ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN ANH TUẤN

PHÁT TRIỂN TÍNH NĂNG LOẠI BỎ DỮ LIỆU TRÙNG LẶP (DATA DEDUPLICATION) CHO DỮ LIỆU ĐÍNH KÈM TRONG HỆ THỐNG THƯ ĐIỆN TỬ SỬ DỤNG PHẦN MỀM HMAILSERVER

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội – 2017

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN ANH TUẤN

PHÁT TRIỂN TÍNH NĂNG LOẠI BỎ DỮ LIỆU TRÙNG LẶP (DATA DEDUPLICATION) CHO DỮ LIỆU ĐÍNH KÈM TRONG HỆ THỐNG THƯ ĐIỆN TỬ SỬ DỤNG PHẦN MỀM HMAILSERVER

Ngành: Công nghệ thông tin
Chuyên ngành: Truyền dữ liệu và Mạng máy tính
Mã số: Chuyên ngành đào tạo thí điểm

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS.HOÀNG XUÂN TÙNG

Hà Nội – 2017

LỜI CAM ĐOAN

Tôi xin cam đoan nội dung trong luận văn là sản phẩm do tôi thực hiện dưới sự hướng dẫn của Thầy giáo Tiến sĩ Hoàng Xuân Tùng. Các kết quả trong khóa luận là hoàn toàn trung thực và chưa được cá nhân, tổ chức nào công bố trong bất kỳ nghiên cứu nào.

Tôi xin chịu trách nhiệm cho lời cam đoan của mình.

Hà Nội, ngày 28 tháng 05 năm 2017

Người cam đoan

Nguyễn Anh Tuấn

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Tên viết tắt	Tên đầy đủ	Ý nghĩa
	Data Deduplication	Chống trùng lặp dữ liệu
AGPLv3	Affero General Public License v3	Giấy phép xuất bản AGPL Ver3
API	Application Programming Interface	Giao diện lập trình ứng dụng
ASCII	American Standard Code for Information Interchange	Chuẩn trao đổi thông tin Hoa Kỳ
COM library	COM library	Thư viện COM
DNS	Domain Name System	Hệ thống phân giải tên miền
Email	Electronic Mail	Thư điện tử
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
IMAP	Internet Message Access Protocol	Một giao thức truy cập thư điện tử
LAN	Local Area Network	Mạng nội bộ
MD5	Message-Digest algorithm 5	Thuật toán MD5
MDA	Mail Delivery Agent	Máy chủ chuyển giao thư
MIME	Multipurpose Internet Mail Extensions	Một chuẩn internet về định dạng cho thư điện tử
MTA	Mail Transfer Agent	Máy chủ gửi thư
MUA	Mail User Agent	Phần mềm thư điện tử
POP3	Post Office Protocol Ver3	Một giao thức truy cập thư điện tử
RFC	Request for Comments	Tài liệu đặc tả các chuẩn, giao thức
SHA-1	Secure Hash Algorithm 1	Thuật toán SHA-1
SMTP	Simple Mail Transfer Protocol	Giao thức truyền tải thư điện tử đơn giản
SPF	Sender Policy Framework	Khung chính sách gửi thư điện tử dùng xác minh người gửi.
SURBL	Spam URI Realtime Blacklist	Một dạng bộ lọc danh sách chống spam

DANH MỤC CÁC BẢNG

DANH SÁCH CÁC HÌNH VẼ, ĐỒ THỊ

LỜI MỞ ĐẦU

Cùng với sự phát triển chung của toàn xã hội, công nghệ thông tin đã từng bước được phát triển và được ứng dụng rộng rãi trong thực tế. Ngày nay, mạng Internet đã phát triển thành một mạng số liệu toàn cầu cho phép nhiều loại hình thông tin truyền đi trên nó. Trong số đó, thư điện tử (email) là một dịch vụ đã và đang trở nên phổ biến hơn bao giờ hết. Email cho phép chúng ta có thể giao dịch, trao đổi các thông tin qua lại một cách nhanh chóng, chính xác với độ tin cậy cao. Tuy nhiên, do đặc thù của một hệ thống email sẽ bao gồm nhiều người dùng và một người dùng có thể nhận được email từ một hoặc nhiều người dùng khác ở trong hoặc ngoài hệ thống. Do vậy, có một vấn đề phát sinh là lượng dữ liệu trùng lặp (thông điệp thư gửi đi, tệp đính kèm,…) có thể sẽ được lưu trữ nhiều lần trên cùng một máy chủ email.

Nhận thức được tính cấp thiết của đề tài, tôi đã tiến hành nghiên cứu các phương pháp có khả năng chống trùng lặp dữ liệu để từ đó ứng dụng trong hệ thống email nhằm mục đích tối giảm sự trùng lặp dữ liệu trong việc gửi / nhận email trong một hệ thống, để từ đó tiết kiệm không gian lưu trữ máy chủ và tăng tốc độ truy xuất dữ liệu cho người dùng. Tên đề tài khóa luận của tôi là: ***"Phát triển tính năng loại bỏ dữ liệu trùng lặp (Data Deduplication) cho dữ liệu đính kèm trong hệ thống thư điện tử sử dụng phần mềm hMailServer".***

Để hoàn thành được khóa luận này, tôi xin được gửi lời cảm ơn chân thành đến Thầy giáo: TS. Hoàng Xuân Tùng, giảng viên khoa Công nghệ thông tin, Trường Đại Học Công Nghệ - Đại Học Quốc Gia Hà Nội đã luôn tận tình hướng dẫn tôi trong suốt thời gian tôi thực hiện đề tài này.

Tôi cũng xin được gửi lời cảm ơn đến tất cả các thầy giáo, cô giáo trong khoa Công Nghệ Thông Tin - Trường Đại Học Công Nghệ đã giảng dạy và trang bị cho tôi những kiến thức để tôi có thể thực hiện khóa luận.

Cuối cùng, tôi xin được gửi lời cảm ơn đến gia đình, các anh chị, bạn bè đồng nghiệp đã luôn tạo điều kiện, giúp đỡ tôi trong suốt quá trình tôi thực hiện đề tài.

Hà Nội, ngày 28 tháng 05 năm 2017

Học viên: Nguyễn Anh Tuấn

CHƯƠNG I: TỔNG QUAN VỀ DATA DEDUPLICATION, HỆ THỐNG EMAIL VÀ MỐI LIÊN QUAN

1.1. Giới thiệu về Data Deduplication.

1.1.1. Data Deduplication là gì?

Một trong những vấn đề mà doanh nghiệp quan tâm hàng đầu là dữ liệu, dữ liệu của họ luôn gia tăng từng ngày. Việc cần có các giải pháp mở rộng cũng như tối ưu hệ thống lưu trữ dữ liệu là điều cần thiết. Chống trùng lắp dữ liệu (Data deduplication) là một kỹ thuật để làm giảm lượng không gian lưu trữ cho tổ chức trong vấn đề lưu trữ dữ liệu. Kỹ thuật này giúp tiết kiệm dung lượng đĩa cứng đáng kể, và hoàn toàn không ảnh hưởng đến dữ liệu hoặc khả năng truy xuất dữ liệu.

Trong hầu hết các tổ chức, các hệ thống lưu trữ thường có chứa bản sao của nhiều mẩu dữ liệu. Cùng một tệp tin có thể được lưu ở nhiều nơi bởi nhiều người sử dụng khác nhau, hoặc hai hay nhiều tệp tin mà không phải là giống nhau vẫn có thể bao gồm nhiều phần dữ liệu giống nhau. Data deduplication sẽ loại bỏ các bản sao mà chỉ lưu lại một bản dữ liệu duy nhất.

Một cách tổng quát, Data Deduplication sẽ so sánh các đối tượng (thường là các tập tin hoặc các khối dữ liệu) và loại bỏ các đối tượng (bản sao) tồn tại trong tập dữ liệu. Như vậy, Data Deduplication chỉ lưu một bản dữ liệu duy nhất trong tập dữ liệu và thay thế các bản sao khác bằng cách sử dụng con trỏ để dẫn trở lại với bản được lưu trữ. [1]

Một ví dụ cụ thể về Data Deduplication: một hệ thống thư điện tử có thể chứa 100 các tệp tin đính kèm giống nhau (có thể trong cùng một email được gửi đi) cùng có dung lượng là 1 MB. Nếu hệ thống email được sao lưu hoặc lưu trữ, tất cả 100 file đính kèm cần được lưu trữ và do đó cần đến 100 MB không gian đĩa cứng. Khi ứng dụng kỹ thuật Data Deduplication, chỉ có một thể hiện của tập tin đính kèm là thật sự được lưu trữ, các trường hợp còn lại sẽ chỉ được tham chiếu tới bản sao lưu. Trong trường hợp này, một nhu cầu lưu trữ 100 MB có thể được giảm xuống chỉ còn 1 MB. [2]

1.1.2. Mục đích của Data Deduplication

Lợi ích chính của Data Deduplication là làm giảm số lượng ổ đĩa mà các tổ chức cần phải trang bị để lưu trữ dữ liệu. Việc loại bỏ các dữ liệu dư thừa sẽ tiết kiệm được một khoản chi phí không hề nhỏ cho mỗi tổ chức. Ở đây không chỉ có chi phí về trang bị phần cứng, mà còn cắt giảm được các chi phí liên quan như hệ thống điện nguồn, hệ thống làm mát, bảo trì, không gian đặt thiết bị. [1],[3]

Trong một vài trường hợp khác, đặc biệt là khi dữ liệu cần được lưu trữ và trao đổi qua mạng như các hệ thống lưu trữ dữ liệu đám mây, chia sẻ dữ liệu dùng chung qua mạng cục bộ hoặc internet. Kỹ thuật Data Deduplication sẽ làm tăng hiệu năng cho hệ thống, giống như là: [1],[3]

- Nếu chúng ta lưu trữ ít, chúng ta sẽ sao lưu dữ liệu ít đi, đồng nghĩa với việc các phương tiện phần cứng dùng cho sao lưu sẽ ít đi.
- Nếu chúng ta lưu trữ ít, lượng dữ liệu trao đổi qua mạng sẽ ít đi, và trong trường hợp có các sự cố, việc khôi phục lại các dữ liệu sẽ nhanh hơn do lượng thời gian giảm vì dữ liệu lưu trữ trước đó đã được loại bỏ trùng lặp.

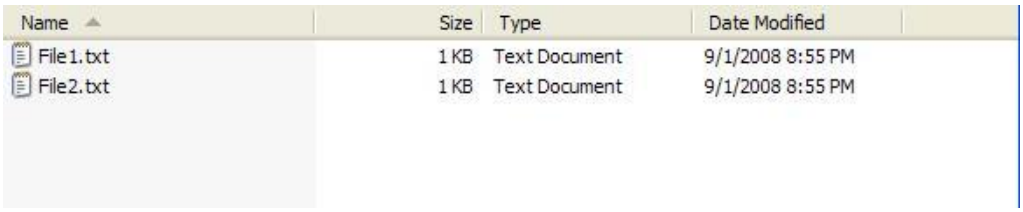
1.1.3. Phân loại Data Deduplication

Theo như tổ chức TechTarget [4-5], Việc phân loại các kiểu Data Deduplicaton có thể dựa theo hướng tiếp cận dữ liệu. Theo đó, có thể chia kỹ thuật Data Deduplication thành ba loại chính như sau:

1.1.3.1. File-level deduplication

Cách tiếp cận File-level là cách tiếp cận ở mức độ đơn giản nhất, thực hiện thông qua việc so sánh các tệp tin chuẩn bị được sao lưu hoặc lưu trữ với những tệp tin đã được lưu trữ trước đó bằng cách kiểm tra các thuộc tính của nó. Nếu tệp tin là duy nhất, tệp tin sẽ được lưu trữ và các chỉ số được cập nhật, nếu không sẽ có một con trỏ để trỏ đến tệp tin hiện đang được lưu trữ. [6]

Một ví dụ của phương thức này là so sánh tên, kích thước, kiểu và ngày chỉnh sửa của 2 tệp tin với cùng tên được lưu trữ trong hệ thống. Nếu các tham số này là trùng khớp, có thể chắc chắn rằng một vài tệp tin là bản sao của các tệp tin khác và có thể xóa một trong số chúng.



Hình 1.1. So sánh hai tệp tin dựa trên các thuộc tính của tệp tin

Như ở Hình 1.1, hai tệp tin File1.txt và File2.txt là có cùng các thuộc tính như kích thước (size), kiểu tập tin (type), ngày chỉnh sửa (date modified) cùng được lưu trong hệ thống, do đó nhiều khả năng hai tệp tin này có nội dung giống nhau.

Ngoài việc so sánh dựa trên các thuộc tính của tệp tin, chúng ta có thể sử dụng cách so sánh chính xác hơn bằng cách so sánh sự khác nhau bên trong mỗi tệp tin. Phương pháp này sẽ tạo ra một hàm băm (hash) duy nhất đại diện cho tệp tin, và sau đó so sánh hàm băm của tệp tin mới với tệp tin gốc. Nếu hai hàm băm này là như nhau thì tức là chúng giống nhau và một tệp tin cần được loại bỏ. [6]

1.1.3.2. Block-level deduplication

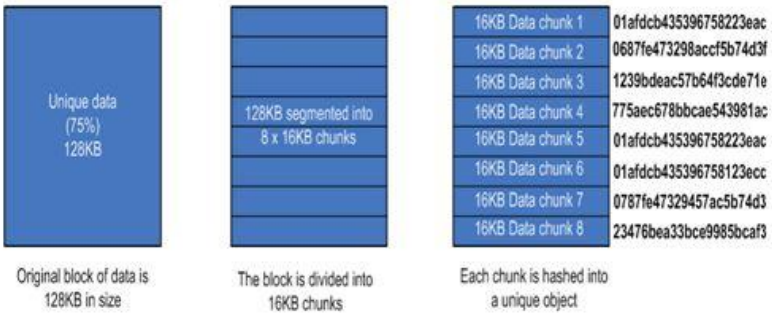
Đây là cách tiếp cận hoạt động ở mức sub-file (mức phụ file), các tập tin sẽ được chia thành các phân đoạn dữ liệu được gọi là khối (chunks hoặc blocks), sau đó các phân đoạn này sẽ được tiến hành kiểm tra về mức độ dư thừa so với các thông tin được lưu trữ trước đó. [6]

Phương pháp tiếp cận phổ biến nhất để xác định dữ liệu trùng lặp là gán một định danh cho một khối dữ liệu, sử dụng thuật toán băm. Kích thước của khối dữ liệu có thể là cố định (fixed block) hoặc có thể sử dụng khối dữ liệu có thể thay đổi được (variable-sized block). Khối kích thước cố định có thể là 8 KB hoặc có thể 64 KB, sự khác biệt ở đây là khối dữ liệu nhỏ có khả năng để xác khối dữ liệu dư thừa là cao hơn.

Nếu một tập tin dư thừa được sửa đổi và sau đó tiến hành kiểm tra lại sự dư thừa với một kích thước khối cố định sẽ rất khó để phát hiện ra các đoạn dữ liệu dư thừa bởi vì các khối trong tập tin đã được thay đổi hoặc di chuyển có sự khác biệt so với thứ tự các khối trong tập tin được lưu trữ trước đó.

Để khắc phục nhược điểm của phương pháp chia khối dữ liệu theo kích thước cố định, người ta sử dụng một phương pháp là chia khối dữ liệu theo kích thước thay đổi. Cách tiếp cận này sẽ tìm các điểm trong một tập tin để có thể phân đoạn dữ liệu cho phù hợp. Thậm chí nếu các khối thay đổi khi một tập tin thay đổi, phương pháp này có nhiều khả năng tìm thấy các đoạn dữ liệu lặp đi lặp lại. Tuy nhiên, phương pháp này sẽ tốn nhiều thời gian để xử lý và phức tạp hơn để triển khai.

