Requirement Specification for Database based email system

Edara Pavan
04305005

Ahmed Shareef
04305003

Vaibhav Selot
04305813

September 8, 2004

Contents

1 Introduction

1.1 Purpose of this document

This document lists the software requirements for IMAP (Internet Mail Access Protocol) based mail access from a database based email system.

1.2 Scope of this document

This document discusses:

1. The functionalities provided by IMAP server.
2. How a Mail client can make use of the above functionalities.
3. An LDAP based authentication scheme.

1.3 Overview

A database based email system uses a database at its back-end. Incoming messages are stored as two parts; the header part and the body part. Since attachments are mime encoded, they are stored as a part of the body. The database contains for each mail header, the location of the corresponding body. The body is stored as a blob on the file system. This has the following advantages:

Storage Savings. A single mail addressed to multiple users (as is the case when a mailing list is used) is stored only once in the file system. The header that is stored in the database contains all the mailboxes that contain the message. This is a significant improvement when the system provides support for mailing lists.

Faster search. In case of a traditional file based email system, a search for information in the mail requires scanning a single file containing all the mails. This is inefficient if the search is for information based on a message header (e.g. To, From, Date information). In such cases, it is faster to search through the mails by querying the database which stores the header information.

2 Functional Requirements

2.1 Description

The following functionalities are provided:

2.1.1 IMAP access

An IMAP based mail system allows a client to access and manipulate electronic mail messages on the server. It permits manipulation of remote message folders (mailboxes), in a way that is functionally equivalent to local mailboxes. IMAP includes operations for creating, deleting, and renaming mailboxes, checking for new messages, permanently removing messages, searching, and selective fetching of message attributes, texts, and portions thereof. It does not specify a means of posting mail, this function is handled by a mail transfer protocol called **SMTP**.

Search for header information. Since we have a database at the backend, the search operation on remotely stored mail is faster than is the case with a file based email system.

Attachment Search. Attachments are stored as part of the body on the file system. The names of attachments may be stored as part of the message attributes in the database. A search based on attachment names is easy to do, avoiding the need to scan the message body, a substantial gain over traditional systems. The end user is provided this functionality by adding a command to the standard IMAP implementation. The search string may be encoded (as quoted-printable or Base64 as the case may be) and sent to the server as an argument to the IMAP request.

Labels. In older systems, users could organize their mails into IMAP folders. A disadvantage of this system is that it has one mail going into only one folder. In order to place a mail in two different folders, a seperate copy of it needs to be created. In our approach we do away with the concept of folders (we still have INBOX, Trash folders), and use labels. User can define labels and attach one or more labels to any message. We make use of the label information to search for mails. This is made faster by storing the label information in the database as a part of the message header. Facility to add more than a single level of labels will be added.

Client plugin for squirrelmail. Squirrelmail will be modified to allow users to manage his messages based on labels instead of folders. The plugin allows search, reorganization of mail based on labels or specific words in the body, attachments. The plugin also provides a thread-based view of the messages to the users.

2.1.2 LDAP based authentication

LDAP provides fast access to data. LDAP gives a hierarchical view to data, the user information in this case. LDAP may be used when the rate of update is very less in comparison to the rate of querying for information. This is indeed the case with users of an email system. Storing the information about users in an LDAP schema allows faster access. The front end processes of the email system, MTA and IMAP server, use LDAP to authenticate or obtain information about the users. The schema may also contain a pointer to the mailbox in the underlying relational database.

3 Non Functional Requirements

The system will be made scalable to allow for more than one IMAP server to be used. In case of high load, requests from one IMAP server can be routed to other IMAP servers.

4 System Interaction

Interaction with this system is either through raw IMAP commands or more commonly via a squirrelmail based web interface. The client provides the user with the facility to retrieve messages, delete messages or add messages to/from his message store. A seperate client plugin for squirrelmail will enable the user to use the features of the IMAP server to:

- Search for specific text in attachments.
- Search for specific words in the header or body of the message

5 Conclusion

The database based email system is an improvement over the traditional file based system in that it avoids duplication of messages, provides faster search and gives the user a better view of his message store allowing him to organize it in a better fashion.