archiveIMAP

We run a small mail server for some of our clients, some of whom retain massive amounts of e-mail. A few never sort anything, but leave it all in the Inbox, which can cause programs like Outlook to become unstable. Additionally, having a lot of mail in one folder can put a strain on the servers as users move around their various folders. Other clients use mail servers which limit the total size of an individual mail account.

Many programs allow auto-archiving of older mail. However, in the case of Outlook, this mail is stored in a local message store (.pst file) which is difficult to back up. And while Thunderbird will move the mail around on the server, it is still creating a huge message store which can negatively impact resources on both the client machine and the mail server.

The main solution we have found is to separate mail into an *active account* and an *archive account*. We create a separate mail account strictly for archival purposes, and keep the *active account* as small as possible for rapid response to the client and fewer resources used on both client and server. The *archive account* is used for permanent storage of old mail, and can be on older, slower hardware, and not automatically synchronized to the client's computer. archiveIMAP is a perl script which facilitates this.

In addition to traditional IMAP servers, archiveIMAP has been successfully used archiving mail from Microsoft Exchange and gmail. In theory, any mail server which supports an IMAP interface should be able to work. archiveIMAP queries the source and target servers for the delimiter. It is possible that one type of server may allow folders to contain characters not allowed on another, but we have not run into that yet.

Client Perspective

After your server administrator has set up archiveIMAP, you can tell her/him the following information:

- Maximum age to keep active email
- Specific folders to not archive
- Whether to delete mail from the active account which has been archived (pretty useless not to)
- Whether to delete folders which no longer contain e-mail
    - If you do this, you can specify that certain folders will never be deleted
- How to store mail in the archive account
    - Original path - mail will be stored in the original folder on the archive account, so if it came out of Inbox/Clients/acme, it will be placed in Inbox/Clients/acme in the archive account
    - Year - Mail will be stored in a folder for the year in which it was sent or received. In this case, if a message was sent in 2014, it will be stored in a folder named 2014 in the archive account
    - Month - Mail will be stored in a folder for the month (two digit month number) in which it was sent or received. So, a message sent in January of 2013 will be stored in the folder 01 on the archive server.
    - Any specific folder name - You can specify a folder name, like Archive, or Inbox where all mail should be stored.

- **Any Combination of the above** - You can specify one or more of the above, and your systems administrator will be able to archive your mail as you see fit. For example, if you are very organized and have all of your mail sorted, you could tell the sysadmin to store them as the original path, then a year and month. Thus, a message in your Inbox/Clients/acme which was received in March of 2014 could end up in Inbox/Clients/acme/2014/03

Some special Notes:

First, once a message has been placed in the archives is it difficult to get back out. Because of this, it is better to be very conservative with your definition of *old*. If you think you want to archive messages older than a year, be conservative and make it 18 months for the first time. If, at a later date, you realize that was too conservative and you really meant 15 months, your sysadmin can make that change easily. However, if you find yourself going to the archives several times a day because you set it for 6 months, it is much more difficult to change that to a year.

Second, be sure you like the way you want your archives set up. Your sysadmin can easily run a trial for you, where the archive account is created but the messages are not removed from your active account. Have your sysadmin do this, maybe for just a year or so, and see if you like it. If so, the sysadmin can delete your archive account and recreate it, then run the archiver for real. However, if you chose the Year/Month option (probably the best for most people), then decide you really wanted the original folder names, reconstructing the original structure is very, very difficult.

Last, why do I say the Year/Month option is the best for most people? A lot of times, you can remember “it was about 2014, I think.” Then, you can just do a search using your e-mail client, but limit the year to 2014. And, your Sent items are right there with your received items. If you like, the sysadmin could also set it up as Year/Month/Original Path, but that can get very complex to navigate for most situations.

Systems Administrators

First, you can get the most recent version of archiveIMAP from our subversion repository.

```
http://svn.dailydata.net/svn/sysadmin_scripts/trunk/archiveIMAP
```

I am TRYING to get used to the whole “create a branch and work on things there” but I'm not very good at it. But, the code base is pretty stable, so there should not be too many changes. Currently (2019-09-19), you can not do the deleteEmptyFolders as the code breaks when it deletes the folder you're currently in (duh). I think I have that fixed, but for now, simply set it to 0.

Check out the folder to whatever directory you want. The configuration file (archiveIMAP.yaml) needs to be in the same directory as the executable. That is hard coded, though we may make fix it later to look in /etc or /usr/local/etc if it turns out to be necessary.

The config file is basic YAML. I love YAML, but I'm always scared of messing things up, so there is a sample .cfg file which contains a sample config file in Perl hash format. You can then run the filter *confToYAML.pl* with cfg file as input and a yaml will come as output, ie

```
./confToYAML.pl < archiveIMAP.cfg.sample > archiveIMAP.yaml
```

which will, obviously, overwrite archiveIMAP.yaml, so you might want to make a backup first. I'm lazy, so I put a bunch of shortcuts into the config file, so read the README and also the commented sample files.

I'd recommend a test run first. Look in the README.

There is some decent documentation, but I'm working on better. The README file has been the source of my sysadmin documentation so far, but I'll see if I can't at least copy it here for future reference. Current version as of 2019-08-22 is v2.1.0.

If you have any fixes/enhancements, let me know (use the contact form, please). I'll steal ideas from anyone :)

This is released under Gnu 3. I thought about BSD or something else, but for now, it is Gnu gpl v3.

Uses the following libraries:

```
use Net::IMAP::Simple; # libnet-imap-simple-ssl-perl
use POSIX; # to get floor and ceil
use YAML::Tiny; # apt-get libyaml-tiny-perl under debian
use Clone 'clone'; # libclone-perl
use Hash::Merge::Simple qw/ merge clone_merge /; # libhash-merge-simple-perl
use Date::Manip; # libdate-manip-perl
use Email::Simple; # libemail-simple-perl
use Date::Parse;
use Time::HiRes;
```

On a Devuan (or any Debian derived) system, you can use the following command:

```
apt-get -y install libnet-imap-simple-ssl-perl libyaml-tiny-perl libhash-
merge-simple-perl libclone-perl libdate-manip-perl libemail-simple-perl
```

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/software/dailydata/archiveimap

Last update: **2019/09/19 07:04**