Một ví dụ về hướng tiếp cận block-level như Hình 1.2:



Hình 1.2. Mô tả về phương pháp block-level (dữ liệu được chia thành các khối nhỏ)

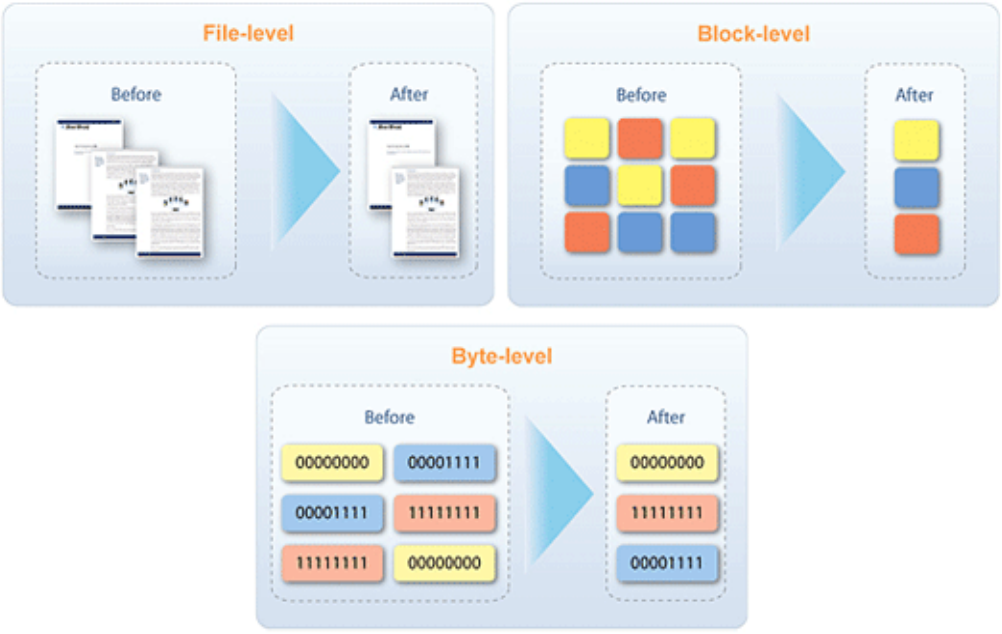
Khi dữ liệu được chia nhỏ thành các khối, sự trùng lặp có thể được hình thành và loại trừ, chỉ có một sự độc lập của mỗi khối là được lưu trữ. Như ở Hình 1.3, khối 1 và khối 5 có chỉ số hàm băm là như nhau nên một trong hai khối này sẽ được loại bỏ và chỉ lưu lại một khối duy nhất.

Hình 1.3. Mô tả về phương pháp block-level (các khối so sánh để loại phần dư thừa)

1.1.3.3. Byte-level deduplication

Đây là cách tiếp cận kiểm tra sự trùng lặp chi tiết hơn so với cách tiếp cận của Block-level, đảm bảo độ chính xác hơn nhưng thường đòi hỏi nhiều kiến thức chuyên sâu cho mỗi loại thiết bị lưu trữ để thực hiện công việc. [7]

1.1.4. So sánh các kiểu Data Deduplication



Hình 1.4. Mô phỏng các kiểu Data Deduplication [7]

1.1.4.1. So sánh File-level với Block-level Deduplication

File-level và Block-level đều có những ưu và nhược điểm riêng tùy thuộc vào các trường hợp hoạt động khác nhau: [4]

- **File-level có thể ít hiệu quả hơn so với Block-level:** Trường hợp có một sự thay đổi trong tập tin sẽ làm cho toàn bộ tập tin bị thay đổi và lưu lại. Chẳng hạn như một bài thuyết trình PowerPoint có thể có một nội dung gì đó thay đổi như một trang tiêu đề, sửa đổi ngày tháng trình bày để phản ảnh một chương trình mới, điều này sẽ dẫn đến tập tin sẽ được lưu lại ở lần thứ hai. Trường hợp này với cách tiếp cận Block-level sẽ chỉ lưu các khối thay đổi giữa một phiên bản của tập tin và các thay đổi tiếp theo.
- **File-level có thể hiệu quả hơn so với Block-level:** việc đánh chỉ mục (index) cho file-level là nhỏ hơn đáng kể so với block-level, thời gian tính toán của file-level ít hơn khi bản sao được xác định. Do đó, hiệu suất lưu trữ, sao lưu tốt hơn, ít bị ảnh hưởng bởi quá trình Data Deduplication.

1.1.4.2. So sánh Block-level với Byte-level Deduplication

Byte-level sử dụng một cách so sánh dữ liệu nguyên thủy nhất – byte by byte (so sánh các byte dữ liệu với nhau). Cách tiếp cận này thực hiện việc kiểm tra đầy đủ toàn bộ dữ liệu, bao gồm cả các phần dữ liệu dư thừa ngay cả khi dữ liệu dư thừa đó là chắc

chắn, do vậy Byte-level tốn khá nhiều thời gian trong việc kiểm tra và thường được áp dụng trong kỹ thuật post-process deduplication (phương pháp sẽ được trình bày ở phần sau). [8]

1.2. Tổng quan về hệ thống Email

1.2.1. Các khái niệm cơ bản về Email

Theo Wikipedia [9], các khái niệm cơ bản về thư điện tử (email) được mô tả:

Email: là viết tắt của chữ Electronic Mail được gọi là Thư điện tử, là một hệ thống chuyển nhận thư qua các mạng máy tính. Email là một phương tiện truyền tin rất nhanh. Một mẫu thông tin có thể được gửi đi ở dạng mã hoá hay dạng thông thường và được chuyển qua các mạng máy tính đặc biệt là mạng Internet. Nó có thể chuyển mẫu thông tin từ một máy nguồn tới một hoặc nhiều máy nhận trong cùng lúc.

Địa chỉ Email: Mỗi người sử dụng email được chỉ định bởi một tên duy nhất cho tài khoản thư điện tử. Tên này được biết đến như là địa chỉ email. Các người sử dụng khác nhau có thể gửi hoặc nhận các thông báo theo địa chỉ email. Thư điện tử thường có mẫu chung là username@domainname (tênngườisửdụng@tênmiền). Ví dụ, admin@k21vnu.com là một địa chỉ email, trong đó admin là tên tài khoản của người sử dụng và k21vnu.com là tên miền.

Phần mềm Email (email Software): là loại phần mềm nhằm hỗ trợ cho người dùng việc chuyển và nhận các mẫu thông tin (thường là dạng chữ). Thông tin có thể đưa vào phần mềm thư điện tử bằng cách thông dụng nhất là gõ chữ bàn phím hoặc bằng các phương thức khác ít dùng hơn như là dùng máy quét hình (scanner), dùng máy ghi hình số (digital camera),... Phần mềm email giúp cho việc tiến hành soạn thảo, gửi, nhận, đọc, in, xoá hay lưu giữ các thông điệp thư được dễ dàng. Có hai trường hợp sử dụng phần mềm thư điện tử, được phân biệt như sau:

- Loại phần mềm thư điện tử được cài đặt trên từng máy tính của người dùng, thường được gọi là email client. Một số phần mềm phổ biến thuộc loại này gồm: Microsoft Outlook, Microsoft Outlook Express, Netscape Comunicator, hay Eudora. Phần mềm thư điện tử này còn có tên là Mail User Agent (MUA). Một cách gọi tên thông dụng khác của phần mềm thư điện tử là ứng dụng thư điện tử (email application).
- Ngược lại, loại phần mềm thư điện tử không cần phải cài đặt mà nó được cung ứng bởi các máy chủ (web server) trên Internet gọi là WebMail, hay phần mềm thư điện tử qua Web. Một số dịch vụ email phổ biến cung cấp loại phần mềm này như là: mail.google.com, hotmail.com.

Máy chủ thư điện tử: là máy tính có nhiệm vụ cung ứng các dịch vụ thư điện tử. Máy chủ này được biết đến với tên gọi Mail Transfer Agent (MTA). Các dịch vu thư điện tử có thể được cung ứng miễn phí hay có lệ phí tuỳ theo nhu cầu và mục đích của người sử dụng.

1.2.2. Lợi ích của hệ thống Email

Email đem lại rất nhiều lợi ích cho người sử dụng, dưới đây là một số các lợi ích chính mà người dùng có thể nhận thấy trong quá trình sử dụng: [9-10]

Tốc độ cao: Vì email được chuyển qua đường Internet nên tốc độ di chuyển của email gần như là tức thời. Với các bức thư tín bình thường trong thực tế, có thể phải mất một vài ngày để thư có thể tới được địa chỉ cần thiết nhưng với email, sau khi nhấn vào nút gửi thư, người nhận đã có thể đọc được nội dung thư của người gửi chỉ sau một hoặc vài phút chờ đợi.

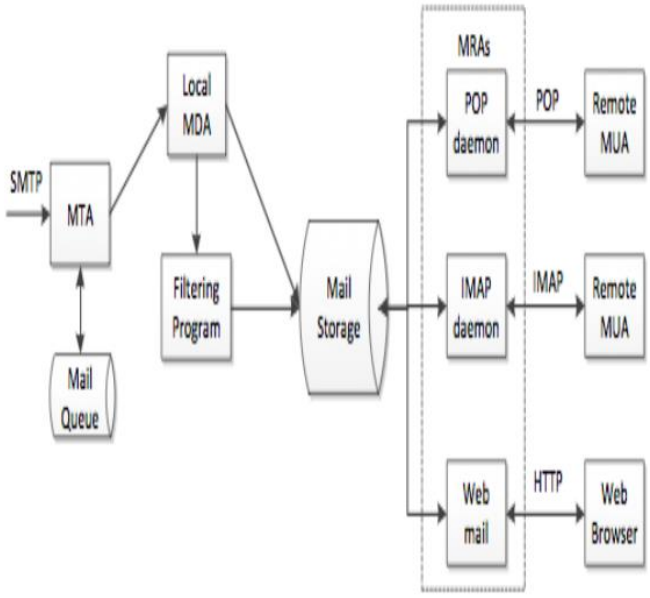
Chi phí rẻ: Với các thư tín bình thường, người dùng phải tốn một khoản chi phí khá lớn khi gửi các bức thư. Còn với email, người dùng chỉ tốn một khoản phí rất nhỏ để kết nối internet cùng với chi phí cho dịch vụ email của họ. Người dùng cũng có thể sử dụng dịch vụ email miễn phí. Khi đó chi phí của họ cho các bức thư hầu như không đáng kể và có thể coi là bằng không khi phí dịch vụ truy cập internet là cố định.

Không có khoảng cách: Với email, người nhận thư cho dù ở khoảng cách rất xa về mặt địa lý hoặc có thể đang ở ngay cùng phòng làm việc với người gửi, việc gửi và nhận thư cũng đều được thực hiện gần như ngay lập tức. Và chi phí cho các bức thư đó cũng đều rẻ như nhau.

Tiện lợi trong lưu trữ: Người dùng khi sử dụng thư điện tử có thể mở phần mềm thư điện tử ra để đọc thư bất kỳ lúc nào nên tiện lợi hơn rất nhiều so với việc phải tìm thư ở các thùng thư. Đồng thời, vì mỗi người dùng thư đều phải nhập mật khẩu vào máy nên thư điện tử sẽ khó bị người khác đọc lén so với thư gửi qua đường bưu điện. Nhưng ngược lại, các kẻ xấu có thể xâm nhập vào hệ thống thư điện tử của bất kỳ người dùng nào nếu như các mật mã hay các hệ thống an toàn cho thư điện tử kém an toàn.

1.2.3. Kiến trúc chung một hệ thống Email

Kiến trúc của một hệ thống email thông thường được mô tả như Hình 1.5: [11]



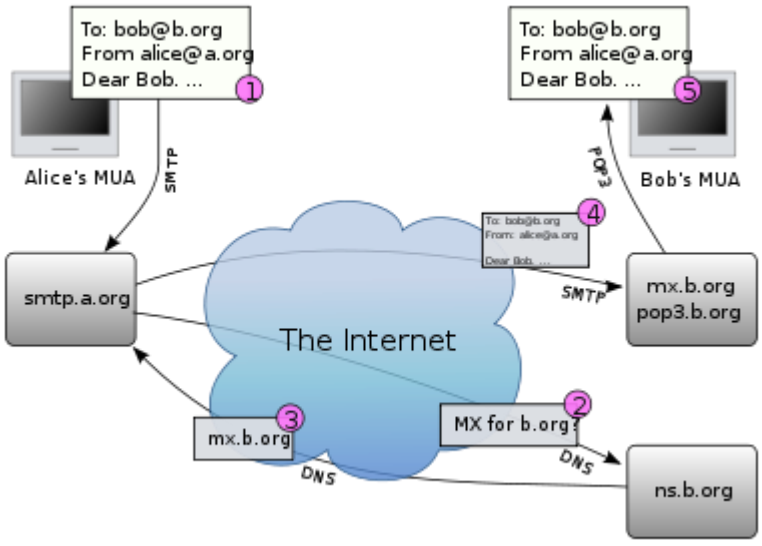
Hình 1.5. Kiến trúc chung của một hệ thống email thông thường

Trong kiến trúc này, có nhiều thành phần khác nhau được mô tả với các nhiệm vụ khác nhau nhưng có liên quan chặt chẽ đến nhau:

- MTA hoạt động như cả phía máy chủ và máy trạm, MTA sẽ gửi message thông qua Internet sử dụng giao thức SMTP. MTA cũng sẽ quản lý và lưu trữ các message trong một hàng đợi mail (mail queue), thực hiện thêm một số các xử lý và chuyển message tới Mail Delivery Agent (MDA). Các chương trình cung cấp dịch vụ MTA tiêu biểu là: Qmail, Sendmail, Postfix (Linux), Edge/Hub Tranpost của MS Exchange Server (Windows).
- MDA: sẽ nhận các mail từ MTA và chịu trách nhiệm gửi email đến đích thực sự (cuối cùng là hòm thư của người nhận). Người sử dụng có thể đăng nhập từ trình duyệt để truy xuất tới hòm thư của họ. Các chương trình cung cấp dịch vụ tiêu biểu là: Procmail, Mail.local, rmail (Linux), Mailbox Server trong MS Exchange (Windows)
- Mail Retrieval Agent (MRA): là một thành phần của máy chủ mail được thiết kế cho MUA (Message User Agent) truy xuất đến email từ xa. POP là giao thức được sử dụng để lấy mail về từ máy chủ, IMAP cũng là một giao thức cho phép truy xuất đến hòm mail khi mà tất cả các email được lưu trữ trên máy chủ. Webmail cũng tương tự như IMAP nhưng nó sử dụng giao thức HTTP. MRA bây giờ thường do các MUA đảm nhiệm đó chính là các POP3, IMAP Client.
- MUA: là các chương trình gửi và nhận mail được cài đặt trên máy người dùng, nó giúp người dùng quản lý, soạn thảo, nhận và gửi mail một cách tiện lợi và nhanh chóng. Các chương trình MUA tiêu biểu là: Outlook (Windows), Evolution (Linux), ThunderBird và Eudora.

1.2.4. Phương thức hoạt động của một hệ thống Email

Hoạt động của hệ thống email có thể dược minh họa qua phân tích một ví dụ cụ thể như Hình 1.6: [9]



Hình 1.6. Mô tả phương thức hoạt động của một hệ thống email

Khi người gửi muốn gửi một email, cần phải chỉ định rõ địa chỉ của người nhận dưới dạng user@domain.ext. Như ví dụ Hình 1.6 là bob@b.org, quá trình gửi email đến địa chỉ bob@b.org được mô tả như sau:

- Bước 1: Người gửi (alice@a.org) dùng chương trình gửi mail tại máy trạm (như Microsoft Outlook, Microsotf Outlook Express) để soạn một lá thư có địa chỉ người nhận là bob@b.org. Người gửi nhấn nút gửi thư (Send) và phần mềm thư điện tử của người gửi sử dụng giao thức SMTP để gửi lá thư đến máy chủ thư điện tử của người gửi (smtp.a.org)
- Bước 2: Máy chủ thư điện tử của người gửi sẽ đọc địa chỉ người nhận là bob@b.org và dựa vào phần tên miền (b.org) để xác định máy chủ thư điện tử của người nhận thông qua hệ thống phân giải tên miền - Domain Name System (DNS).
- Bước 3: Máy chủ DNS của domain b.org (ns.b.org) sẽ trả về các bản ghi MX cho domain b.org (mx.b.org). Bản ghi MX này xác định máy chủ của người nhận sẽ nhận email gửi đến, qua hệ thống DNS.
- Bước 4: Mát chủ smtp.a.org sẽ gửi thư của alice@a.org tới máy chủ mx.b.org sử dụng SMTP. Máy chủ mx.b.org có thể cần chuyển tiếp thư này tới các máy chủ mail khác (nếu có) trước khi thư được gửi đến hòm thư của người nhận,
- Bước 5: Để nhận thư, người nhận bob@b.org yêu cầu phần mềm thực hiện việc lấy thư trên máy chủ bằng các giao thức IMAP hoặc POP.

