Email Aggregation using IMAP, PHP & MySQL Database

Antonius H. Carvalho

Master of Computer Application, Mumbai University, Mumbai, India.

***Abstract - In* today's world almost all people use email services provided by different vendors / providers. At times they have 2 or more email accounts, for office, personal and other uses. This leads to opening up of multiple browser tabs / windows and thus makes organization of email / accounts difficult. Though 3rd party desktop applications allow for synchronization with multiple accounts, it is not a portable solution. The aim here is to provide a similar solution but hosted as a SaaS Product or a cloud service. This will allow a single Web Application to control multiple mail services allowing for transmission and retrieval of mails from different email services using IMAP protocol and storage of bare minimum data in a database to support quick and efficient access to mails using a PHP & HTML based web client.**

Keywords: **IMAP, MySQL, SaaS, PHP, Cron, Mailbox, Email, SMTP.**

I. INTRODUCTION

IMAP is a standard for retrieving mails hosted by most modern mail servers. It is an open protocol allowing for complete synchronization of emails and folders within a user's mailbox. Most users use desktop clients like Microsoft Outlook, Mozilla Thunderbird and others or a web based application provided by the email service provider. The problem begins when a user need to be logged in to two mail accounts at the same time. This leads to multiple browser windows or browser tabs being kept open, which in turn leads to unnecessary memory consumption by the browser and in older or lower specification systems eventually leads to the system becoming too slow to use. Our goal is to find the means to aggregate multiple mailboxes / email accounts provided by multiple email service providers and provide a way to uniformly access them using a single web application. To achieve this we shall make use of PHP and its IMAP module to retrieve email headers and basic mail information using PHP-CLI scripts and Forking, which will be called periodically using Linux Cron service, the email headers and associated data will be stored in a database and tables hosted using a MySQL server. The frontend user facing application will be written in HTML using a combination of CSS, JavaScript and JQuery for the purpose of improving aesthetics and providing added functionality such as AJAX and effects which cannot be accomplished by HTML alone. Content of each email will be fetched on demand, upon opening of a mail, from the mail server. This email content will not be stored in the database as it will lead to excessive storage space usage per email account.

II. IMAP

Internet Message Access Protocol [1] (IMAP) is a communications protocol for email retrieval and storage developed by Mark Crispin in 1986 at Stanford University as an alternative to POP. IMAP, unlike POP, specifically allows multiple clients simultaneously connected to the same mailbox, and through flags stored on the server, different clients accessing the same mailbox at the same or different times can detect state changes made by other clients. It is an Application Layer Internet protocol that allows an e-mail client to access e-mail on a remote mail server. The current version is IMAP version 4 revision 1 (IMAP4rev1). An IMAP server typically listens on well-known port 143. IMAP over SSL (IMAPS) is assigned well-known port number 993.

E-mail clients using IMAP generally leave messages on the server until the user explicitly deletes them. This and other characteristics of IMAP operation allow multiple clients to manage the same mailbox. Most e-mail clients support IMAP in addition to Post Office Protocol (POP) to retrieve messages; however, fewer e-mail services support IMAP. IMAP offers access to the mail storage. Clients may store local copies of the messages, but these are considered to be a temporary cache.

Incoming e-mail messages are sent to an e-mail server that stores messages in the recipient's e-mail box. The user retrieves the messages with an e-mail client that uses one of a number of e-mail retrieval protocols. Some clients and servers preferentially use vendor-specific, proprietary protocols, but most support SMTP for sending e-mail and POP and IMAP for retrieving e-mail, allowing interoperability with other servers and clients.

ISSN: 2278 – 1323

III. MySQL

MySQL [2] is (as of July 2013) the world's second most widely used relational database management system (RDBMS) and most widely used open-source RDBMS. It is named after co-founder Michael Widenius's daughter, My. The SQL acronym stands for Structured Query Language. It is an open source project available under GNU General Public License as well as other Agreements.

MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

MySQL ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

IV. SaaS

Software as a service (SaaS) [3] is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. It is sometimes referred to as "on-demand software". SaaS is typically accessed by users using a thin client via a web browser. SaaS has become a common delivery model for many business applications, including office and messaging software, payroll processing software, DBMS software, management software, CAD software, development software, gamification, virtualization, accounting, collaboration, customer relationship management (CRM), management information systems (MIS), enterprise resource planning (ERP), invoicing, human resource management (HRM), content management (CM) and service desk management. SaaS has been incorporated into the strategy of all leading enterprise software companies. One of the biggest selling points for these companies is the potential to reduce IT support costs by outsourcing hardware and software maintenance and support to the SaaS provider.

The term "software as a service" (SaaS) is considered to be part of the nomenclature of cloud computing, along with infrastructure as a service (IaaS), platform as a service (PaaS), desktop as a service (DaaS), backend as a service (BaaS), and information technology management as a service (ITMaaS).

V. PHP

PHP [4] is a server-side scripting language created in 1995 and designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the reference implementation of PHP (powered by the Zend Engine) is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Preprocessor, which is a recursive backronym.

PHP code can be simply mixed with HTML code, or it can be used in combination with various templating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications.

The canonical PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

Despite its popularity, no written specification or standard existed for the PHP language until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014, there is ongoing work on creating a formal PHP specification.

VI. CRON

The software utility Cron [5] is a time-based job scheduler in Unix-like computer operating systems. People who set up and maintain software environments use cron to schedule jobs (commands or shell scripts) to run periodically at fixed times, dates, or intervals. It typically automates system maintenance or administration—though its general-purpose nature makes it useful for things like

ISSN: 2278 – 1323

connecting to the Internet and downloading email at regular intervals.

Cron is driven by a crontab (cron table) file, a configuration file that specifies shell commands to run periodically on a given schedule. The crontab files are stored where the lists of jobs and other instructions to the cron daemon are kept. Users can have their own individual crontab files and often there is a system wide crontab file (usually in /etc or a subdirectory of /etc) that only system administrators can edit.

Each line of a crontab file represents a job, and is composed of a CRON expression, followed by a shell command to execute.

While normally the job is executed when the time/date specification fields all match the current time and date, there is one exception: if both "day of month" (field 3) and "day of week" (field 5) are restricted (not "*"), then one or both must match the current day.

VII. Email

Electronic mail, [7] most commonly referred to as email or e-mail is a method of exchanging digital messages from an author to one or more recipients. Modern email operates across the Internet or other computer networks. Some early email systems required the author and the recipient to both be online at the same time, in common with instant messaging. Today's email systems are based on a store-and-forward model. Email servers accept, forward, deliver, and store messages. Neither the users nor their computers are required to be online simultaneously; they need connect only briefly, typically to a mail server, for as long as it takes to send or receive messages. Historically, the term electronic mail was used generically for any electronic document transmission. For example, several writers in the early 1970s used the term to describe fax document transmission. As a result, it is difficult to find the first citation for the use of the term with the more specific meaning it has today.

An Internet email message consists of three components, the message envelope, the message header, and the message body. The message header contains control information, including, minimally, an originator's email address and one or more recipient addresses. Usually descriptive information is also added, such as a subject header field and a message submission date/time stamp.

Originally a text-only (ASCII) communications medium, Internet email was extended to carry, e.g. text in other character sets, multi-media content attachments, a process standardized in RFC 2045 through 2049. Collectively, these RFCs have come to be called Multipurpose Internet Mail Extensions (MIME). Subsequent RFCs have proposed standards for internationalized email addresses using UTF-8. Electronic mail predates the inception of the Internet and was in fact a crucial tool in creating it, but the history of modern, global Internet email services reaches back to the early ARPANET. Standards for encoding email messages were proposed as early as 1973 (RFC 561). Conversion from ARPANET to the Internet in the early 1980s produced the core of the current services. An email message sent in the early 1970s looks quite similar to a basic text message sent on the Internet today.

