Application Walk-Through: Directories and Email Systems

Notes from CAMP Directory Workshop
February 3-6, 2004 Tempe Arizona

Moderated by: Michael Gettes
Scribe: Nate Klingenstein

Directory-enabled means the directory is part of the infrastructure. Sendmail can utilize LDAP as its primary database of record; this means that if user information in LDAP changes, Sendmail reflects this instantaneously; however, if LDAP goes down, then Sendmail does too.

Before the discussion can delve down into the precise attributes or the mechanics of what goes where, the set of assumptions and environment into which you're deploying must be fully understood. These will help inform the answers to the questions of which attributes are used where, in what fashion, and why.

Directory services are extremely fast at search and read operations, but will they be fast enough for your needs? Each and every e-mail will require at least one and possibly several lookups. For outbound messages, there may be other attributes in the directory that will help Sendmail rewrite the From field, such as changing gettes@foo.duke.edu into Michael.Gettes@duke.edu. Inbound messages may need to have the recipient rewritten from its original form. There are then multiple operations, such as forwarding of this message to other boxes that can occur. After understanding the scale of these hits, rough calculations can determine how hard the directory will be hit and perhaps whether there is need for a separate e-mail directory, which could either be a replica with limited indexing or a smaller, dedicated directory with special mail attributes.

Instead of limiting the system's functionality and capabilities in an attempt to limit the ways in which users can cause themselves pain, it is much wiser to develop intelligent front-ends that take every measure to prevent users from doing these things.

With effective DNS MX records and good configuration of Sendmail, very elaborate configurations can be performed. This can be very useful for politically-sensitive issues, such as allowing for the Physics department to utilize the central mail system while retaining the branding physics.u.edu. The opposite can be done as well; presentation of a universal address format, where everyone is publicly known as foo@u.edu, but mail is routed to an internal foo@physics.u.edu.

A lot of commonly used attributes are presupposed to have certain meanings and values by applications. In some instances, these are handled in a relatively consistent way and presented reasonably well; in others, where usage (such as multiple values) is unexpected or the application interprets the attribute differently, it can cause extremely erratic application behavior. This leads to times when it's good (names, phone number, etc.) to
use these standard attributes because of application behavior and others where it's bad (e-mail specific routing, etc. which can cause the client to usurp aspects of the architecture).

Indexing is very good; it's what makes LDAP so fast. Substring indexing allows for matches of *string*, but not true pattern-matching; equality matches if the value is precisely equal to the search string; presence tracks whether the attribute exists in the entry; and soundex tracks phonetic sounds to perform "sounds like" capabilities. There are many instances where indexing virtually must be used for performance reasons, but logic and an evaluation of the situation can define which type of indexing should be used. Frequently, applications will expect a given type of indexing, and this should be supported. However, every time an indexed attribute is deleted or added, the index must change too, which is a performance hit; LDAP is built much more for read operations than write operations. Indexes can take up disk space and large amounts of memory, so they need to be used judiciously.