Network Working Group
Request for Comments: 2517
Category: Informational

R. Moats
R. Huber
AT&T

February 1999

Building Directories from DNS: Experiences from WWWSeeker

Status of this Memo

Copyright Notice

Abstract

There has been much discussion and several documents written about the need for an Internet Directory. Recently, this discussion has focused on ways to discover an organization's domain name without relying on use of DNS as a directory service. This memo discusses lessons that were learned during InterNIC Directory and Database Services' development and operation of WWWSeeker, an application that finds a web site given information about the name and location of an organization. The back end database that drives this application was built from information obtained from domain registries via WHOIS and other protocols. We present this information to help future implementors avoid some of the blind alleys that we have already explored. This work builds on the Netfind system that was created by Mike Schwartz and his team at the University of Colorado at Boulder [1].

1. Introduction

Over time, there have been several RFCs [2, 3, 4] about approaches for providing Internet Directories. Many of the earlier documents discussed white pages directories that supply mappings from a person's name to their telephone number, email address, etc.

More recently, there has been discussion of directories that map from a company name to a domain name or web site. Many people are using DNS as a directory today to find this type of information about a given company. Typically when DNS is used, users guess the domain name of the company they are looking for and then prepend "www.". This makes it highly desirable for a company to have an easily

guessable name.

There are two major problems here. As the number of assigned names increases, it becomes more difficult to get an easily guessable name. Also, the TLD must be guessed as well as the name. While many users just guess ".COM" as the "default" TLD today, there are many two-letter country code top-level domains in current use as well as other gTLDs (.NET, .ORG, and possibly .EDU) with the prospect of additional gTLDs in the future. As the number of TLDs in general use increases, guessing gets more difficult.

Between July 1996 and our shutdown in March 1998, the InterNIC Directory and Database Services project maintained the Netfind search engine [1] and the associated database that maps organization information to domain names. This database thus acted as the type of Internet directory that associates company names with domain names. We also built WWWSeeker, a system that used the Netfind database to find web sites associated with a given organization. The experienced gained from maintaining and growing this database provides valuable insight into the issues of providing a directory service. We present it here to allow future implementors to avoid some of the blind alleys that we have already explored.

2. Directory Population

2.1 What to do?

There are two issues in populating a directory: finding all the domain names (building the skeleton) and associating those domains with entities (adding the meat). These two issues are discussed below.

2.2 Building the skeleton

In "building the skeleton", it is popular to suggest using a variant of a "tree walk" to determine the domains that need to be added to the directory. Our experience is that this is neither a reasonable nor an efficient proposal for maintaining such a directory. Except for some infrequent and long-standing DNS surveys [5], DNS "tree walks" tend to be discouraged by the Internet community, especially given that the frequency of DNS changes would require a new tree walk monthly (if not more often). Instead, our experience has shown that data on allocated DNS domains can usually be retrieved in bulk fashion with FTP, HTTP, or Gopher (we have used each of these for particular TLDs). This has the added advantage of both "building the skeleton" and "adding the meat" at the same time. Our favorite method for finding a server that has allocated DNS domain information is to start with the list maintained at

http://www.alldomains.com/countryindex.html and go from there. Before this was available, it was necessary to hunt for a registry using trial and error.

When maintaining the database, existing domains may be verified via direct DNS lookups rather than a "tree walk." "Tree walks" should therefore be the choice of last resort for directory population, and bulk retrieval should be used whenever possible.

2.3 Adding the meat

A possibility for populating a directory ("adding the meat") is to use an automated system that makes repeated queries using the WHOIS protocol to gather information about the organization that owns a domain. The queries would be made against a WHOIS server located with the above method. At the conclusion of the InterNIC Directory and Database Services project, our backend database contained about 2.9 million records built from data that could be retrieved via WHOIS. The entire database contained 3.25 million records, with the additional records coming from sources other than WHOIS.