Trường hợp người gửi không sử dụng các phần mềm gửi thư mà dùng WebMail để gửi thì Bước 1 trong chuỗi xử lý trên sẽ không xảy ra mà thay vào đó SMTP Server của người gửi sẽ xử lý trực tiếp. Tương tự như trường hợp người nhận không sử dụng các phần mềm để nhận thư.

1.2.5. Các giao thức sử dụng trong hệ thống Email

1.2.5.1. Giao thức SMTP

SMTP (Simple Mail Transfer Protocol): là giao thức chuyển thư đơn giản. SMTP được đề xuất lần đầu tiên vào năm 1982. SMTP là một giao thức tiêu chuẩn được sử dụng cho gửi thư điện tử một cách đơn giản và đáng tin cậy thông qua internet. Đây là giao thức có nhiệm vụ vận chuyển email giữa các máy chủ trên đường trung chuyển đến địa chỉ nhận cũng như việc chuyển thư điện tử từ máy khách đến máy chủ. Hầu hết các hệ thống thư điện tử gửi thư qua Internet đều sử dụng giao thức này. Các thư điện tử có thể được truy xuất bởi một chương trình mail tại máy người dùng. Những chương trình này phải dùng giao thức POP hoặc IMAP.

Bảng 1.1. Mô tả một số các lệnh của giao thức SMTP [12]

STT	Miêu tả lệnh
1	**HELLO** Lệnh này kích hoạt cuộc hội thoại SMTP.

STT	Miêu tả lệnh
2	**EHELLO** Đây là một lệnh thay thế để kích hoạt cuộc hội thoại. ESMTP chỉ dẫn rằng Server người gửi muốn sử dụng giao thức SMTP mở rộng.
3	**MAIL FROM** Chỉ địa chỉ người gửi.
4	**RCPT TO** Xác định người nhận của thư điện tử. Để phân phối thông báo tương tự tới nhiều người sử dụng, lệnh này có thể lặp lại nhiều lần.
5	**SIZE** Lệnh này cho Server biết kích cỡ của thông báo được đính kèm bằng lượng byte.
6	**DATA** Lệnh này báo hiệu rằng một luồng dữ liệu sẽ theo sau. Tại đây, luồng dữ liệu là phần thân của thông báo.
7	**QUIT** Được sử dụng để kết thúc kết nối SMTP.
8	**VERFY** Lệnh này được sử dụng bởi Server nhận để thẩm tra xem có hay không tên sử dụng đã cung cấp là có hiệu lực hay vô hiệu lực.
9	**EXPN** Lệnh này giống với VERFY, ngoại trừ việc nó sẽ liệt kê tất cả các tên người sử dụng khi nó sử dụng với một danh sách phân phối.

1.2.5.2. Giao thức IMAP

IMAP (Internet Message Access Protocol): là giao thức truy cập thư (từ) Internet. Giao thức này cho phép truy nhập và quản lý các thư điện tử từ các máy chủ mà không cần tải thư xuống trên máy tính nội bộ. Thư điện tử sẽ được giữ và được duy trì bởi Server từ xa. IMAP cho chúng ta khả năng thực hiện bất kỳ hành động nào như tải về, xóa thư mà không cần đọc nó. IMAP cũng cho phép người sử dụng khả năng tìm kiếm các thư điện tử. Phiên bản mới nhất của IMAP là IMAP4.

Bảng 1.2. Mô tả một số các lệnh của giao thức IMAP [12]

STT	Miêu tả lệnh
1	**IMAP_LOGIN** Lệnh này mở kết nối.

STT	Miêu tả lệnh
2	**CAPABILITY** Lệnh này yêu cầu liệt kê tất cả các khả năng mà Server hỗ trợ.
3	**NOOP** Lệnh này được sử dụng như là một cuộc thăm dò định kỳ cho các thông báo mới hoặc trạng thái thông báo được cập nhật trong suốt thời gian không hoạt động.
4	**SELECT** Lệnh này giúp chọn một hòm thư để truy cập vào các thông báo.
5	**EXAMINE** Lệnh này tương tự như SELECT, ngoại trừ việc không có sự thay đổi nào tới hòm thư được cho phép.
6	**CREATE** Lệnh này được sử dụng để tạo một hòm thư với một tên cụ thể.
7	**DELETE** Lệnh này được sử dụng để xóa vĩnh viễn một hòm thư với tên đã cho.
8	**RENAME** Lệnh này được sử dụng để thay đổi tên của một hòm thư.
9	**LOGOUT** Lệnh này thông báo Server rằng Client đã thực hiện công việc đó. Server phải gửi phản hồi BYE không đính kèm thẻ ghi tên trước phản hồi OK và sau đó đóng kết nối mạng.

1.2.5.3. Giao thức POP

POP (Post Office Protocol): là giao thức tải thư về. Giao thức này được dùng để tải về các thư từ một máy chủ và hỗ trợ các truy cập ngoại tuyến đến các thư, vì thế giao thức này đòi hỏi ít thời gian sử dụng Internet hơn. POP không chấp nhận phương tiện tìm kiếm. Hiện phiên bản POP mới nhất là POP3.

Bảng 1.3. Mô tả một số các lệnh của giao thức POP. [12]

STT	Miêu tả lệnh
1	**LOGIN** Lệnh này mở kết nối.
2	**STAT** Lệnh này được sử dụng để hiển thị số thông báo hiện tại trong hòm thư.

STT	Miêu tả lệnh
3	**LIST** Lệnh này được sử dụng để nhận tổng hợp các thông báo mà mỗi thông báo được chỉ.
4	**RETR** Lệnh này giúp chọn một hòm thư để truy cập tới các thông báo.
5	**DELE** Lệnh này được sử dụng để xóa một thông báo.
6	**RSET** Lệnh này được sử dụng để khôi phục lại phiên làm việc về trạng thái ban đầu của nó.
7	**QUIT** Lệnh này được sử dụng để thoát khỏi phiên làm việc.

1.2.5.4. So sánh giữa hai giao thức IMAP và POP

Bảng 1.4. So sánh hai giao thức IMAP và POP

STT	POP	IMAP
1	Theo cách hiểu chung được sử dụng để hỗ trợ một Client.	Được thiết kế để kiểm soát nhiều Client.
2	Các thông báo được truy cập ngoại tuyến.	Các thông báo được truy cập trực tuyến mặc dù nó cũng hỗ trợ chế độ ngoại tuyến.
3	POP không chấp nhận phương tiện tìm kiếm.	IMAP cung cấp khả năng tìm kiếm các thư điện tử.
4	Tất cả thông báo phải được tải xuống.	Cho phép sự truyền tải có lựa chọn của các thông báo tới các Client.
5	Chỉ môt hòm thư có thể được tạo ra trên Server.	Nhiều hòm thư có thể được tạo ra trên Server.
6	Không thích hợp cho truy cập vào dữ liệu không phải thư điện tử.	Phù hợp cho việc truy cập vào dữ liệu không phải thư điện tử như các đính kèm.
7	Các lệnh POP thường được tóm tắt trong các mã hóa của 3 hoặc 4 chữ cái, ví dụ STAT.	Các lệnh IMAP không được tóm tắt, chúng là lệnh đầy đủ, ví dụ STATUS.

STT	POP	IMAP
8	Nó yêu cầu sự sử dụng nguồn tài nguyên trên Server nhỏ nhất.	Các Client hoàn toàn phụ thuộc vào Server.
9	Các thư điện tử một khi được tải xuống thông thường được thiết đặt không thể được truy cập từ một vài vị trí khác.	Cho phép các thư điện tử được truy cập từ nhiều vị trí.
10	Các thư điện tử không được tải xuống một cách tự động.	Người sử dụng có thể quan sát các tiêu đề và người gửi của thư điện tử và sau đó quyết định có tải thư xuống hay không.
11	POP đòi hỏi ít thời gian sử dụng internet hơn.	IMAP đòi hỏi nhiều thời gian sử dụng internet hơn.

1.2.6. Định dạng thư điện tử (Message format)

Theo Wikipedia [9], các định dạng cho thư tín điện tử hiện tại được định nghĩa bởi RFC 5322 và các đính kèm về nội dung đa phương tiện được định nghĩa trong RFC 2045 đến RFC 2049 được gọi chung là Multipurpose Internet Mail Extensions (MIME). RFC 5322 thay thế cho RFC 2822 năm 2008, trước đó RFC 2822 thay thế cho RFC 822 năm 2001 (RFC 822 là một tiêu chuẩn cho email trên Internet trong gần 20 năm). RFC 822 được xuất bản năm 1982 dựa trên RFC 733, RFC mà trước đó được xây dựng cho ARPANET (là một mạng chuyển mạch gói đầu tiên).

Định dạng cho thư điện tử gồm hai phần chính: tiêu đề thư (message header) và nội dung thư (message body). Trong đó:

- Message header được cấu trúc gồm các trường như: địa chỉ gửi (From), địa chỉ nhận (To), địa chỉ gửi kèm (CC), tiêu đề thư (Subject), ngày gửi (Date) và một vài thông tin khác về email. Trong quá trình vận chuyển thư giữa các hệ thống, máy chủ SMTP sẽ sử dụng các tham số và các thông tin thuộc phần message header.
- Phần thứ hai là message body sẽ chứa nội dung thư, giống như một kiểu văn bản phi cấu trúc, đôi khi chứa một khối chữ ký ở cuối thư.

Phần message header được tách khỏi phần message body bằng một dòng trống.

1.2.6.1. Message header

Mỗi một thư điện tử có một header, được cấu trúc thành các trường thông tin. Mỗi một trường có một tên và một giá trị được mô tả thông qua cú pháp trong tài liệu đặc tả của RFC 5322.

Một số trường thông tin trong header là bắt buộc, chẳng hạn như From, To và Date. Một số trường là tùy chọn, nhưng thường xuyên được sử dụng như Subject, CC.

Một ví dụ về message header: [13]

```
Return-Path: <example_from@dc.edu>
X-SpamCatcher-Score: 1 [X]
Received: from [136.167.40.119] (HELO dc.edu)
by fe3.dc.edu (CommuniGate Pro SMTP 4.1.8)
with ESMTP-TLS id 61258719 for example_to@mail.dc.edu; Mon, 23 Aug 2004
11:40:10 -0400
Message-ID: <4129F3CA.2020509@dc.edu>
Date: Mon, 23 Aug 2005 11:40:36 -0400
From: Taylor Evans <example_from@dc.edu>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.1)
Gecko/20020823 Netscape/7.0
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: Jon Smith <example_to@mail.dc.edu>
Subject: Business Development Meeting
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit
```

Thông thường các trường thông tin của message header là được ẩn bên trong messages và người dùng sẽ không nhìn thấy, ngoại trừ chỉ có trường tiêu đề thư (subject) là được hiển thị cho người dùng nhìn thấy.

Theo đặc tả của RFC 5322, message header có nhiều dòng và mỗi một dòng đều được bắt đầu với một ký tự có thể in ra được. Tên trường được bắt đầu với ký tự đầu tiên của dòng và được kết thúc trước ký tự dấu hai chấm ":", sau dấu hai chấm là một giá trị tương ứng với tên trường. Tên trường và các giá trị được giới hạn là 7 bít mã ký tự ASCII. Những giá trị không phải mã ASCII có thể được đại diện bởi ký hiệu mã hóa sử dụng định dạng MIME. Trong message header, một số các trường thông tin được sử dụng phổ biến nhất và giá trị của nó:

- From: tên người gửi và địa chỉ email (có bao gồm địa chỉ IP nhưng được ẩn)
- To: địa chỉ email và tên của người nhận
- Date: ngày giờ gửi email
- Subject: nội dung bất kỳ được nhập vào trước khi gửi email cho người nhận

Bên cạnh đó, message header cũng cung cấp các thông tin để định tuyến một email sẽ được chuyển từ một máy tính cần gửi tới một máy khác.

1.2.6.2. Message body

Message body là phần chính của một email. Nó chứa nội dung của thư, hình ảnh và các dữ liệu khác (như là tệp tin đính kèm). Các tệp tin đính kèm là một phần của message body, thông thường chúng được hiển thị một cách riêng biệt.

Message body có thể sử dụng các định dạng là plaint text (văn bản đơn giản) hoặc HTML (ngôn ngữ đánh dấu siêu văn bản) để soạn nội dung email gửi đi. HTML có nhiều ưu điểm hơn về mặt trình bày mang tính đồ họa, cung cấp cho người dùng nhiều sự tiện lợi hơn trong việc chỉnh sửa như về màu sắc, in đậm, in nghiêng, đặt các liên kết,… Ngày nay, trong hầu hết các trường hợp sử dụng chủ yếu là các định dạng HTML, chỉ trừ một số trường hợp đặc biệt như là hạn chế dung lương email gửi đi hoặc một lý do nào khác mà người dùng mới nên sử dụng định dạng plaint-text.

1.2.6.3. MIME format

MIME là một tiêu chuẩn Internet mở rộng định dạng cho thư điện tử, hầu như mọi thư điện tử Internet được truyền qua giao thức SMTP đều theo định dạng MIME. MIME được xây dựng để hỗ trợ: [14]

- Văn bản soạn thảo bởi các bộ ký tự khác mã ASCII
- File đính kèm phi văn bản: âm thanh, video, hình ảnh, các chương trình ứng dụng,…
- Message body với nhiều phần nội dung.
- Message header trong tập các ký tự khác mã ASCII

MIME được đặc tả trong các tài liệu RFC 2045, RFC 2046, RFC 2047, RFC 4288, RFC 4289 và RFC 2049, và tích hợp với SMTP được quy định chi tiết trong RFC 1521 và RFC 1522.

1.3. Vấn đề Data Deduplication trong các hệ thống Email

1.3.1. Lợi ích của Data Deduplication trong hệ thống Email.

Trong các hệ thống email, để trao đổi công việc hoặc thảo luận một chủ đề nào đó cho một nhóm người dùng, thông thường mỗi một tổ chức đều sử dụng một kiểu địa chỉ có thể gọi là địa chỉ nhóm được xây dựng sẵn bên trong máy chủ email. Việc sử dụng các địa chỉ email chung cho cùng một nhóm đem lại một lợi ích quan trọng trong quá trình trao đổi và thảo luận giữa các thành viên. Tuy nhiên, điều này dẫn đến một vấn đề là dữ liệu email gửi đến nhóm sẽ được lưu lại nhiều bản sao giống nhau tại hòm thư của mỗi thành viên trong nhóm.

Ví dụ: trong một Công ty có một nhóm làm việc về một dự án, mỗi thành viên sau khi làm xong phần công việc của mình sẽ gửi kết quả của phần công việc đó tới tất cả các thành viên khác trong nhóm như một tệp đính kèm. Quá trình trao đổi giữa các thành viên được diễn ra nhiều lần cho đến khi dự án hoàn thành và toàn bộ các thành viên trong nhóm sẽ cùng nhận được các tệp tài liệu đính kèm giống nhau được lưu trữ trên cùng máy chủ email.

