;login:

THE MAGAZINE OF USENIX & SAGE

October 2001 • Volume 26 • Number 6

inside:

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

ISPadmin

by Robert Haskins

Robert Haskins is currently employed by WorldNET Internet Services, an ISP based in Norwood, MA. After many years of saying he wouldn't work for a telephone company, he is now affiliated with one.

rhaskins@usenix.org

DNS/IP Address Infrastructure

This installment of ISPadmin looks at ways ISPs design and implement their domain name system (DNS) infrastructure. For any service provider who has a range (or ranges) of IP addresses and/or domains allocated to it, DNS is at the core of the services offered. Just imagine the Internet today without DNS! IP address management and DNS are, by their very nature, intertwined.

Introduction

The domain name system's job is to map names to IP addresses and IP addresses to names. It works by delegating "zones" of data (namespace as well as IP space) out to the organizations who use it. The delegated nature of DNS makes management easy as the "owners" of the data are responsible for maintaining it. DNS is, by many accounts, the single most successful implementation of a distributed database.

The DNS protocol is defined by a number of RFCs; see the DNS Resources Directory for an excellent compilation of references (including RFCs) for DNS. The DNS-related RFCs (draft and standard) are far too numerous to list here.

For a small provider, a DNS design is likely to be relatively straightforward. The interesting DNS/IP address problem is for the larger provider, where more than two DNS servers are required. Also, a larger provider will likely have a much larger pool of IP addresses which require management.

The issue of DNS touches upon many areas, including:

- Billing
- NOC troubleshooting and maintenance
- IP address allocation
- Service delivery
- IP routing

While I will touch briefly on each of the above areas, I will focus on DNS deployment and architecture.

One might wonder what it takes to manage and support a typical DNS infrastructure. At Ziplink, about 500 domains were hosted and approximately 80,000 IP addresses (one per dial port) were managed by one staff member half-time. The server machines required for this infrastructure included three Sun Ultra 10 class machines including one shared master and two dedicated slaves/caches which handled both inbound and outbound requests. The shared machine (which had other services besides BIND running on it) handled all of the data for the DNS records Ziplink was authoritative for, feeding the two dedicated slaves/caches which responded to both internal and external DNS requests. The slave machines seldom ran at a load average greater than 1, and the load put on the shared machine by DNS was negligible.

Zone file record-keeping was a fully manual process at Ziplink, which accounted for the relatively large amount of time spent managing the DNS database. Many providers do not buy commercial tools or develop custom programs for managing their DNS records. If the provider does develop tools, they will likely not be very sophisticated and will require more manual data entry than a commercially available tool.

DNS Levels and Multiple Servers

There are several reasons why there are two classes or levels of DNS servers. The Internic requires two registered nameservers. Utilizing two DNS levels reduces the chance of errors as data is entered only once instead of twice. Also, this design allows for minimal impact to the "customer facing" (machines customers use for service) servers. Under BIND, each time a zone file is updated, the nameserver must be restarted. Utilizing a two-level design, the only time customer-facing servers are restarted is when a domain is added or deleted (i.e, a change to the named.conf is required).

In a perfect world, the two DNS servers would be on separate subnets fed by different routers in widely disparate geographical locations on the provider's network. Doing so would present the highest level of redundancy. This redundancy can be taken to very high levels. Imagine having multiple machines across your network with identical IP address(es), and by the magic of routing protocols be able to route to the closest one, even to another machine entirely if the closest one is down.

DNS for a Smaller Provider

Once again, the biggest issue driving a smaller provider is cost. As a result (and by virtue of the fact they are a small provider), at most, two DNS machines are usually deployed as depicted in (see Figure 1). In very small shops, they will be shared machines, which perform other functions (mail and/or RADIUS seems to be common).

One machine, labeled "primary DNS" in Figure 1, is where all changes are made to the zone files. Often, the provider will have written a script to assist in management of the zone data, and will utilize CVS or other source management tools as well. Some nameserver traffic will be pointed at this machine, but an effort will be made to ensure most of the load gets pointed at the machine marked "slave DNS." The word "primary" indicates the machine where zone data originates.

The machine marked "slave DNS" will usually be set up as a DNS slave or caching server, obtaining all of its authoritative data (zones about which the root nameservers query it) from the machine labeled "primary DNS." Doing so ensures the data is always in sync with the primary server, so there is no difference between what the two servers report.