Email is an information and communications technology. It uses technology to communicate a digital message over the Internet. Users use email differently, based on how they think about it. There are many software platforms available to send and receive. Popular email platforms include Gmail, Hotmail, Yahoo! Mail, Outlook, and many others.

Network-based email was initially exchanged on the ARPANET in extensions to the File Transfer Protocol (FTP), but is now carried by the Simple Mail Transfer Protocol (SMTP), first published as Internet standard 10 (RFC 821) in 1982. In the process of transporting email messages between systems, SMTP communicates delivery parameters using a message envelope separate from the message (header and body) itself.

VIII. SMTP

Simple Mail Transfer Protocol (SMTP) [8] is an Internet standard for electronic mail (e-mail) transmission. First defined by RFC 821 in 1982, it was last updated in 2008 with the Extended SMTP additions by RFC 5321 - which is the protocol in widespread use today.

SMTP by default uses TCP port 25. The protocol for mail submission is the same, but uses port 587. SMTP connections secured by SSL, known as SMTPS, default to port 465 (nonstandard, but sometimes used for legacy reasons).

Although electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically use SMTP only for sending messages to a mail server for relaying. For receiving messages, client applications usually use either POP3 or IMAP.

Although proprietary systems (such as Microsoft Exchange and IBM Notes) and webmail systems (such as Outlook.com, Gmail and Yahoo! Mail) use their own non-standard protocols to access mail box accounts on their own mail servers, all use SMTP when sending or receiving email from outside their own systems.

ISSN: 2278 – 1323

IX. Email Aggregation using IMAP, PHP & MySQL Database

Unlike other web based email services which provide a web based client but force the user to sign up for an email account provided by them before using their service. Initially all the users details, such as login username and password, IMAP / SMTP Server details will be stored in tables in a database. The system is designed in a manner which multiple email accounts can be linked to a single user account locally in the application.

The account is basically where a user will be provided with a username and password combination to protect their account as well as all login details for multiple IMAP email accounts. Also when the account and server details are stored in a table, an additional column will be used to store a binary value indicating account initialization. This binary value, initially set to 0, will be set to 1 when the account is initialized.

The email account initialization occurs only once during the whole process. It is triggered by a PHP script which is periodically run to check for new accounts every minute.

Cron allows the script to be executed at periodic intervals as required. The execution of this initialization script is handled by cron. The entry for each script to be executed as per schedule is entered as one line per scheduled job. These entries are stored in crontab, a file that acts as a table for cron.

e.g.: Sample crontab entry to run our script

* * * * * cd /home/test_user/mail; /usr/bin/php -f ./init.php >/dev/null 2>&1

Here the PHP interpreter is executed as a normal command line or bash application. The "-f" flag is used to instruct the interpreter that all code to be executed is specified in the file, given by the filename following the "-f" flag. The output of this is piped to "/dev/null" which essentially acts as a black hole so that any script, which needs a display or display device to be present for output, can display any message or errors.

The display is in the form of messages sent to STDOUT and STD ERROR. STDOUT displays standard messages which are not critical or not generated due to any exception that has occurred. STDERR is used to display all errors generated our output due to exceptions occurring before, during or after execution. If these messages are not piped to "/dev/null/", using 2>&1, then on any output being generated the script will immediately halt execution. Essentially the line ">/dev/null 2>&1" will pipe all output to a black hole or nothing, allowing for any script or process to be executed as a daemon or background process in Unix-like operating systems.

The initialization script, is a PHP script run as a CLI script and its duty is to find an uninitialized account when it does this it reads the accounts email and server details it then passes these values on further processing. The script first opens a connection, also known as a stream, to the IMAP server using imap_open()[9] function of PHP. This is an inbuilt PHP function designed for use with IMAP servers as well as POP3 & NNTP servers.