Một vài trường hợp khác cũng dẫn đến sự trùng lặp dữ liệu lưu trữ trong hệ thống email là cùng một người nhận có thể sẽ nhận được cùng một tài liệu giống nhau từ một hoặc nhiều người gửi khác nhau hoặc sự trùng lặp dữ liệu có thể xảy ra trong trường hợp phức tạp hơn khi một email được gửi tới nhiều nhóm người dùng (gồm nhiều người nhận trong mỗi nhóm và một người có thể cùng thuộc nhiều nhóm).

Trong một hệ thống email, người dùng thường được cấp một không gian lưu trữ email của họ ở trên máy chủ. Người dùng có thể truy xuất đến email của họ từ bất kỳ thiết bị nào như là máy tính cá nhân, thiết bị di động hoặc các thiết bị xử lý thông minh khác. Trên thực tế, các công ty lớn như Google, Microsoft, Yahoo,… cung cấp một dịch vụ email cho người dùng với một không gian lưu trữ nhất định phụ thuộc vào dịch vụ của mỗi nhà cung cấp. Trong trường hợp muốn có được nhiều không gian lưu trữ hơn, người dùng phải trả thêm một khoản chi phí. Trong các trường hợp này, việc tiết kiệm không gian lưu trữ là điều vô cùng cần thiết, có ý nghĩa thiết thực cho cả người dùng và cho các nhà cung cấp dịch vụ email.

Do vậy, việc áp dụng Data Deduplication cho hệ thống email sẽ giúp loại bỏ được các dữ liệu dư thừa trong tập các dữ liệu được lưu trữ trên máy chủ email. Cũng giống như với các hệ thống lưu trữ dữ liệu khác, Data Deduplication sẽ giúp tiết kiệm không gian lưu trữ, tiết kiệm chi phí cho đầu tư đĩa cứng, chi phí bảo trì, sao lưu dữ liệu, đồng thời giúp tăng cường hiệu năng của hệ thống và rút ngắn thời gian tương tác với dữ liệu email cho người dùng.

1.3.2. Hệ thống email và khả năng Data Deduplication.

Do tính chất phổ biến của email nên ngày càng có nhiều giải pháp cung cấp dịch vụ email từ nhiều nhà cung cấp khác nhau. Từ các dịch vụ email miễn phí đến các dịch vụ email trả phí, tùy theo quy mô và nhu cầu sử dụng mà mỗi tổ chức cần lựa chọn cho mình các giải pháp sao cho hiệu quả và tối ưu nhất.

Tuy nhiên, ở góc độ các nhà cung cấp dịch vụ, dù là cung cấp giải pháp nào thì các nhà cung cấp cũng cần phải lựa chọn các nền tảng của máy chủ email để phát triển và khai thác dịch vụ. Theo tài liệu trên Wikipedia, có rất nhiều các máy chủ email với sự đa dạng về nền tảng hệ điều hành và đặc tính khác nhau: [15]

Bảng 1.5. So sánh tính năng của một số máy chủ email phổ biến hiện nay

	Hệ điều hành			**Tính năng**			**Lưu trữ**	**Giấy phép**
Mail Server	**Linux/ Unix**	**Windo ws**	**Mac OS**	**SM TP**	**POP3**	**IMAP**	**File system**	**License**
Exim	Yes	Yes (via Cygwin)	Yes	Yes	Dovec ot,UW IMAP	Dovec ot,UW IMAP	Yes	GPLv2+

	Hệ điều hành			Tính năng			Lưu trữ	Giấy phép
Mail Server	**Linux/ Unix**	**Windows**	**Mac OS**	**SMTP**	**POP3**	**IMAP**	**File system**	**License**
hMailServer	No	Yes	No	Yes	Yes	Yes	Yes	GNU AGPL
MDaemon Messaging Server	No	Yes	No	Yes	Yes	Yes	Yes	Proprietary
Mercury Mail Transport System	No	Yes	No	Yes	Yes	Yes	Yes	Proprietary donationware
Microsoft Exchange Server	No	Yes	No	Yes	Yes	Yes	Yes (up to 2003 only)	Proprietary
WinGate	No	Yes	No	Yes	Yes	Yes	Yes	Proprietary
Apache James	Yes	Yes	Yes	Yes	Yes	Yes	Yes	ASLv2
IBM Lotus Domino	Yes	Yes	No	Yes	Yes	Yes	No	Proprietary
Kerio Connect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Proprietary
Dovecot	Yes	No	Yes	No	Yes	Yes	maildir, mbox, dbox	Mixed: MIT andLGPL 2.1
Eudora Internet Mail Server	No	No	Yes	Yes	Yes	Yes	No	Proprietary
Courier Mail Server	Yes	No	Yes	Yes	Yes	Yes	maildir	GPLv3
Cyrus IMAP	Yes	No	Yes	No	Yes	Yes	Yes	4-clause BSD
Oracle Communications Messaging Server	Yes	No	No	Yes	Yes	Yes	Yes	Proprietary

	Hệ điều hành			Tính năng			Lưu trữ	Giấy phép
Mail Server	**Linux/ Unix**	**Windo ws**	**Mac OS**	**SM TP**	**POP3**	**IMAP**	**File system**	**License**
Postfix	Yes	No	Yes	Yes	Dovec ot,UW IMAP	Dovec ot,UW IMAP	Yes	IBM Public License
qmail	Yes	No	Yes	Yes	Yes	Dovec ot,UW IMAP	Yes	Public domain
Sendmail	Yes	No	Yes	Yes	Dovec ot,UW IMAP	Dovec ot,UW IMAP	Yes	Sendmail License
Zimbra	Yes	No	Yes	Yes	Yes	Yes	Yes	ZPL and proprietary editions

Như thông tin ở Bảng 1.5, chúng ta có thể thấy sự đa dạng của các máy chủ email, mỗi máy chủ được xây dựng để hỗ trợ cho một hệ điều hành hoặc đa hệ điều hành, hỗ trợ các giao thức phổ biến và hơn nữa là một số máy chủ mail được cung cấp miễn phí dưới dạng các giấy phép mã nguồn mở cho người sử dụng.

Qua việc tìm hiểu dựa trên trang thông tin chính thức (website) về các máy chủ email thì hầu như các máy chủ email chưa có sẵn các tính năng về Data Deduplication. Chỉ một số ít các máy chủ email đã được tính hợp thêm tính năng này ở những phiên bản gần đây. Chẳng hạn như:

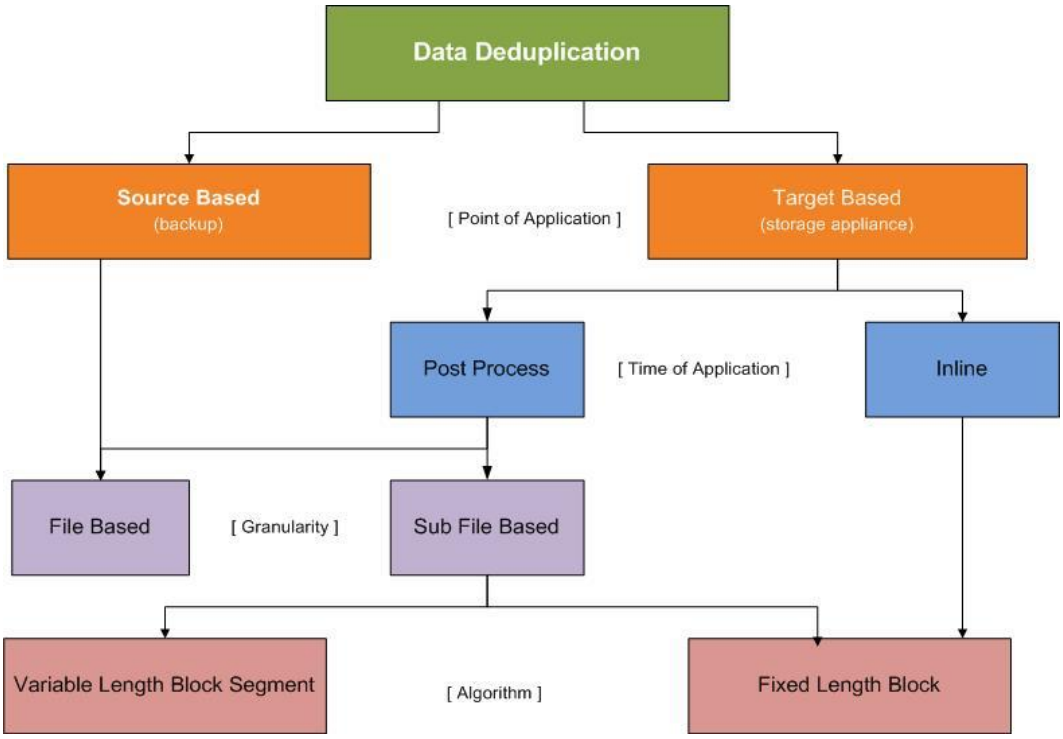
- Hệ thống email Zimbra đã tích hợp tính năng Data Deduplication trong trường hợp khi có một mail được gửi cho một nhóm người dùng thì hệ thống sẽ chỉ lưu một bản email duy nhất và các thành viên trong nhóm sẽ trỏ đến bản gốc được lưu trữ để lấy email.
- Hệ thống Dovecot từ phiên bản 2.1 trở lên được tích hợp khả năng deduplication ở các tệp tài liệu đính kèm trong email.

CHƯƠNG II: PHƯƠNG THỨC THỰC HIỆN DATA DEDUPLICATION VÀ GIẢI PHÁP CHO HỆ THỐNG EMAIL

2.1. Phương thức thực hiện Data Deduplication

Phương thức thực hiện Data Deduplication phụ thuộc vào kiểu sản phẩm và nhà cung cấp sản phẩm. Chẳng hạn như nếu kỹ thuật Deduplication được tích hợp trong một thiết bị sao lưu hoặc một giải pháp lưu trữ, quá trình thực hiện chắc chắn sẽ rất khác so với việc thực hiện thông qua một phần mềm Deduplication độc lập. [1]

Trong khi khái niệm chung về Data Deduplication là tương đối dễ hiểu thì việc ứng dụng kỹ thuật này là khá phức tạp. Kỹ thuật Data Deduplication khi triển khai thực hiện cần tham chiếu theo các yếu tố kỹ thuật như mô tả trong Hình 2.1 để có được một giải pháp triển khai cho phù hợp: [2]



Hình 2.1. Mối tương quan giữa các yếu tố kỹ thuật của công nghệ Deduplication

Theo Hình 2.1, có thể phân lớp các yếu tố kỹ thuật như sau:

- **Kiểu ứng dụng** (Point of Application): Source và Target
- **Thời điểm** (Time of Application): Inline và Post-Process
- **Mức độ chi tiết** (Granularity): File và Sub-File level
- **Thuật toán (**Algorithm**):** Fixed-size blocks và variable length data segments

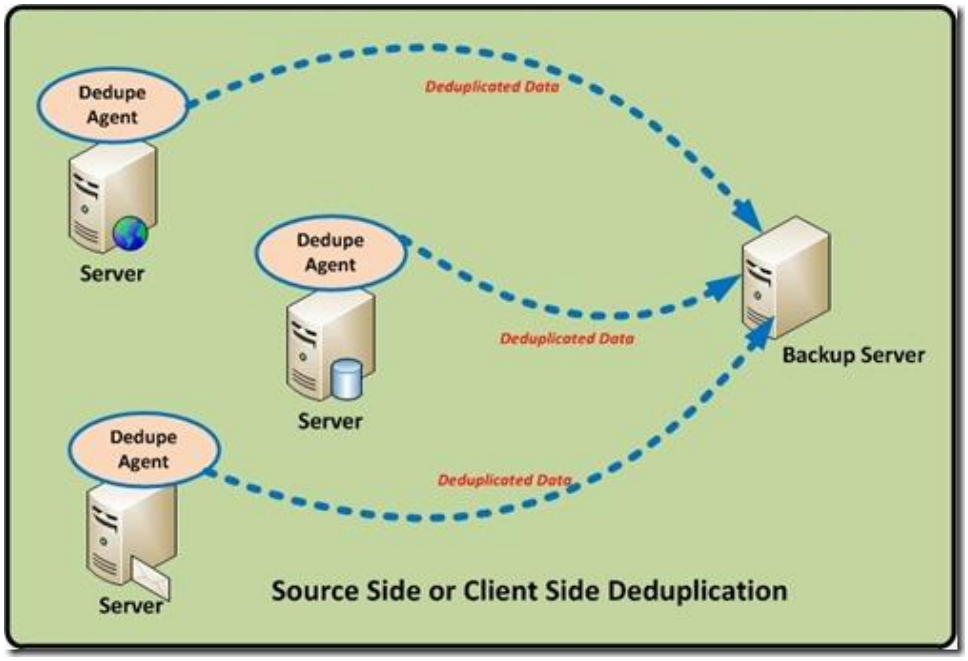
2.1.1. Source và Target Deduplication

Kỹ thuật Data Deduplication được lựa chọn thực hiện theo một trong hai cách: thực hiện bởi các phần mềm chạy trên máy tính (tại nguồn – Source Deduplication) hoặc thực hiện tại các thiết bị lưu trữ dữ liệu cần sao lưu (tại đích – Target Deduplication).

2.1.1.1. Source Deduplication

Trong trường hợp Source Deduplication, các bản sao dữ liệu trùng lặp sẽ được loại bỏ trước khi được gửi đến hệ thống sao lưu. Ưu điểm của kỹ thuật này là giảm được băng thông và thời gian cần thiết cho việc sao lưu dữ liệu. Tuy nhiên, nhược điểm là tiêu thụ nhiều tài nguyên của bộ xử lý tại nguồn dữ liệu ban đầu và sẽ khó khăn để tích hợp với các hệ thống hoặc ứng dụng đã có sẵn. Kỹ thuật Source Deduplication sử dụng phần mềm được cài đặt trên máy tính để loại bỏ dữ liệu trùng lặp. [1],[5],[16]

Lauren Whitehouse, một nhà phân tích cao cấp của Enterprise Strategy Group, đã nhận xét rằng kỹ thuật Source Deduplication là rất thích hợp cho việc sao lưu các dữ liệu từ xa và nhỏ. Ngoài ra, ông Whitehouse cũng cho rằng môi trường ảo hóa là trường hợp hiệu quả cho việc sử dụng Source Deduplication vì một lượng lớn các dữ liệu dư thừa trong các tập tin đĩa cứng của máy ảo. Tuy nhiên, nếu có nhiều máy ảo cùng được chia sẻ tài nguyên từ một máy vật lý thì việc chạy nhiều tính toán hàm băm (hash) tại cùng một thời điểm có thể dẫn đến quá tải cho các tài nguyên của máy chủ vật lý. [5]



Hình 2.2. Mô tả kỹ thuật Deduplication tại nguồn

Một số các ứng dụng nổi tiếng hiện nay được sử dụng cho việc sao lưu dữ liệu có bao gồm kỹ thuật Source Deduplication: [5]