In this setup, all DNS queries (both on and off the provider's network) are handled by both of the nameservers. Once the network is larger, this setup will likely change, and specific machines will be dedicated to inbound and outbound requests as outlined in the next section.

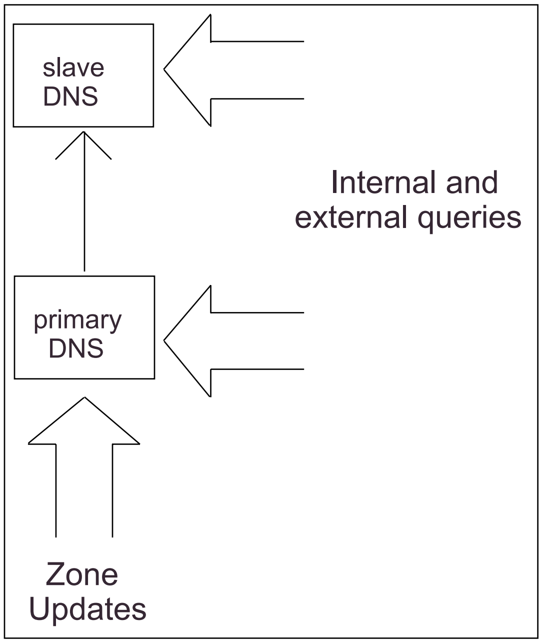

Figure 1

DNS for a Larger Provider

A larger network operator is going to be more concerned about redundancy and reliability than cost. As a result, they it will likely split their its DNS infrastructure into two pieces: one servicing internal requests (i.e., dialup ports, cable modems, DSL customers, etc.), and one servicing external requests (i.e., domains/IP addresses hosted by the provider). A bigger ISP might utilize the design shown in Figure 2 for their its external DNS traffic (requests originating outside the provider's network for domains/IP addresses hosted by the provider).

The machine marked "primary" in Figure 2 would be the single machine where all changes are made for which the provider is authoritative. No external requests would,

