Attribute Distribution and Search for Internet Resource Discovery

Michael F. Schwartz
University of Colorado - Boulder
Published in Internet Society News 1(3), Summer 1992

In the previous issue of Internet News, I pointed out that resource discovery involves two basic problems: characterizing the resources of interest using name/attribute descriptions, and distributing this information so it can be searched flexibly and efficiently. I also discussed a number of approaches to the characterization problem. The current article considers attribute distribution and search.

The most straightforward solution to the distribution/search problem is to centralize resource information. This approach is taken by Archie, which stores anonymous FTP directory listings on a central server. WAIS uses a centralized server to maintain a directory of WAIS servers. To date, centralized information has worked quite well in Archie and WAIS. Archie maintains information about nearly 1,000 Internet archive sites, and handles thousands of queries per day. There are hundreds of WAIS servers registered in the top-level directory, and new servers are added often.

The problem with a centralized solution, of course, is that the central server can become a performance bottleneck and a critical point of failure, particularly as the scale of the system increases. The difficulty in sustaining reasonable response times in the face of tremendous popular demand for Archie has moved the community to create replica servers. Doing so distributes the load, yet creates auxiliary problems of distributing the data and maintaining consistency between replicas. A future version of Archie will address these problems using "lazy" update semantics to distribute data among replicas.

To reduce the scalability and consistency problems of a fully replicated directory, one can chose a solution where only parts of the resource data are maintained on any particular server. A common approach is to impose some organizational properties on the data, and distribute data according to these properties. For example, the X.500 directory service standard divides information hierarchically. The tree is divided by country at the top level, and by administrative organization (company, university, etc.) at the next level down. Since the information in a hierarchy can be divided into arbitrarily many pieces, hierarchical directories scale well. Yet, it is only efficient to search hierarchical information according to the one way it is organized. For example, in X.500 it is efficient to find information about a person from a known country and organization, but it would be infeasible to find people according to their technical interests or other criteria that exist in the individual resource records, but that are not represented in the tree structure.

One can mimic the effect of representing multiple search criteria in a hierarchy by maintaining separate structures with symbolic links to the "main" data, but searches still require expensive distributed operations. If, on the other hand, one does not support search operations, symbolic linking can provide an acceptable mechanism. For example, the Prospero file system uses symbolic links to provide views of information in anonymous FTP and other Internet file systems. Users can browse the information, but search operations are not supported.

There are other ways to partially replicate resource attribute data beyond hierarchical distribution. One approach is to distribute information randomly among a set of servers, and cache the most popular information at each server. This approach requires that one sacrifice the ability to perform exhaustive searches. I expermented with one such protocol, optimized for locating a subset of the available copies of popular resources.

Another approach is to construct records describing particular collections of information available in a structured information space (such as a hierarchical file system) and register these records into auxiliary indices that can be searched with "flat" search operations. For example, one could register particularly important/popular directories in a large file system into indices focused on particular technical topics, so that users can search for information about these topics without regard to the organization in the main file system. This approach is used by the perspective discovery paradigm discussed in the previous issue of

this newsletter.