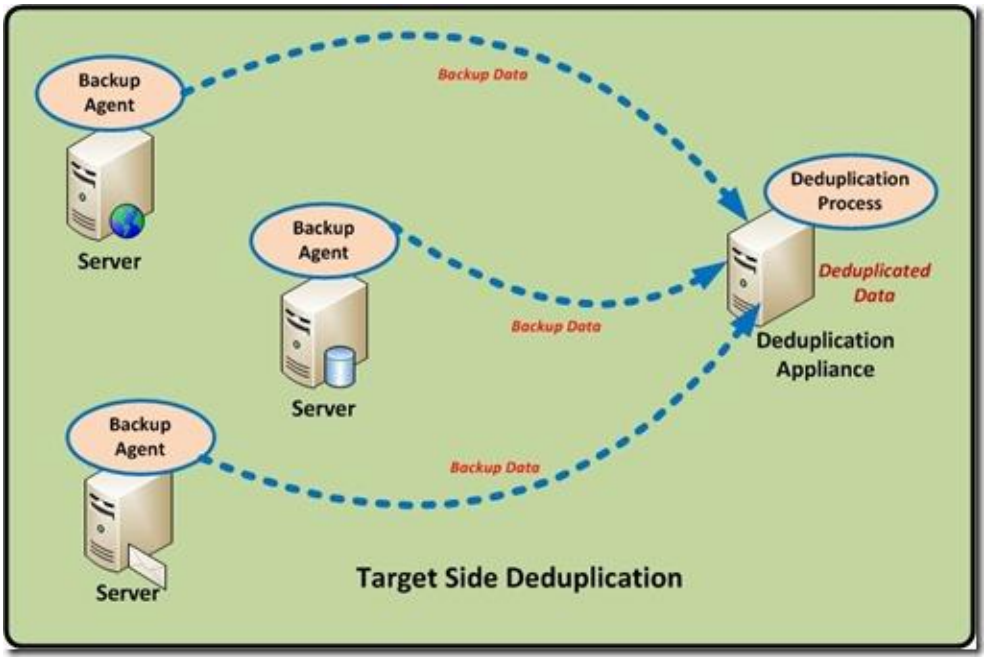
- Phần mềm NetBackup và Backup Exec của Symantec
- Phần mềm Avamar của EMC
- Phần mềm ArcServe Backup của CA
- Phần mềm Tivoli Storage Manager (TSM) with ProtecTier của IBM

2.1.1.2. Target Deduplication

Ngược lại với phương pháp loại bỏ dữ liệu trùng lặp Source Deduplication, kỹ thuật Target Deduplication sẽ loại bỏ các dữ liệu dư thừa tại các thiết bị sao lưu (backup

appliance) – thường là một thiết bị NAS (Network Attached Storage) hoặc VTL (Virtual Tape Library). Kỹ thuật này làm giảm dung lượng lưu trữ cần thiết cho sao lưu dữ liệu nhưng không làm giảm số lượng dữ liệu được gửi thông qua mạng LAN hoặc WAN trong suốt quá trình sao lưu. [1],[5],[16]

Lauren Whitehouse đã nói rằng kỹ thuật Target Deduplication có thể được sử dụng với khối lượng dữ liệu sao lưu lớn hoặc nhỏ, trong khi không làm giảm hiệu suất hoạt động của quá trình sao lưu. [5]



Hình 2.3. Mô tả kỹ thuật Deduplication tại đích

Kỹ thuật Target Deduplication có thể ứng dụng phù hợp trong môi trường nếu sử dụng nhiều ứng dụng sao lưu và một số ứng dụng không có chức năng Data Deduplication được xây dựng sẵn. Một số các hệ thống được xây dựng dựa trên Target Deduplication hiện đang sử dụng trên thực tế như là: [5]

- DXi series của Quantum
- TSM của IBM
- Hydrastor series của NEC
- File-interface Deduplication System (FDS) của FalconStor Software
- Data Domain series của EMC

2.1.2. Inline và Post-Process Deduplication

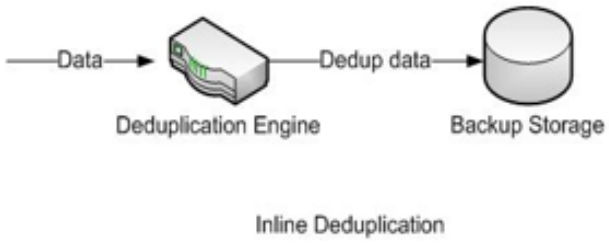
Trong kỹ thuật Target Deduplication, quá trình Data Deduplication được chia ra gồm xử lý dữ liệu trùng lặp theo thời gian thực (Inline) hoặc xử lý sau khi dữ liệu được lưu trữ trong thiết bị lưu trữ (post-process). [1]

2.1.2.1. Inline Deduplication

Inline Deduplication loại bỏ dữ liệu dư thừa theo thời gian thực như là khi dữ liệu đang được ghi vào thiết bị lưu trữ. Các sản phẩm phần mềm có xu hướng sử dụng quá

trình Inline Deduplication vì các dữ liệu sao lưu không tập trung ở một ổ đĩa trước khi nó được loại bỏ sự trùng lặp.

Ưu điểm của kỹ thuật này là tăng hiệu quả tổng thể bởi vì dữ liệu chỉ được kiểm tra và xử lý một lần. Tuy nhiên, nhược điểm của kỹ thuật này là giảm mức độ trùng lặp ít hơn và chủ yếu được sử dụng theo cách tiếp cận các khối dữ liệu có chiều dài cố định (fixed-length block). [1],[5]



Hình 2.4. Mô tả kỹ thuật Inline Deduplication

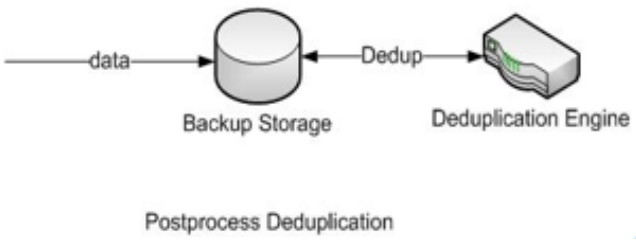
Kỹ thuật này xét về phương diện nào đó khá giống với kỹ thuật Source Deduplication khi đều làm tăng lên quá trình xử lý của bộ vi xử lý (CPU) và giới hạn tổng lượng dữ liệu cuối cùng được gửi đến thiết bị sao lưu. Một số phần mềm sử dụng kỹ thuật Inline Deduplication: [5]

- Phần mềm Cloud Backup của Asigra Inc
- Phần mềm Simpana của CommVault Systems Inc

2.1.2.2. Post-process Deduplication

Kỹ thuật Post-Process Deduplication là hoạt động loại bỏ dữ liệu trùng lặp trên tập các dữ liệu đã được lưu trữ. Kỹ thuật này có các ưu và nhược điểm ngược lại so với kỹ thuật Inline Deduplication. [1],[5]

Trong kỹ thuật Post-Process Deduplication, quá trình Data Deduplication là tách biệt với quá trình sao lưu. Vì vậy, kỹ thuật này sẽ không làm giảm hiệu năng của quá trình sao lưu dữ liệu tới thiết bị lưu trữ. Tuy nhiên, do các bản sao dữ liệu đều được truyền tới thiết bị lưu trữ trước khi chúng được loại bỏ nên cần đảm bảo băng thông cho việc truyền tải dữ liệu và không gian đĩa đủ rộng để chứa tập tất cả các dữ liệu đầy đủ và để phục vụ quá trình Data Deduplication.



Hình 2.5. Mô tả kỹ thuật Post-Process Deduplication

Một số các sản phẩm sử dụng kỹ thuật Post-Process Deduplication của một số hãng nổi tiếng trên thế giới: [5]

- Sản phẩm StorageWorks StoreOnce của Hewlett-Packard
- Hệ thống sao lưu DXi series của Quantum Corp sử dụng cả hai kỹ thuật Inline và Post-Process Deduplication.

2.1.3. File và Sub-File Level

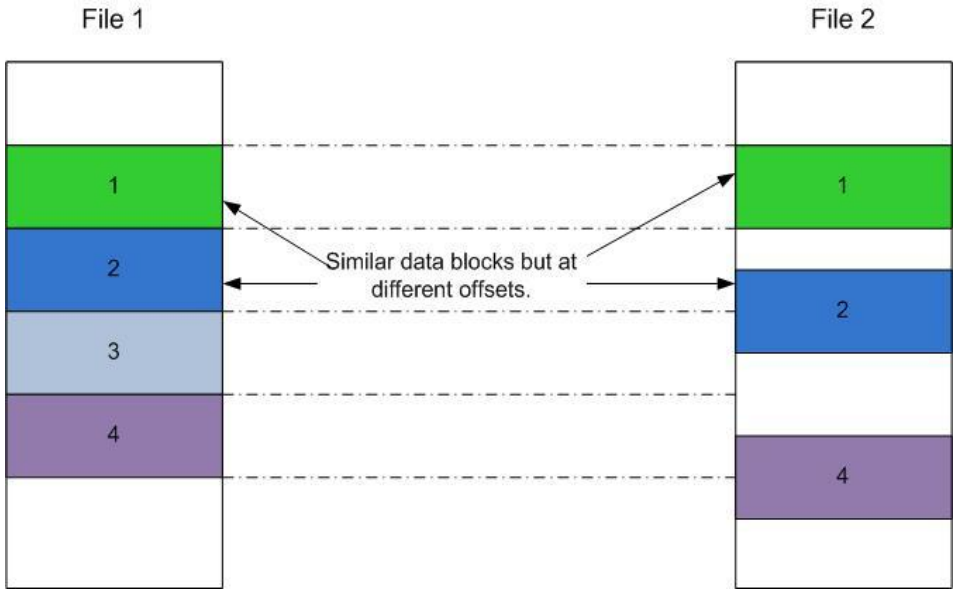
Các thuật toán loại bỏ dữ liệu trùng lặp có thể được áp dụng vào tập tin (file level) hoặc áp dụng vào từng khối dữ liệu bằng cách chia nhỏ tập tin (sub-file level). [1],[5]

File Level cho phép loại bỏ dữ liệu trùng lặp một cách đơn giản bằng cách tính checksum (phổ biến nhất là MD5 và SHA-1) của tệp dữ liệu và so sánh với checksum của những tệp dữ liệu đã được sao lưu trước đó. Đây là cách đơn giản và nhanh chóng nhưng mức độ chống trùng lặp là ít hơn, cách này không giải quyết được trường hợp có sự trùng lặp tìm thấy bên trong các tệp dữ liệu.

Sub-File Level là kỹ thuật loại bỏ dữ liệu trùng lặp bằng cách chia nhỏ các tập tin thành các khối (blocks) có kích thước cố định (fixed size block) hoặc có kích thước độ dài thay đổi (variable size block), sau đó sử dụng một thuật toán băm (hash-based algorithm) tiêu chuẩn để tìm thấy các khối dữ liệu tương tự và loại bỏ chúng. [1],[5]

2.1.4. Fixed-Length Blocks và Variable-Length Data Segments

Fixed-length Blocks là hướng tiếp cận theo khối dữ liệu chiều dài cố định, tức là tiến hành chia tệp tin đầy đủ thành các khối có chiều dài cố định và thực hiện các hàm tính toán checksum (như MD5 hoặc SHA) để tìm thấy bản sao trùng lặp. Mặc dù phương pháp này cho phép tìm kiếm các khối dữ liệu lặp đi lặp lại nhưng có thể có nhiều hạn chế do trong các tệp dữ liệu có thể có những khối dữ liệu trùng nhau nhưng các phân đoạn dữ liệu không phải lúc nào cũng trùng nhau. Ví dụ như Hình 2.6, hai khối dữ liệu của hai tệp "file 1" và "file 2" có các khối dữ liệu tương tự nhưng lại khác vị trí (offset).



Hình 2.6. Khối dữ liệu tương tự nhau nhưng có thể khác vị trí

Bởi vậy, hạn chế lớn nhất của phương pháp này là hai bộ dữ liệu với một số lượng nhỏ của sự khác biệt có thể có rất ít khối chiều dài cố định giống hệt nhau.

Kỹ thuật Variable-Length Data Segment là một phương pháp phân chia các dòng dữ liệu thành các phân đoạn dữ liệu có chiều dài thay đổi được, phương pháp này cho phép tìm thấy các ranh giới khối giống nhau trong các ngữ cảnh và vị trí khác nhau. Điều này giúp cho việc phát hiện và loại bỏ các khối dữ liệu dư thừa được đầy đủ hơn. [1],[5],[17]

2.1.5. Thuật toán băm (Hash-based Algorithms)

Phương pháp loại bỏ dữ liệu dư thừa dựa trên hàm băm sẽ xử lý các phần của dữ liệu bằng một thuật toán băm (hash algorithms), điển hình nhất là MD5 và SHA-1. Phương pháp này tạo ra một chỉ số duy nhất cho mỗi phần của dữ liệu, sau đó chỉ số này sẽ được đem so sánh với một chỉ số băm khác hiện có. Nếu chỉ số băm đó đã tồn tại trên các chỉ số thì dữ liệu sẽ không cần phải lưu trữ. Ngược lại, chỉ số băm đó sẽ được thêm vào và dữ liệu sẽ được lưu trữ. [1]

Thuật toán SHA-1 ban đầu được đưa ra để tạo chữ ký mật mã cho các ứng dụng bảo mật. SHA-1 tạo ra một chỉ số băm 160 bít và là duy nhất cho mỗi phần dữ liệu.

Thuật toán MD5 là một hàm băm 128 bít và cũng được thiết kế cho mục đích mã hóa. Tuy nhiên, SHA-1 được xem như là thuật toán bảo mật tốt hơn so với MD5.

Trên thực tế, một số các nhà cung cấp không nhất thiết phải sử dụng các thuật toán băm có sẵn như MD5 hoặc SHA-1. Thay vào đó, họ sử dụng các phương thức tùy chỉnh để xác định dữ liệu trùng lặp như là sử dụng một thuật toán băm khác của chính họ kết hợp với các phương pháp khác. Ví dụ như hai nhà cung cấp Diligent và Sepaton đã sử dụng một phương pháp tùy chỉnh để xác định dữ liệu dư thừa và kết hợp với việc so sánh ở cấp độ bít (bit-level). [6]

2.2. Một số các sản phẩm ứng dụng Data Deduplication

Như vậy, để có được thành công khi triển khai kỹ thuật Data Deduplication cần kết hợp nhiều yếu tố một cách phù hợp. Nhìn chung, các yếu tố chính ảnh hưởng đến kỹ thuật Data Deduplication gồm có:

- Số lượng các ứng dụng và số lượng người dùng cuối tạo ra dữ liệu
- Tổng sổ dữ liệu và sự thay đổi hàng ngày của dữ liệu
- Kiểu dữ liệu (thư điện tử, tài liệu, dữ liệu về âm thanh,…)
- Chính sách sao lưu dữ liệu (hàng ngày, hàng tuần, đầy đủ,…)
- Thời gian lưu trữ dữ liệu trong bao lâu

Trên thực tế hiện nay, có một số lượng lớn các sản phẩm có chức năng Data Deduplication từ nhiều nhà cung cấp khác nhau. Người sử dụng có thể lựa chọn một sản phẩm phần cứng hoặc phần mềm với các đặc trưng công nghệ thuộc một trong các kỹ thuật được mô tả ở trên sao cho phù hợp với ứng dụng triển khai và nhu cầu của mỗi tổ chức. [6]

Bảng 2.1. So sánh các sản phẩm deduplication của một số các nhà cung cấp

Nhà cung cấp (Vendor)	**Phần cứng hoặc Phần mềm (H/w or S/w)**	**Thiết bị lưu trữ sử dụng (VTL, NAS, SAN,…)**	**Thuật toán sử dụng (Algorithm)**	**Thời điểm (Inline or post-process)**	**Kiểu công nghệ (Source or target)**
Copan	H/w	VTL and NAS	SHA-1	Post-process	Target
Data Domain	H/w	VTL and NAS	SHA-1	Inline	Target
Dell/Equallogic	See Exagrid	-	-	-	-
EMC	H/w	VTL, NAS, SAN attached	SHA-1 and MD5	Post-process	Target
EMC/Avamar	S/w	-	SHA-1 and MD5	Inline	Source
ExaGrid	H/w	NAS	-	Post-process	Target
FalconStor	both	VTL and NAS	SHA-1 with optional MD5	Post-process	Target
Fujitsu	See Avamar	-	-	-	-
HP	H/w	VTL	SHA-1	Inline	Target
Hitachi Data Systems (HDS)	See Diligent and Exagrid	-	-	-	-
IBM/Diligent	S/w	VTL	Custom	Inline	Target
NetApp	S/w (in OS)	NAS/SAN	Custom	Both	Both
Overland Storage	H/w	VTL	Custom	Inline	Target
Pillar Data Systems	See Data Domain, Diligent, Falconstor, Symantec	-	-	-	-
Quantum/ADIC	Both	VTL and NAS	MD5	Both	Target
Sepaton	S/w	VTL	Custom	Post-process	Target

Spectra Logic	See Falconstor	-	-	-	-
Sun/StorageTek	See Falconstor	-	-	-	-
Symantec	S/w	-	SHA-1	Inline	Source

2.3. Giải pháp chống trùng lặp dữ liệu trong Email

Phương thức thực hiện Data Deduplication là một sự kết hợp của nhiều yếu tố. Để có một giải pháp Data Deduplication hiệu quả nhất cần có một giải pháp phù hợp giữa các yếu tố này.