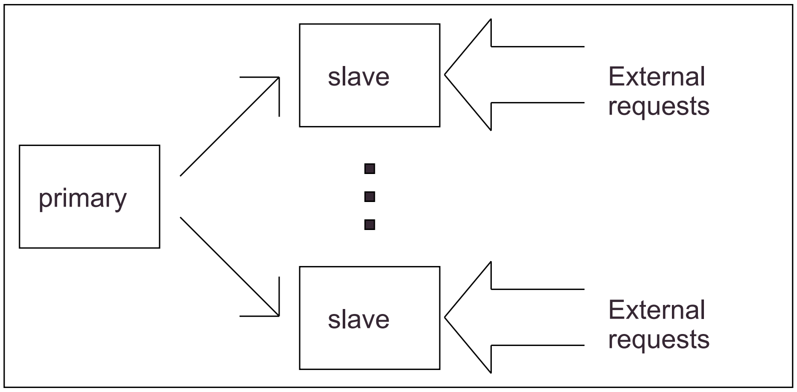

Figure 2

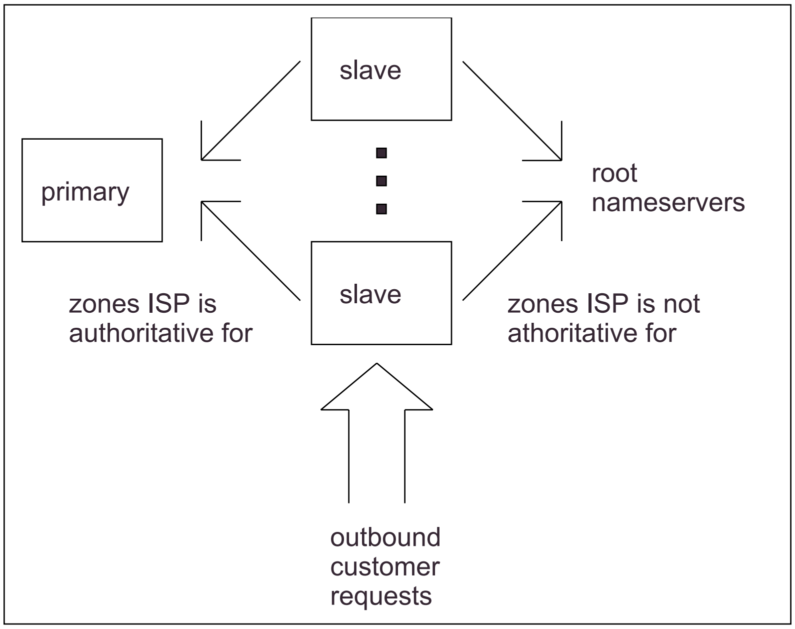

Figure 3

under normal circumstances, reach this machine. Its sole purpose is to feed data to the machines identified as "slave" which actually answer the queries coming in from networks outside of the provider's own networks. If you did a query on the root nameservers for data this provider is authoritative for, the machines labeled "slave" would show up. These "slave" machines' configuration would point to the internal machine marked "primary" in order to ensure they each reported consistent data. The "slave" machines would probably not have a pointer to the root nameservers, in order to encourage internal clients to utilize the caching/slave servers engineered expressly for this purpose.

Figure 3 illustrates how a larger provider might handle internal requests (name-service requests coming from its own "internal" network). Machines marked "slave" would be simple nameserver slave boxes, in the case of a dialup ISP deployed at the points of presence on the provider's network. The goal is to have the DNS servers as close to the end subscriber as possible. Of course, these caching servers would be like secondary servers in the sense they would be allowed to query the ISP's primary nameservers for zone data the ISP is authoritative for. Engineering DNS in this fashion enables fast access to all zones while reducing the load on the root nameservers to the extent possible.

DNS Server Software

The vast majority of ISPs, both large and small, utilize the Internet Software Consortium's (ISC) Berkeley Internet Name Domain (BIND) software. BIND has been around for many years and has been the subject of many security alerts. It would certainly be interesting to see some statistics on the usage of BIND and its alternative nameserver software, but I would guess the percentage of all sites on the Internet today utilizing BIND (or its derivatives) would be above 90%. If anyone has any pointers to such statistics, I'd love to hear from you.

BIND is considered the "reference implementation" for DNS, and the standard by which other nameservers are judged. While it has had its security issues (I am not aware of any security holes that have not been patched by the ISC), it does remain in wide use by the service provider community and in the Internet at large. The latest version of BIND is 9.1.2, which was released May 4, 2001. Quoting the ISC BIND Web site, "BIND version 9 is a major rewrite of nearly all aspects of the underlying BIND architecture." Check the ISC Web site for more information on BIND 9.

Most providers are running BIND 8, as BIND 9 will take some time to be "certified" and rolled into production. The process for certifying a new BIND version for production use could be something like the following (applicable to just about any new application in most information technology environments).

First, the provider will begin testing a new release of BIND in the lab for some period of time, enabling the staff to get familiar with the new features, bugs, etc. Once they are comfortable with the server and have come up with appropriate configurations for the production environment, a handful of low-use servers are upgraded for a few weeks. Finally, a complete rollout into all production machines is performed. All through the process, a way to get back to the previous version is preserved.

A couple of other DNS implementations bear mentioning. Perhaps the most well known is the djbdns server, by the author of qmail, Daniel J. Bernstein. Being aware of the security issues of BIND, the author has offered $500 "to the first person to publicly report a verifiable security hole in the latest version of djbdns." A less known server is Dents, an open source but not yet production-quality server. I am aware of a few providers who use djbdns, but none who are using Dents.

Another option for providers is to allow someone else to host their name service. A small provider might want to start by hosting their DNS records at a DNS provider while they focus on the rest of their business. Over the long term, however, most providers opt to host their own DNS as it is a critical part of providing Internet service. Perhaps for this reason, there are few commercial DNS service providers, and none whatsoever dedicated to the service provider market.

Namesecure is a commercial DNS service provider, but their initial focus was the name to dynamic IP resolution (for example, cable modem or server which connected via dialup for a few hours a day) resolution for end subscribers, not specifically hosting DNS services for service providers. Namesecure has since morphed into primarily a "value added" domain registrar similar to Verisign. Dynamic DNS is a free provider of DNS services, but again, their focus is almost entirely end users.

Interaction with ISP Operations

Most commercial ISP billing/provisioning systems and at least one free one (Freeside) I know of perform DNS provisioning by creating BIND-compatible configuration (named.conf) and zone files as part of their respective systems. This automation makes billing and provisioning DNS much more accurate and cost effective for the ISP.

