dispatching

Technical details

- **Router**: directly mapping *ghostIP* to Ethernet broadcast address (all one bits) may not work because any ARP request from a chosen server may overwrite the mapping to associate *ghostIP* with that server's Ethernet address ([Stevens 94] p.63). Our solution is to set up a routing entry at the router that asks all packets destined for *ghostIP* to be routed to *unusedIP*, which is a legal subnet address in the cluster LAN and is not used by any machine. In addition, a permanent ARP entry is inserted ([Stevens 94] p.63) to associate *unusedIP* with the Ethernet broadcast address so that when the router routes a packet to *unusedIP*, it actually broadcasts the packet.

- **Servers**: all server machines are `ifconfig alias` to *ghostIP* so that they can receive packets with destination address *ghostIP*.
  - All server machines run the modified kernel that makes the decision to accept or discard *ghostIP* packets based on a simple `mod` operation in the `leread()` routine ([Wright 95] pp.103-105).

- A potential problem is that some operating systems, including NetBSD, do not allow a TCP packet to be processed if it is received from an Ethernet broadcast address ([Wright 95] p.943). So ONE-IP changes the Ethernet address in the Ethernet header from the broadcast address to the local host's address once the server decides to accept the packet.

---

3. Summary

We have described the design and the implementation of two techniques for supporting server clustering with a single-IP image. The routing-based dispatching technique modifies the IP forwarding routines. In order to keep the packet's destination address unchanged and use the primary IP addresses of server machines to perform final routing, issues such as cached route management and ICMP-redirect messages need to be addressed. The broadcast-based dispatching technique modifies the input routines at the device driver. Since packets are dispatched with unicast IP address but broadcast Ethernet address, care must be taken to ensure that the higher-layer protocols at a selected server machine will accept the packets.

---

4. References

[Anderson 96]
D. Anderson, T. Yang, V. Holmedahl and O. H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers", *IPPS'96*, April, 1996.

[Attanasio 92]
C. R. Attanasio, and S. E. Smith, "A Virtual Multiprocessor Implemented by an Encapsulated Cluster of Loosely Coupled Computers", *IBM Research Report RC18442*, 1992

[Cisco WWW]
Cisco Local Director.

[Damani 97]
O. P. Damani, P. Y. Chung, Y. Huang, C. Kintala, and Y. M. Wang, " ONE-IP: Techniques for hosting a service on a cluster of machines," in *Proc. the Sixth Int. World Wide Web Conference*, April 1997.

[IBM WWW]
IBM Interactive Network Dispatcher

[Kwan 95]
T. T. Kwan, R. E. McGrath, and D. A. Reed, "NCSA's World Wide Web Server: Design and Performance", *IEEE Computer*, pp. 68-74, Nov. 1995.

[Stevens 94]
W. R. Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley, 1994.

[Wang 97]
Y. M. Wang, P. Y. Chung, C. M. Lin, and Y. Huang, " HAWA: A client-side approach to high-availability web access," presented at *the Sixth Int. World Wide Web Conference*, April 1997.

[Wright 95]
G. R. Wright and W. R. Stevens, TCP/IP Illustrated, Volume 2, Addison-Wesley, 1995.

[Yoshikawam 97]
C. Yoshikawam, B. Chun, P. Eastham, A. Vahdat, T. Anderson, and D. Culler, "Using Smart Clients to Build Scalable Services", USENIX'97, Jan. 1997.