An example of usage of the imap_open function is: $stream = imap_open($mailbox,$user,$pass);

All there parameters are strings. $mailbox is a string having optional required and parameters, its format is: {server:port/protocol/security/flags}mailbox name".

The flags are used to specify different methods which can be used to access the mailbox. The mailbox name is the name of the folder which a user needs to access. The username and password are plain text strings. The output of the imap_open() function is a resource which is used with other IMAP functions of PHP as a parameter, it can be stored in a variable.

The next step is to use IMAP to fetch all folders existing in a user's mailbox. After all folder names are fetched, individual email headers are fetched from within each folder, using IMAP. The very first step in this chain is to list all available folders. This is done by using [10] imap_list() function.

Sample code: imap_list($stream,"{server}", pattern);

Here the function uses the open IMAP connection, given by $stream, to access the server. The "{server}" string is used as a reference, which is the same as server details specified while opening an IMAP connection. The last parameter, pattern, is usually a string of the value "*" which specifies that all folders names should be fetched from the server. The output of this is an array of folder names. The folder names are saved in a database table where they are linked to the email address and user account using foreign keys.

Once all folder names are fetched, the script proceeds to check on the number of messages held within each folder. Then using a loop the script fetches each header and stores header information such as sender, receiver, cc, subject, unread/read status, attachment/s and a Unix timestamp. In addition to this, a MD5 hash is generated so that the script

ISSN: 2278 – 1323

knows if any change to the message, such as read/unread status change, has occurred.

Once all header have been fetched the script terminates the IMAP connection.

To speed up the whole process of fetching email folder names and headers forking is used. Forking allows multiple copies of a PHP script to run simultaneously. Forking can be performed only as a CLI process locally and is not supported nor triggered by webservers when scripts are accessed via http. No special changes are needed to the script but it has been observed that MySQL connections break or are dropped when forking is used, as a work around the connection has to be established within each function and terminated at the end of each function rather than establishing it at the start of the script, which is the method of choice for normal scripts.

Forking is handled by [11] pcntl_fork(), a PHP function from the PCNTL functions group. PCNTL stands for process control, they are a set of functions that help in handling processes that are run as CLI scripts. The pcntl_fork() function causes the current script to split into a parent and child script. It generates a pid and using a control structure like an if…else if … else statement we can check if forking occurred successfully or not and if the current process is a parent or chid process.

Actual email contents will only be accessed when a user tries to access a particular mail. This will be fetched using a combination of imap_open(), imap_fetchstructure() [12], imap_bodystruct() [13] and other PHP IMAP functions, passing different parameters to them as required.

Due to the ability to store all mail headers in a database, accessing emails are much faster. It also allows for proper indexing of emails and advanced searching capability. The overall size of the database will be very small as only headers are being stored, this allows us to have many users without any space constraints.

X. REFERENCES

[1] "Internet Message Access Protocol", https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol.
[2] "MySQL", https://en.wikipedia.org/wiki/MySQL.
[3] "Software as a service", https://en.wikipedia.org/wiki/Software_as_a_service.
[4] "php", https://en.wikipedia.org/wiki/PHP.
[5] "Cron", https://en.wikipedia.org/wiki/Cron.
[6] "Email box", https://en.wikipedia.org/wiki/Email_box.
[7] "Email", https://en.wikipedia.org/wiki/Email.
[8] "Simple Mail Transfer Protocol", https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol.
[9] "imap_open", http://php.net/manual/en/function.imap-open.php.
[10] "imap_list", http://php.net/manual/en/function.imap-list.php.
[11] "pcntl_fork", http://php.net/manual/en/function.pcntl-fork.php.
[12] "imap_fetchstructure", http://php.net/manual/en/function.imap-fetchstructure.php.
[13] "imap_bodystruct", http://php.net/manual/en/function.imap-bodystruct.php.

About Author:
Antonius H. Carvalho**:** Currently pursuing Master of Computer application (MCA) degree from, Institute of Management and Computer Studies (IMCOST), Thane.

ISSN: 2278 – 1323