INTERNET-DRAFT
Expires: February 21, 1997

Chi Chu
Research 2000, Inc.
August 1996

IP Cluster
draft-chu-ip-cluster-00.txt

Status of this Memo

1. Abstract

This Internet-Draft is intended to provide a means for "IP Clustering" across multiple servers. It is meant as an improved alternative to the various solutions for distributing WWW traffic already attempted by the IETF DNS Working Group. In addition, the clustering method can be applied not only to a heavily visited web server, but also to any overloaded TCP/IP servers such as a domain name server. The IP Cluster provides two primary functions: IP traffic distribution to multiple servers and fault-tolerance.

2. Introduction

The notion of distributing IP (Web) data traffic to multiple server machines has already been foray-ed by the various DNS methods mentioned or described in RFC 1794 [1]. The basic drawbacks for all these methods are similar:

* short or zero TTL for DNS records - this is not intended by the DNS specification and incurs a few unpleasant consequences;
* heavy DNS traffic - since secondary or non-authoritative DNS servers cannot effectively cache the data, all these methods generate heavy DNS queries across the global Internet, bombarding a chain of servers in the name space;

* potentially high delay - if any server in the DNS chain experiences outage or bottleneck, the response to the initial query would be significantly delayed if an alternate DNS server were required to process the query.
* the primary DNS server becomes the single point of failure - since the TTL is very small or zero, outage of the primary server for even a small period of time results in failed DNS lookup;
* easier to spoof - a DNS record can be easily "spoof-ed" to mislead a client to a bogus host name to IP address mapping;

and among other drawbacks that are method specific. In short, these DNS methods solve one problem that may be beneficial to a single site, but create another that can be quite undesirable for the Internet at large. Just imagine what would happen if every website decides to implement a DNS method for distributing its web traffic.

Clearly, it is imperative and highly desirable that an alternative solution be established that does not suffer the same drawbacks discussed above, and yet the new solution not introduce a new problem equal in its severity to the network at large.

3. The Alternative

3.1 Applicable Topology

The proposed method requires an IP router connecting to multiple servers in a switch-like configuration. That is, each server machine is directly connected to a unique physical port/interface on the router. Since a router interface can be a LAN or serial interface, this IP cluster formation can span locally or be distributed via wide-area network.

3.2 Description

The nature of IP load balancing requires that for a given IP host name, typically corresponding to some network services, the data traffic and thus the processing be actually distributed among several server machines, with some control. This idea of a "virtual server" provides transparent services to a remote client. The virtual server itself consists a number of host machines, or cluster members, each performing a set of services. In a true cluster environment, a cluster member performs a set of services or functions that may be different from that of another member within the cluster.

Similar to all the DNS load balancing methods, the proposed method described in this document assumes symmetric host processing. Namely, the so-called "IP Cluster" consists of a number of cluster

members each of which performs the same set of services (although strictly speaking, it does not have to be necessarily so).

The alternative method does not rely on the dynamic host name to IP address mapping. Instead, it relies on the concept of a Virtual IP (VIP) address. This VIP address is configured as the host IP address for all cluster members. Each cluster member is directly connected to a unique router interface port, much like an Ether-Switch configuration, topologically.

The VIP address appears to the outside world as just another unique IP address, with the usual DNS host name to IP address mapping in the traditional static sense. To each of the IP cluster members, it thinks that this VIP address is its globally unique host address.

However, this VIP address appears very differently to the IP router to which all the cluster members are directly connected to. With careful and deliberate choice of the VIP address (e.g., xx.xx.xx.63 for a Class C network), and with the appropriate subnet (or variable subnet) mask enabled in the router interface ports, this unique host IP address is in effect a broadcasting address as for as the router is concerned. Consequently, upon receiving an IP packet with destination address equal to this VIP address, the router will attempt to, assuming configured properly, broadcast the packet to all its relevant interfaces.

Each of the "broadcasted" interface, however, is configured with a simple filter. This simple filter basically filters on the IP source address of the incoming packet. Thus with each interface filter permitting only a unique and non-overlapping portion of the IP address space to route through, we have effectively achieved high-performance "IP-Switching".

Furthermore, since this portioning of the IP address space can be well controlled by each interface filter's bitmasking and wildcarding, load balancing can be accomplished now with respect to CPU, memory, IO, or all of the above, depending upon the application nature of the IP cluster.

4. A Scalable Model

The IP clustering model described above should scale very well. Physically, the "Virtual IP" clustering is primarily limited by the number of router interface ports. In terms of performance, the scalability of this model is limited mostly by network bandwidth technology and the router performance which is usually orders of magnitude greater than a workstation server's ability to deliver the same data throughput.

In short, this IP clustering model should scale quite linearly.

5. Fault Resilience and Fault Tolerance

Fault Resilience (FR) here means the ability of the IP cluster to

be able to

* automatically redistribute its parallel server processing in the event of any single cluster member (i.e., server machine) failure, and
* automatically restore to the normal parallel processing once the failed server has recovered (by whatever means).

The IP clustering method described in this proposal should be able to support the above requirements. There are a number of viable implementations, however; and I shall briefly describe the basic concept.