Bên cạnh đó, kỹ thuật Data Deduplication khi được áp dụng vào một loại ứng dụng cụ thể cũng cần phải xem xét việc lưu trữ, xử lý dữ liệu trên mỗi ứng dụng được thực hiện như thế nào để có được một phương án triển khai phù hợp.

Đối với hệ thống email, dữ liệu lưu trữ là các nội dung trao đổi giữa các người dùng trong và ngoài hệ thống. Đối với đa số các máy chủ email thì các dữ liệu này được lưu trữ trên đĩa cứng của máy chủ. Như chúng ta đã phân tích trong chương 1, một thông điệp thư điện tử gồm có hai phần chính là message header và message body. Trong đó, message header là phần tiêu đề chứa các thông tin liên quan đến quá trình gửi / nhận và giúp định tuyến cho email được gửi đến đích, message body là phần nội dung chính của email có thể bao gồm nhiều loại nội dung khác nhau như văn bản, hình ảnh, liên kết, các tệp đính kèm,… Như vậy, có thể nhận thấy rằng phần dữ liệu thuộc message body là phần dữ liệu chiếm đa số dung lượng trong một email và có thể có các dữ liệu dư thừa chiếm tỷ lệ cao. Trong phần dữ liệu này thì các tệp tin đính kèm thường là phần có dung lượng lớn nhất so với các phần dữ liệu còn lại.

Để xây dựng được một giải pháp tốt về Data Deduplication cho hệ thống email, chúng ta cần phải tìm hiểu rất kỹ về kiến trúc, các luồng xử lý, lưu trữ dữ liệu của mỗi máy chủ email để từ đó xác định được các trường hợp có thể sẽ xảy ra trùng lặp và sau đó thiết kế một giải pháp phù hợp cho mỗi máy chủ email.

Một cách chung nhất cho các máy chủ email, có thể nhận thấy rằng có ba trường hợp có thể dẫn đến dư thừa dữ liệu và các giải pháp cho từng trường hợp như sau:

- **Trường hợp 1:** Dữ liệu dư thừa xuất hiện khi người gửi tiến hành gửi email cho một nhóm người dùng (gồm nhiều người nhận).

Đây là trường hợp thường hay gặp nhất trong thực tế. Để loại bỏ dữ liệu dư thừa trong trường hợp này có thể tiến hành loại bỏ dữ liệu là toàn bộ nội dung email gửi đi hoặc chỉ loại bỏ phần dữ liệu giống nhau bên trong email gửi đi (như là tệp tin đính kèm). Khi đó, máy chủ email chỉ giữ lại một bản duy nhất cho email gửi đi hoặc lưu trữ

duy nhất một lần cho các tệp tin đính kèm, các người dùng trong danh sách nhận được email sẽ được đặt một con trỏ đến vùng dữ liệu đã lưu trữ này.

Trong trường hợp này, khi tiếp cận Data Deduplication ở mức độ File-level sẽ đạt được hiệu quả cao, ngoài ra cũng có thể sử dụng Data Deduplication ở mức độ Block-level nhưng sẽ tốn thời gian xử lý hơn mà hiệu quả đem lại chưa chắc cao hơn File-level.

❖ **Trường hợp 2:** Dữ liệu dư thừa xuất hiện khi người dùng nhận được cùng một email từ nhiều người gửi khác nhau.

Trường hợp này khác với trường hợp thứ nhất là dữ liệu dư thừa được giới hạn chỉ xuất hiện trên hòm thư của một người dùng, dữ liệu dư thừa sẽ xảy ra khi các mail gửi đến sau có nội dung trùng với email gửi đến trước. Một vấn đề khó xử lý ở đây là làm thế nào để xác định được các phần dữ liệu giữa các email này có sự trùng lặp. Trong trường hợp này, hướng tiếp cận Data Deduplication ở mức độ Block-level sẽ đem lại hiệu quả cao hơn so với mức độ File-level.

❖ **Trường hợp 3:** Dư liệu dư thừa xuất hiện khi email được gửi tới nhiều nhóm người dùng cùng lúc (gồm nhiều người nhận trong mỗi nhóm và mỗi người nhận có thể cùng thuộc nhiều nhóm).

Trong thực tế, trường hợp này xảy ra ít hơn hai trường hợp trên. Tuy nhiên, trường hợp này sẽ trở nên phức tạp khi cùng lúc có sự kết hợp giữa hai trường hợp trên (như là một người nhận thuộc nhiều nhóm khác nhau và một email của người gửi được gửi tới nhiều nhóm cùng lúc). Trong trường hợp này, tùy theo mức độ dữ liệu dư thừa mà các kỹ thuật Data Deduplication sử dụng có thể là tối ưu hoặc chưa được tối ưu.

2.4. Đề xuất lựa chọn hMailServer để thực nghiệm

Khi triển khai giải pháp email, mỗi một tổ chức có thể lựa chọn sử dụng giải pháp email miễn phí hoặc trả phí. Tuy nhiên, Nếu một tổ chức muốn có một hệ thống email chuyên nghiệp có dạng tênngườidùng@tênmiềncôngty thì chắc chắn tổ chức đó phải sở hữu một tên miền riêng và đồng thời có một hệ thống email riêng.

Khi nói đến hệ thống email riêng cho một tổ chức, chúng ta không thể không nhắc đến các giải pháp email nổi tiếng chẳng hạn như:

- Microsoft Exchange, Mdaemon, Kerio Connect, IBM Lotus Domino, hMailServer là các giải pháp mail chạy trên hệ điều hành Windows
- Postfix, Qmail, Sendmail, Dovecot, Zimbra, Cyrus IMAP là các giải pháp mail chạy trên hệ điều hành Linux và Mac OS

Trong số các giải pháp email trên, đối với các giải pháp email dành cho hệ điều hành Windows, hầu hết là được cung cấp dưới dạng trả phí (tức là người dùng phải trả phí bản quyền để sử dụng). Trong khi đó, các giải pháp email cho Linux và Mac OS có nhiều giải pháp email được cung cấp miễn phí cho người dùng kèm theo các điều khoản sử dụng nhất định (vui lòng xem chi tiết tại bảng 1.2 thuộc chương 1).

Trong phạm vi thực hiện khóa luận, để thực hiện tích hợp thêm tính năng Data Deduplication thì cần phải lựa chọn một giải pháp email mà cho phép mở rộng thêm tính năng. Trong số các giải pháp email hiện có, có thể thấy rằng giải pháp hMailServer dành cho môi trường Windows; giải pháp Dovecot, Cyrus IMAP, Zimbra dành cho môi trường Linux, Mac OS là những sự lựa chọn phù hợp nhất.

Tuy nhiên, do tính phổ biến và phù hợp với nhiều đối tượng người dùng nên trong khóa luận này, tôi đã lựa chọn hMailServer để triển khai thực nghiệm tính năng Data Deduplication. hMailServer được biết đến là một giải pháp mã nguồn mở miễn phí cho hệ điều hành Windows. So với các giải pháp email nổi tiếng khác thì hMailServer có phần hạn chế hơn về mặt tính năng, nhưng bù lại hMailServer là một giải pháp miễn phí được thiết kế đặc biệt phù hợp với những doanh nghiệp vừa và nhỏ (số lượng người dùng khoảng dưới 1000 người). hMailServer hỗ trợ cài đặt, cấu hình đơn giản hơn rất nhiều so với Exchange và Mdaemon. Ngoài ra, hMailServer có một cộng đồng người sử dụng giúp dễ dàng trao đổi kinh nghiệm và phù hợp cho việc phát triển một hệ thống mã nguồn mở.

CHƯƠNG III: TÍCH HỢP TÍNH NĂNG DEDUPLICATION TRONG HỆ THỐNG HMAILSERVER

3.1. Tổng quan về hMailServer

3.1.1. Giới thiệu về hMailServer

hMailServer là một máy chủ email miễn phí, một bộ nguồn mở dành cho hệ điều hành Microsoft Windows. hMailServer được sử dụng bởi các nhà cung cấp dịch vụ Internet, các công ty, chính phủ, trường học và những người đam mê ở nhiều nơi trên thế giới. hMailServer hỗ trợ các giao thức e-mail phổ biến (IMAP, SMTP và POP3) và dễ dàng tích hợp với nhiều hệ thống Webmail hiện tại. hMailServer có cơ chế bảo vệ linh hoạt trước thư rác dựa trên số điểm và có thể tích hợp hệ thống quét virus ngay trên thiết bị của người dùng để quét tất cả các email gửi đến và gửi đi. [18-19]

hMailServer được sáng lập và phát triển bởi Martin Knafve, phiên bản đầu tiên được phát hành năm 2002, phiên bản mới nhất khi thực hiện luận văn là 5.6.5 build 2367 được phát hành ngày 07/06/2016. hMailServer được viết bằng ngôn ngữ C++ và C#.

hMailServer được cấp phép theo AGPLv3 và có thể được sử dụng miễn phí trong hầu hết các tình huống thương mại. Bộ mã nguồn của hMailServer được lưu trữ trên GitHub tại địa chỉ: https://github.com/hmailserver/hmailserver

hMailServer có tích hợp sẵn bộ tính năng chống thư rác như SPF, SURBL. hMailServer cũng cho phép tích hợp với các hệ thống chống thư rác của bên thứ ba như là SpamAssassin và ASSP.

Bất kỳ một hệ thống Webmail có hỗ trợ IMAP và SMTP đều có thể sử dụng cùng với hMailServer. RoundCube và SquirrelMail là những hệ thống Webmail phổ biến thường được sử dụng cùng với hMailServer.

hMailServer có thể sử dụng hệ cơ sở dữ liệu tích hợp sẵn là Microsoft SQL Server Compact Edition hoặc sử dụng bộ cơ sở dữ liệu bên ngoài như MySQL, Microsoft SQL hoặc PostgreSQL.

Các thư điện tử (email messages) được lưu trữ ở trên đĩa cứng theo định dạng MIME.

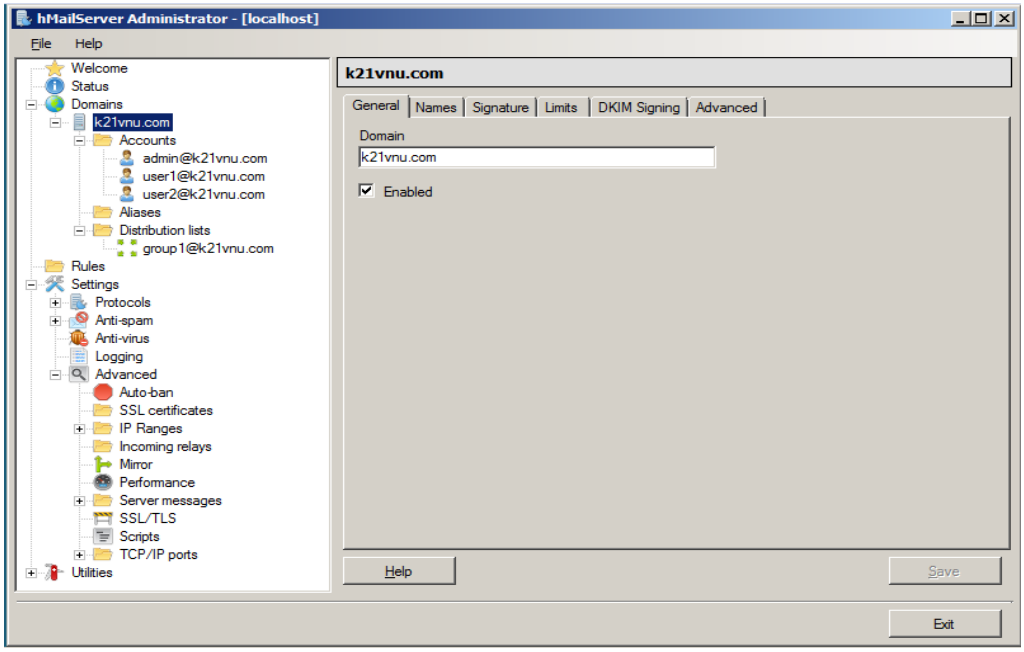
3.1.2. Các tính năng của hMailServer

hMailServer cung cấp đầy đủ tất cả các tính năng quan trọng cho một máy chủ email. Thêm vào đó, hMailServer luôn được cập nhật liên tục và những người dùng luôn nhận được sự hỗ trợ rất lớn từ những người khác hoặc những nhà phát triển khác trong cộng đồng hMailServer. [19]

3.1.2.1. Cài đặt và cấu hình đơn giản

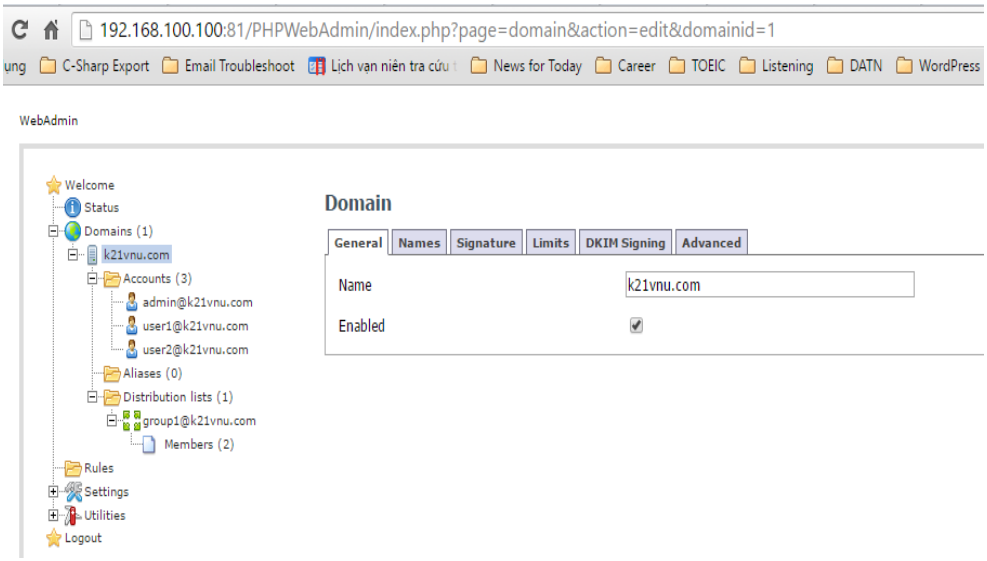
Việc cài đặt và cấu hình hMailServer là rất đơn giản. Các máy chủ được cài đặt sẽ đi kèm với một công cụ quản lý gọi là hMailServer Administrator. Công cụ này giúp

người quản trị có thể thêm các tên miền, tài khoản, chỉ định các thiết lập, kích hoạt chương trình quét virut và rất nhiều các thiết đặt khác.



Hình 3.1. Quản lý hMailServer bằng công cụ quản trị

Ngoài việc quản lý hMailServer bằng công cụ, hệ thống còn cho phép người quản trị có thể cấu hình bằng giao diện web, ở đó người quản trị có thể dùng để cấu hình tất cả các thành phần của hMailServer.



Hình 3.2. Quản lý hMailServer bằng giao diện web

3.1.2.2. Khả năng bảo mật cao

hMailServer được cấu hình sẵn để có chế độ an toàn cao khi thực hiện chuyển tiếp và xác thực các email. Điều này là đặc biệt quan trọng để không ai có thể sử dụng máy chủ để thực hiện gửi các tin nhắn rác. hMailServer cũng hỗ trợ các bộ mã nguồn mở ClamAV quét vi-rút rất nổi tiếng. Ngoài ra, hMailServer tích hợp sẵn các tính năng như là black-list, while-list, hỗ trợ các cơ chế chống thư rác như SPF và MX Lookups.