In our experience this information contains many factual and typographical errors and requires further examination and processing to improve its quality. Further, TLD registrars that support WHOIS typically only support WHOIS information for second level domains (i.e. ne.us) as opposed to lower level domains (i.e. windrose.omaha.ne.us). Also, there are TLDs without registrars, TLDs without WHOIS support, and still other TLDs that use other methods (HTTP, FTP, gopher) for providing organizational information. Based on our experience, an implementor of an internet directory needs to support multiple protocols for directory population. An automated WHOIS search tool is necessary, but isn’t enough.

3. Directory Updating: Full Rebuilds vs Incremental Updates

Given the size of our database in April 1998 when it was last generated, a complete rebuild of the database that is available from WHOIS lookups would require between 134.2 to 167.8 days just for WHOIS lookups from a Sun SPARCstation 20. This estimate does not include other considerations (for example, inverting the token tree required about 24 hours processing time on a Sun SPARCstation 20) that would increase the amount of time to rebuild the entire database.

Whether this is feasible depends on the frequency of database updates provided. Because of the rate of growth of allocated domain names (150K-200K new allocated domains per month in early 1998), we provided monthly updates of the database. To rebuild the database

each month (based on the above time estimate) would require between 3 and 5 machines to be dedicated full time (independent of machine architecture). Instead, we checkpointed the allocated domain list and rebuild on an incremental basis during one weekend of the month. This allowed us to complete the update on between 1 and 4 machines (3 Sun SPARCstation 20s and a dual-processor Sparcserver 690) without full dedication over a couple of days. Further, by coupling incremental updates with periodic refresh of existing data (which can be done during another part of the month and doesn't require full dedication of machine hardware), older records would be periodically updated when the underlying information changes. The tradeoff is timeliness and accuracy of data (some data in the database may be old) against hardware and processing costs.

4. Directory Presentation: Distributed vs Monolithic

While a distributed directory is a desirable goal, we maintained our database as a monolithic structure. Given past growth, it is not clear at what point migrating to a distributed directory becomes actually necessary to support customer queries. Our last database contained over 3.25 million records in a flat ASCII file. Searching was done via a PERL script of an inverted tree (also produced by a PERL script). While admittedly primitive, this configuration supported over 200,000 database queries per month from our production servers.

Increasing the database size only requires more disk space to hold the database and inverted tree. Of course, using database technology would probably improve performance and scalability, but we had not reached the point where this technology was required.

5. Security Considerations

The underlying data for the type of directory discussed in this document is already generally available through WHOIS, DNS, and other standard interfaces. No new information is made available by using these techniques though many types of search become much easier. To the extent that easier access to this data makes it easier to find specific sites or machines to attack, security may be decreased.

The protocols discussed here do not have built-in security features. If one source machine is spoofed while the directory data is being gathered, substantial amounts of incorrect and misleading data could be pulled in to the directory and be spread to a wider audience.

In general, building a directory from registry data will not open any new security holes since the data is already available to the public. Existing security and accuracy problems with the data sources are likely to be amplified.

6. Acknowledgments

This work described in this document was partially supported by the National Science Foundation under Cooperative Agreement NCR-9218179.

7. References

[1] M. F. Schwartz, C. Pu. "Applying an Information Gathering Architecture to Netfind: A White Pages Tool for a Changing and Growing Internet", University of Colorado Technical Report CU-CS-656-93. December 1993, revised July 1994.

URL:ftp://ftp.cs.colorado.edu/pub/cs/techreports/schwartz/Netfind

[2] Sollins, K., "Plan for Internet Directory Services", RFC 1107, July 1989.

[3] Hardcastle-Kille, S., Huizer, E., Cerf, V., Hobby, R. and S. Kent, "A Strategic Plan for Deploying an Internet X.500 Directory Service", RFC 1430, February 1993.

[4] Postel, J. and C. Anderson, "White Pages Meeting Report", RFC 1588, February 1994.

[5] M. Lottor, "Network Wizards Internet Domain Survey", available from http://www.nw.com/zone/WWW/top.html

8. Authors' Addresses

Ryan Moats
AT&T
15621 Drexel Circle
Omaha, NE 68135-2358
USA

EMail: jayhawk@att.com

Rick Huber
AT&T
Room C3-3B30, 200 Laurel Ave. South
Middletown, NJ 07748
USA

EMail: rvh@att.com

9. Full Copyright Statement