Essentially what needs to be done here to achieve FR is similar to what is done in a "classic" cluster environment. Each cluster member monitors the health and status of the other cluster members. When a failure is detected, each monitoring member (which means all but the failed machine) automatically enables itself to support a portion of the services or functions that it is configured to assume for that failed cluster member. When the failed cluster member becomes alive again (usually through a heartbeat), all other cluster members will fall back to their normal mode of processing.

While I will not delve into all the relevant issues of building a Fault Tolerant (FT) IP Cluster, suffice to say, however, with this IP cluster model, one may easily build a Fault Tolerant IP Cluster against any single point of host-or-network failure within the cluster.

6. Implementation

The implementation of this IP cluster assumes that a router used for IP switching is capable of forwarding IP broadcast packets. While most routers have limited broadcast forwarding capability (e.g., some may not forward TCP/IP broadcast packets), this limitation should be easily removed by a perspective router vendor by relaxing the artifically imposed transport-layer filtering (which is not entirely a router's business to begin with).

Reconfiguration of the IP-Switch/router filters for achieving better load balance should be performed by an automated script. Since this type of reconfiguration is considered system down-time for the IP cluster, the implementation of such a script should minimize the down-time by, for instance, separating logging into the router from actually modifying the filters with human control.

As for communications between cluster members (i.e., heartbeat, etc.), any number of protocols can be used. It may be as simple as ping and tcp-echo, or as sophisticated as a new multicast

protocol.

7. Performance

As already mentioned in Section 4, the IP clustering method described in this proposal should be extremely fast. The so-called IP cluster here is essentially an IP-Switch (as opposed to an Ether or FastEther Switch) connecting to number of cluster members each taking full advantage of the underlying transmission medium without the usual network contention.

Assuming that one is to configure a "Super IP-Switch" with maximum IO ports and each port is connected to the highest bandwidth technology and server machine available, the only issue with regard to performance then is the router's routing capability, particularly the router's CPU required to perform the interface filtering.

We can rest assured, however, that this interface filtering or the router's routing performance cannot realistically be an issue for two reasons. Reason one, because of bitmasking and wildcarding, each interface filter list should be very short and compact. (I don't see more than six lines in each access list unless the same router is also used for firewalling, etc.) Reason two, long before one reaches such routing performance issues, any reasonable organization would want to add a second router into the same IP cluster. The VIP clustering model supports multiple routers as an integral part of a single IP cluster. In fact, building such an IP cluster with multiple routers is one step towards building a fault-tolerant IP cluster.

One question remains: How effective is the load balancing scheme based on the IP source address filtering, which if not effective, would defeat a lot of this high-performance claim. I would say: pretty effective, especially if the client base is very large (which is what this proposal is intended to accomplish to begin with).

This is simply a basic principle of statistical analysis: when there is a large number of statistical samples, with each sample behaving randomly and wildly, the overall statistical distribution is often predictable and well behaved. In fact, the larger the number, the more predictable and better behaved the statistical envelope would be. Thus, this statistical property works greatly in favor of this Internet-Draft's intent to use the IP cluster to support very large client base.

Assuming one has setup the proposed IP cluster with multiple servers. It makes no sense to talk about how good the load balance actually is when the traffic is light enough that if all the traffic gets distributed to a single cluster member that that

member server is still not overloaded. Good load balance becomes relevant when traffic is heavy enough that some or all of the cluster members must share significant (but still not necessarily equal) portions of the traffic load. It is important to keep the

perspective that the real purpose of clustering is to avoid server overloading and not to artificially maintain equal load balance at all time. The beauty of this IP clustering model is that the more traffic and the larger the client base grows, the better and more evenly the cluster distributes the load without incurring any processing overhead.

The above load analysis simply means that an effective IP cluster does not require fully dynamic load balancing per IP packet. In fact, a truly dynamic load balancing scheme on per packet basis would adversely affect the performance of such an IP cluster. How often (e.g., once a month, etc.) and what criteria (e.g., CPU, memory, IO) the load balance sampling and analyzing should be performed in order to re-tune, if necessary, the IP-Switch/router access filter lists are application dependent.

8. Security Considerations

While the DNS methods for IP clustering relies on dynamic host name to IP address mapping, which can easily be "spoof-ed", the Virtual IP method does not suffer the same level of security issues for the simple reason that it is more difficult to spoof (and spoof it well) the routing topology of the Internet than to spoof a DNS record.

Additionally, this Virtual IP clustering model does not preclude any security schemes that are available under a non-cluster single server environment, firewalls included.

9. Acknowledgments

Much appreciation is due to Mike Lee and Josh Sierles for enlightening me with the DNS load balancing methods, and to Josh again for referring me to RFC 1794.

10. References

[1] Brisco, T., "DNS Support for Load Balancing", RFC 1794, Rutgers University, April 1995.

11. Author's Address

Chi Chu
Research 2000, Inc.
265 Cherry Street, 16G
New York, New York 10002
USA

Phone: 212-598-9455

Email: chi@soho.ios.com
URL: http://soho.ios.com/~chi

This document expires February 21, 1997.