3.1.2.3. Khả năng tích hợp mở rộng

hMailServer đi kèm với một thư viện COM. Sử dụng thư viện COM, hMailServer có thể tích hợp các kịch bản (scripts) hoặc xây dựng các ứng dụng đầy đủ nhằm mục đích mở rộng tính năng cho hMailServer.

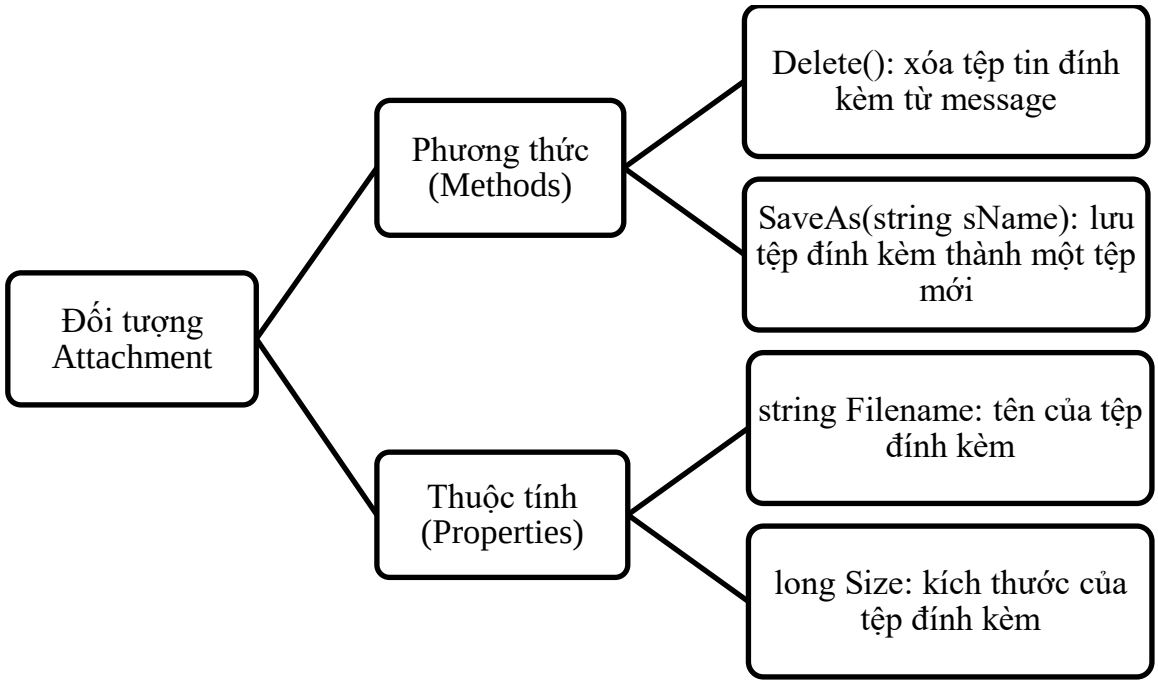
3.1.2.4. Các tính năng khác

hMailServer là giải pháp cung cấp đầy đủ các tính năng quan trọng cho một máy chủ email. Do đó, hMailServer bao gồm đầy đủ các tính năng của một máy chủ email thông thường. Tất cả các tính năng này đều được mô tả và hướng dẫn cấu hình khá chi tiết trên phần tài liệu trực tuyến của hMailServer tại liên kết:
https://www.hmailserver.com/docs

3.1.3. Thư viện COM và API sử dụng trong hMailServer

Thư viện COM và API cho phép người lập trình có thể viết các kịch bản và ứng dụng độc lập để tích hợp với hMailServer. Hầu như tất cả các đối tượng trong hMailServer đều có thể truy xuất được bằng cách sử dụng thư viện COM. Chúng ta có thể viết ứng dụng đầy đủ hoặc chỉ đơn giản viết một kịch bản để thực thi tự động khi có một sự kiện nào đó xảy ra, ví dụ như viết một kịch bản để lọc các thư đến trước khi nó được chuyển đến hộp thư của người nhận. [20]

hMailServer được thiết kế theo hướng đối tượng, tức là có nhiều đối tượng trong một hệ thống hMailServer, mỗi đối tượng đều có các phương thức và thuộc tính riêng. Ví dụ như Hình 4.3, đối tượng Attachment là đối tượng đại diện cho một tệp đính kèm cụ thể trong một email, nó bao gồm có các phương thức Delete(), SaveAs() và các thuộc tính gồm Filename và Size.



Hình 3.3. Một ví dụ về các phương thức và thuộc tính của đối tượng Attachment

Trong các đối tượng của hmailServer, đối tượng Application là đối tượng gốc (root) trong mô hình COM của hMailServer. Sử dụng đối tượng này, có thể truy xuất đến tất cả các đối tượng và các thiết đặt bên trong hMailServer. Tuy nhiên, trước khi có thể truy cập bất kỳ thuộc tính và phương thức trên đối tượng Application, chúng ta phải gọi phương thức Application.Authenticate và cung cấp kèm theo thông tin tài khoản xác thực hợp lệ.

Thư viện API trong hMailServer hoạt động cũng giống như hầu hết các thư viện COM khác, đặc biệt nó có thể tạo ra các "Trigger" như là những kịch bản mà được thực thi khi có một hành động nào đó xảy ra.

Các bước cơ bản cần thực hiện khi người dùng muốn truy xuất đến API hoặc COM của hMailServer:

- Bước 1: Tạo ra một thể hiện của đối tượng Application trong hMailServer. Điều này phụ thuộc vào ngôn ngữ lập trình được sử dụng, chẳng hạn như trong VBScript, có thể thực hiện bằng cách sử dụng dòng lệnh: CreateObject("hMailServer.Application").
- Bước 2: Yêu cầu xác thực. Trước khi có thể sử dụng bất kỳ một phương thức khác trong API, có một yêu cầu là phải xác thực. Việc này được thực hiện bằng cách sử dụng phương thức Authenticate() trên đối tượng Application. Điều này cũng sẽ ngăn chặn các người dùng không được cấp phép truy cập đến hệ thống hMailServer.
- Bước 3: Gọi phương thức và các thuộc tính cần sử dụng.

Dưới đây là một vài ví dụ về kịch bản (Script) và các hàm thực thi tự động khi có một sự kiện nào đó xảy ra (Trigger) được viết bằng Visual Basic for Applications (VBA). [21-22]

Ví dụ 1: viết một kịch bản cho phép thay đổi mật khẩu tài khoản của người dùng, kịch bản này được viết bằng VBA như sau:

```
' khai báo và tạo ra một thể hiện của đối tượng Application
  Dim obApp
  Set obApp = CreateObject("hMailServer.Application")
' thực hiện xác thực để có quyền thay đổi và sử dụng tài nguyên hMailServer
  Call obApp.Authenticate("Administrator", "Enter_password")
' Khai báo domain mà chúng ta muốn cấu hình
  Dim obDomain
  Set obDomain = obApp.Domains.ItemByName("example.com")
' lựa chọn tài khoản thuộc domain mà chúng ta muốn thay đổi mật khẩu
  Dim obAccount
  Set obAccount = obDomain.Accounts.ItemByAddress("account@example.com")
```

```
' thay đổi mật khẩu của người dùng thành “123456”
  obAccount.Password = "123456"
  obAccount.Save
```

Để kịch bản này hoạt động, chỉ cần thực hiện theo các bước sau:

- Sao chép kịch bản trên vào một trình soạn thảo “text editor” và lưu thành tệp tin có phần mở rộng là “.vbs”, chẳng hạn “vidu1.vbs”
- Trong máy tính chạy hệ điều hành windows, click đúp chuột vào tệp tin vừa được lưu (như là vidu1.vbs) để thực thi và kết quả ngay lập tức được áp dụng.

Ví dụ 2: viết một hàm trigger thực hiện việc ghi dòng thông báo “Hello World” tới bản ghi log (Event log) khi một message là được hMailServer chấp nhận xử lý.

```
Sub OnAcceptMessage(oClient, oMessage)
  EventLog.Write("Hello World")
End Sub
```

Hàm này sẽ được thực thi khi có một sự kiện xảy ra, sự kiện xảy ra thông thường được tạo thông qua các luật (Rules) trong hMailServer.

3.1.4. Môi trường phát triển của hMailServer

hMailServer là máy chủ email miễn phí và là mã nguồn mở nên việc phát triển mở rộng các tính năng được hỗ trợ tối đa từ tác giả và cộng động người sử dụng. Để phát triển hMailServer, chúng ta có thể thực hiện theo hai hướng tiếp cận:

Phương án 1: viết các kịch bản (script, trigger) để mở rộng tính năng. Với phương án này, hMailServer hỗ trợ hai ngôn ngữ kịch bản là VBScript và JScript.

Phương án 2: viết các ứng dụng đầy đủ hoặc chỉnh sửa sourcecode của hMailServer được lưu trữ trên GitHub. Phương án này phức tạp hơn và đòi hỏi người phát triển phải hiểu rất sâu về các đối tượng và luồng tương tác bên trong của hMailServer. Để thực hiện theo cách này, hMailServer yêu cầu môi trường phát triển ứng dụng gồm có: [23]

- Visual Studio 2013 Update 3 (dùng để phát triển ứng dụng)
- Database: sử dụng MS SQL hoặc MySQL hoặc PostgreSQL
- InnoSetup (dùng để xây dựng các chương trình cài đặt)
- Các thư viện: hMailServer sử dụng các thư viện của bên thứ ba gồm có OpenSSL, Boost và ngôn ngữ lập trình đi kèm Perl (Perl ActiveState ActivePerl Community Edition)

Như vậy, tùy theo mục đích phát triển ứng dụng mà đội ngũ phát triển nên cân nhắc việc lựa chọn phương án phù hợp để xây dựng các ứng dụng đạt hiệu quả tốt nhất.

3.2. Xây dựng hệ thống Email với hMailServer

3.2.1. Giới thiệu các thành phần cài đặt và quản trị

hMailServer có thể cài đặt trên nhiều hệ điều hành khác nhau của Microsoft như Windows XP, Vista, 7, 8, 10 hoặc Windows Server 2003, 2008, 2012 ở tất cả các phiên bản. hMailServer hầu hết là tương thích với các phần mềm chạy trên Windows. Để ổn định và phù hợp với việc triển khai thực tế, chúng ta sẽ tiến hành cài đặt như sau:

- Sử dụng phiên bản mới nhất của hMailServer tại thời điểm thực hiện luận văn là 5.6.5 build 2367 để tiến hành cài đặt.

- Lựa chọn hệ điều hành Windows Server 2008 để cài đặt.

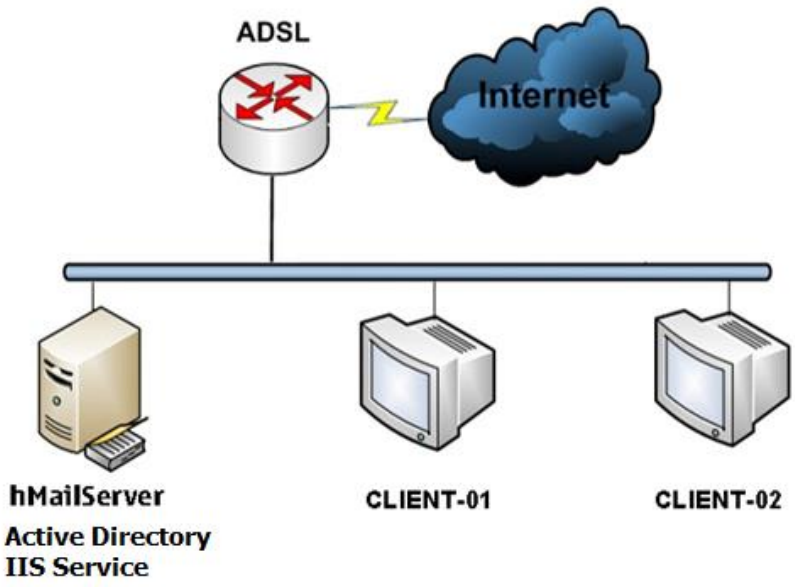
- Lựa chọn cơ sở dữ liệu là Microsoft SQL Server 2008 Express R2.

- Lựa chọn SquirrelMail làm Webmail.

- Lựa chọn chương trình mail client là Thunderbird.

- Lựa chọn bộ sản phẩm Xampp (tích hợp sẵn Apache, PHP, MySQL,…) để tạo máy chủ web nhằm mục đích tạo môi trường để chạy Webmail (sử dụng SquirelMail) và để chạy PHPWebAmin (công cụ quản trị máy chủ hMailServer qua web).

- Lựa chọn dịch vụ IIS (Internet Information Service) là một dịch vụ chạy trên nền hệ điều hành windows bao gồm nhiều dịch vụ khác nhau như web server, FTP server,.. được dùng với mục đích tạo ra đường link tham chiếu tới tệp tin đính kèm trong email sau khi tệp tin đính kèm được lưu trữ tại máy chủ.



Hình 3.4. Mô hình triển khai hệ thống hMailServer

Trên máy chủ Windows Server 2008, chúng ta sẽ tiến hành nâng cấp máy chủ lên thành Domain Controller (bằng cách cài đặt dịch vụ Active Directory của Microsoft). Máy chủ này sẽ vừa đóng vai trò máy chủ quản lý tập trung vừa đóng vai trò là máy chủ cài đặt phần mềm hMailServer. Các tài khoản email được tạo ra trên phần mềm hMailServer sẽ được ánh xạ đồng nhất với tài khoản người dùng trên Active Directory. Điều này sẽ giúp hệ thống quản lý được các tài khoản email dựa theo tài khoản của

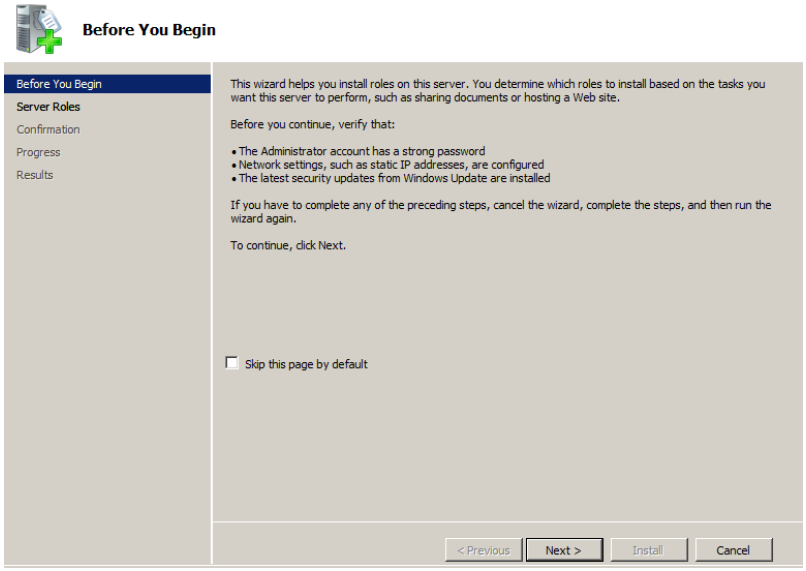
người dùng trong domain và giúp tăng cường tính bảo mật để xác thực người dùng mỗi khi người dùng truy cập vào đường link tham chiếu đến tệp tin đính kèm trong email.