The ISP's NOC personnel usually have access to the various nameservers to perform zone file updates and troubleshooting. This relieves engineering personnel from routine tasks and troubleshooting while giving the customer a better response time.

Network engineers at an ISP typically dictate how IP addresses are suballocated, once American Registry of Internet Numbers (ARIN) allocates a network to the ISP. Network engineering department input is usually required when provisioning new IP numbers or when setting up DNS name entries for network equipment.

Many ISPs in the recent past have shied away from allocating static IP addresses to customers due to the complexities of routing and managing this costly resource. Dialup ISPs associated with competitive local exchange carriers (CLECs) who are receiving reciprocol compensation from incumbant local exchange carriers (ILECs) encourage the use of static IP addresses. Static IP address customers tend to spend many hours online; the CLEC gets more money in the form of reciprocol compensation from the ILEC! I may cover the topic of IP addresses and related issues (ARIN, rwhois, IP address allocation/management, etc.) in a future column of ISPadmin.

Many ISPs in the recent past have shied away from allocating static IP addresses to customers due to the complexities of routing and managing this costly resource

Miscellaneous DNS Related Topics

DNS entries for the ISP's zone would vary depending upon the business plan and history of the ISP. Typical DNS entries for a dialup ISP owning the domain "isp.net" would be the following:

- www.isp.net – Web site for ~accounts
- smtp.isp.net – where customer outbound mail points to
- pop.isp.net – where customer POP clients point to
- pop3.isp.net – points to same IP as pop.isp.net
- mail.isp.net – points to same IP as pop.isp.net
- ftp.isp.net – anonymous FTP service, if provided by ISP
- news.isp.net – Usenet news machine(s)

Of course, using the magic of DNS round robin (or other load balancing mechanisms such as a layer 4 switch), multiple IP addresses can be returned for several machines providing duplicate services for redundancy or load purposes. A smaller provider would probably not have a need to do load balancing.

For hosted domains, the customer would dictate what entries should be placed into their DNS zone file. Of course, ISPs do not usually host DNS records unless the entity requesting the hosting has some sort of a business relationship with the ISP. Even with "secondarying" DNS records, usually the person requesting the secondary buys some sort of service from the ISP. There is at least one free public provider of secondary (and primary) DNS on the Internet called "The Public DNS Service" sponsored by register.com.

Network Address Translation (NAT) is a technique used by many organizations (especially enterprises) to reduce the number of IP addresses used. Typically, traditional ISPs are able to justify enough IP address space to cover their customer usage and do not deploy NAT as an enterprise would. An ISP's customer may need to deploy NAT because they doesn't want to pay the cost of additional IP address space, or the ISP doesn't have the space to allocate. Another way to reduce IP address usage is by utilizing Apache's (or other Web server's) virtual hosting capability. Name-based virtual hosting is the Web server's ability to serve multiple Web sites from one IP address. Utilizing name-based virtual hosting will drastically reduce the number of IP addresses required to serve large numbers of hosted Web sites.

Conclusion

DNS at the smaller scale is handled with two machines, a primary for making changes and responding to external requests, and a secondary for internalrequests. A larger network provider is likely to split up their DNS infrastructure: one machine to handle internal requests originating on its network and one to answer external requests not originating on its network, for from domains/IP addresses for which the ISP is authoritative. There are some free as well as commercial DNS service providers, but none aimed expressly at the service provider market. This requires most ISPs to implement and manage their own infrastructure.

Next time, I'll examine how ISPs large and small set up their Web hosting infrastructure. In the meantime, send your questions and comments regarding ISP infrastructure and system administration to me!

REFERENCES

DNS Resources Directory: *http://www.dns.net/dnsrd/*

ISC's BIND: *http://www.isc.org/products/bind*

Daniel J. Bernstein's djbdns page: *http://cr.yp.to/djbdns.html*

djbdns: *http://www.djbdns.org/*

qmail: *http://www.qmail.org/*

Dents: *http://sourceforge.net/projects/dents/*

Namesecure: *http://www.namesecure.com/*

Verisign: *http://www.verisign.com/*

Dynamic DNS: *http://www.dyndns.org/*

Freeside: *http://www.sisd.com/freeside*

ARIN: *http://www.arin.net*

The Public DNS Service: *http://soa.granitecanyon.com/*

NAT starting point: *http://linas.org/linux/load.html*

Apache virtual hosting page: *http://httpd.apache.org/docs/vhosts/*