3.2.2. Cài đặt máy chủ Active Directory và dịch vụ IIS

3.2.2.1. Cài đặt máy chủ Active Directory

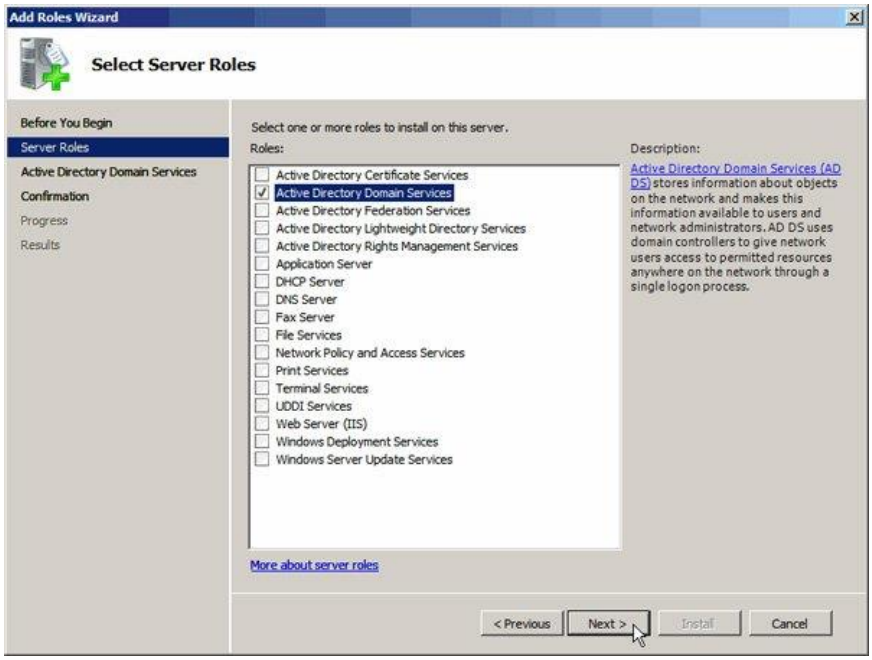
Trước khi cài đặt phần mềm hMailServer, chúng ta sẽ tiến hành cài đặt dịch vụ Active Directory trên máy chủ Windows Server 2008 nhằm quản lý tập trung các tài khoản người dùng để sử dụng trong cùng một hệ thống email.

Để cài đặt, trong máy chủ Windows Server 2008 mở Server Manager, chọn Roles và chọn mục Add Roles:



Hình 3.5. Trình thuật sĩ cài đặt Roles hiện lên khi click chọn Add Roles

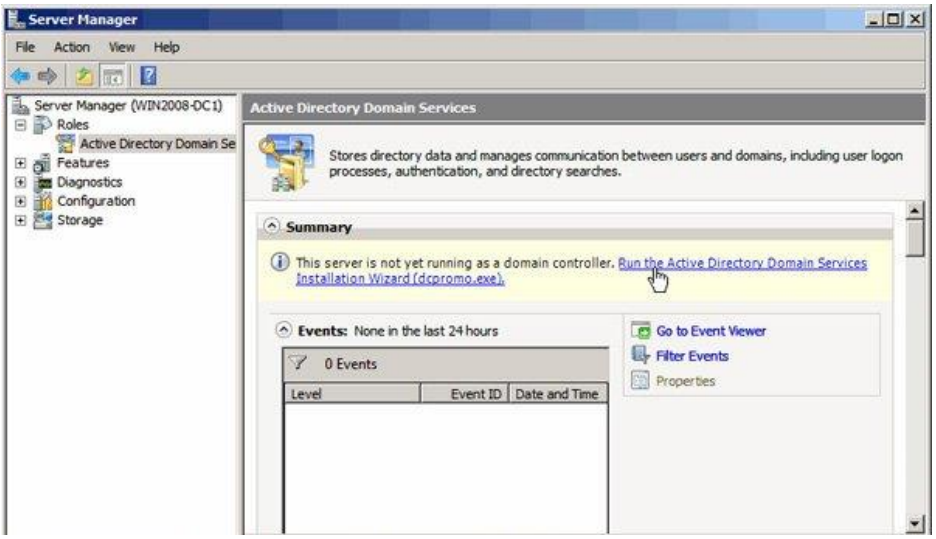
Click Next, trong cửa sổ Select Server Roles, Chọn Active Directory Domain Services nhấn Next để cài đặt:



Hình 3.6. Chọn dịch vụ Active Directory để cài đặt

Nhấn Next và chọn Install ở bước tiếp theo, hệ thống sẽ tiến hành cài đặt dịch vụ và sau cùng nhấn Close để hoàn tất.

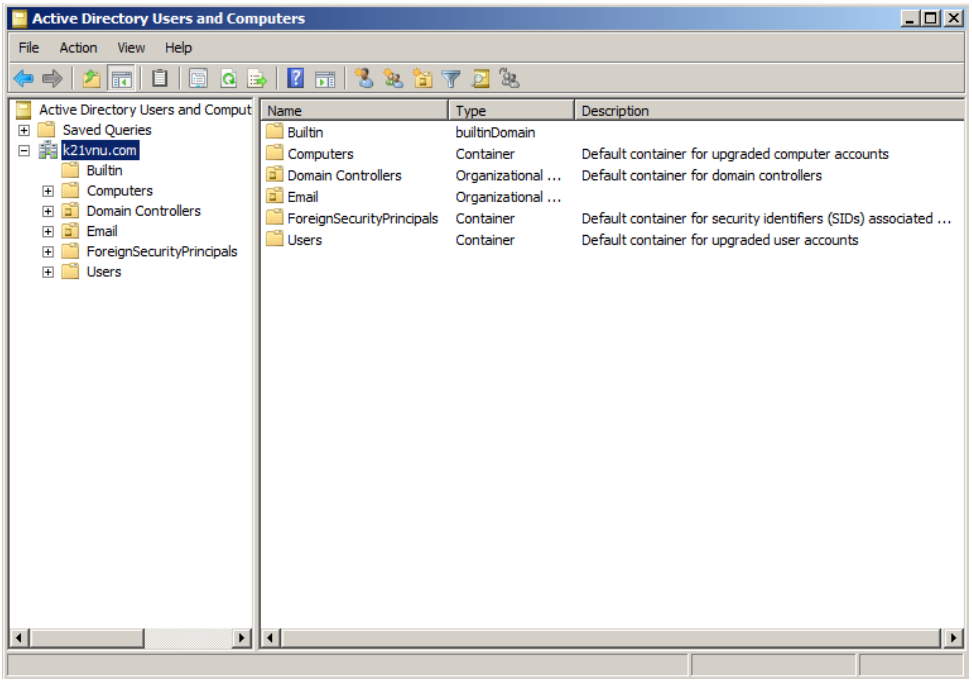
Sau khi cài đặt xong dịch vụ Active Directory, để sử dụng dịch vụ chúng ta cần phải kích hoạt bằng cách chạy lệnh DCPROMO từ Command Prompt hoặc kích hoạt bằng cách chạy click vào đường dẫn theo thông báo từ cửa sổ Server Manager:



Hình 3.7. Màn hình thông báo kích hoạt dịch vụ Active Directory

Lần lượt làm theo các hướng dẫn trong cửa sổ hiện ra để kích hoạt dịch vụ Active Directory. Quá trình này sẽ cho phép người quản trị khai báo các thông tin về domain, cầu hình DNS và một số các thông số khác cho hệ thống.

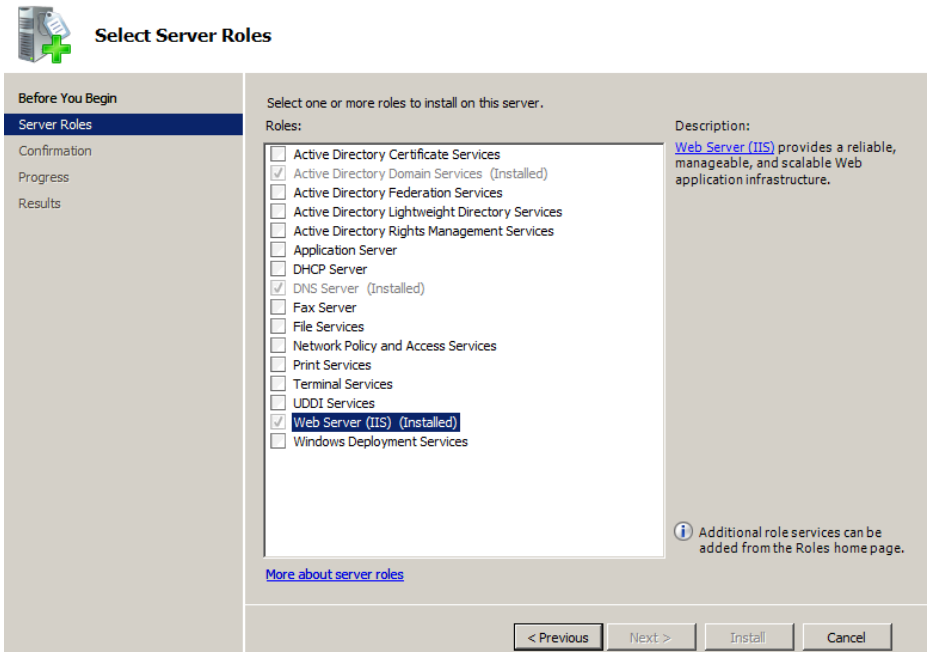
Sau khi quá trình hoàn tất, máy chủ Windows Server 2008 đã có thêm vai trò của Domain Controller, chúng ta có thể sử dụng một trong số các công cụ quản lý là Active Directory Users and Computers để quản lý danh sách các tài khoản người trong trong cùng Domain:



Hình 3.8. Công cụ quản lý Active Directory Users and Computers

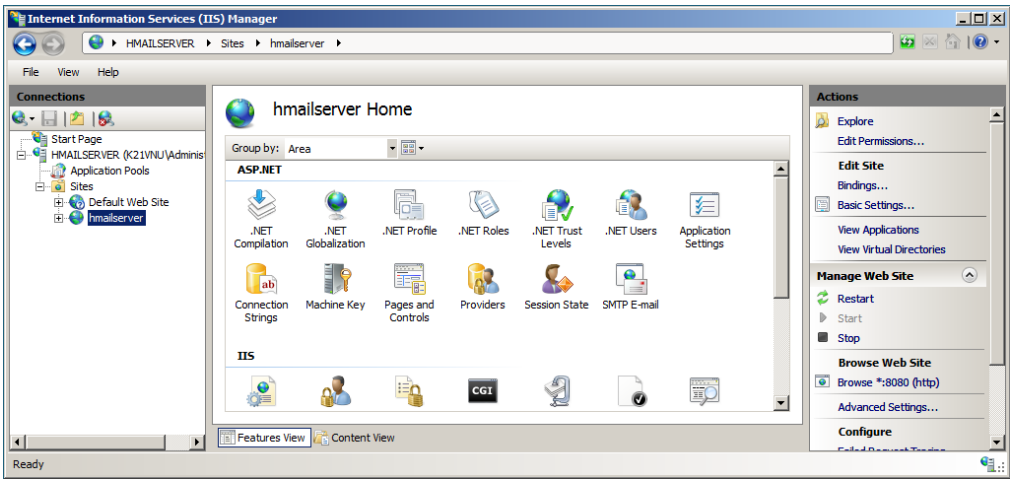
3.2.2.2. Cài đặt dịch vụ IIS

Tiếp theo đó, chúng ta tiến hành cài đặt dịch vụ IIS dùng làm web server để tạo đường link tham chiếu thay thế cho tệp tin đính kèm trong email. Để cài đặt dịch vụ IIS, chọn Server Manager, chọn Roles sau đó click chọn Add Roles. Tại cửa sổ Select Server Roles, đánh dấu chọn mục Web Server (IIS) để tiến hành cài đặt:



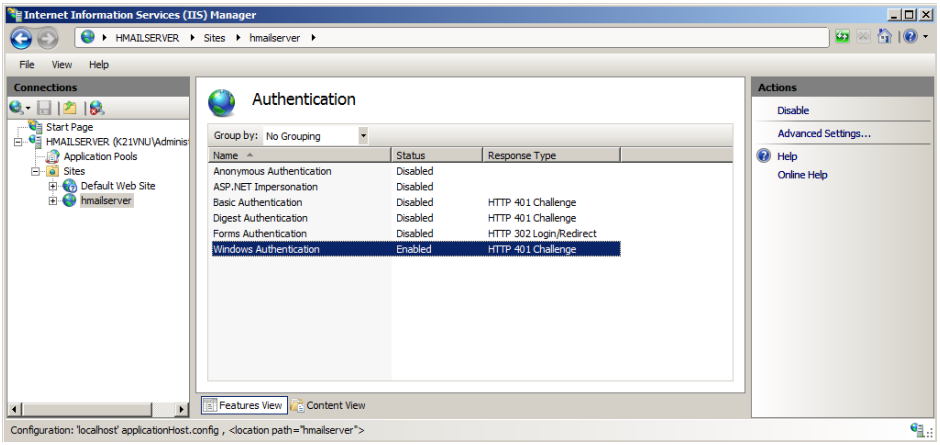
Hình 3.9. Lựa chọn dịch vụ Web Server (IIS) để cài đặt

Sau khi cài đặt xong dịch vụ, chúng ta tiến hành thêm mới một website để xác định thư mục lưu trữ các tệp tin đính kèm và để tạo đường link tham chiếu thay thế cho tệp tin đính kèm trong email gửi đi.



Hình 3.10. Thêm mới website để lưu trữ và tạo link cho các tệp đính kèm

Để yêu cầu xác thực khi truy cập đường link tham chiếu tới tệp tin đính kèm trong email, chúng ta cần kích hoạt dịch vụ xác thực Windows Authentication trong cấu hình của trang web vừa tạo trên dịch vụ IIS:



Hình 3.11. Cấu hình yêu cầu xác thực bằng tài khoản windows trên IIS

3.2.3. Cài đặt và Cấu hình hệ thống hMailServer

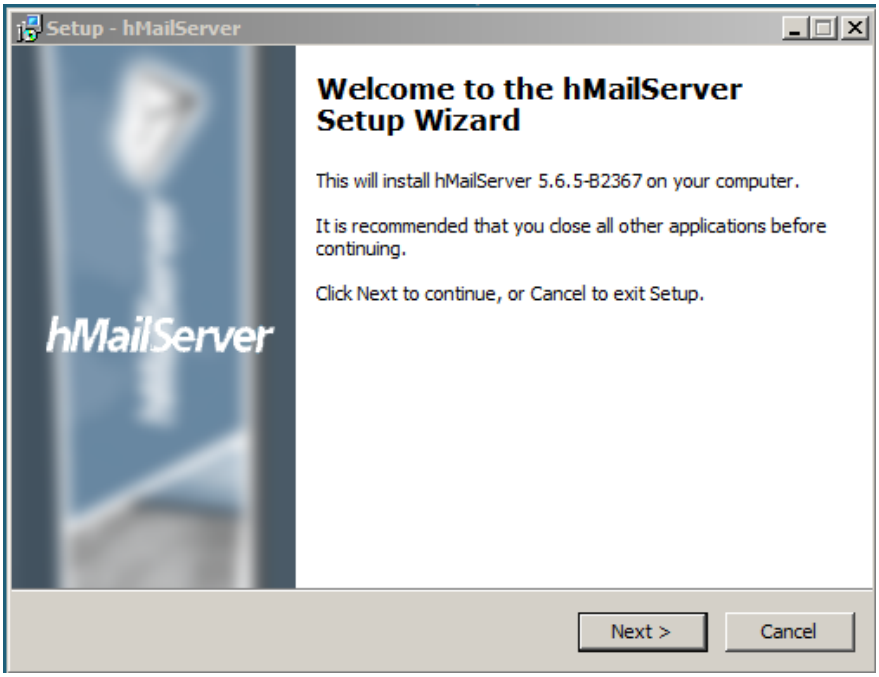
3.2.3.1. Cài đặt máy chủ hMailServer

Tiến hành tải về chương trình cài đặt của hMailServer tại địa chỉ: https://www.hmailserver.com/download . Version: hMailServer 5.6.5 – build 2367.

Lựa chọn máy chủ Windows Server 2008 để cài đặt hMailServer. Máy chủ này được cấu hình trước gồm 2 card mạng như sau:

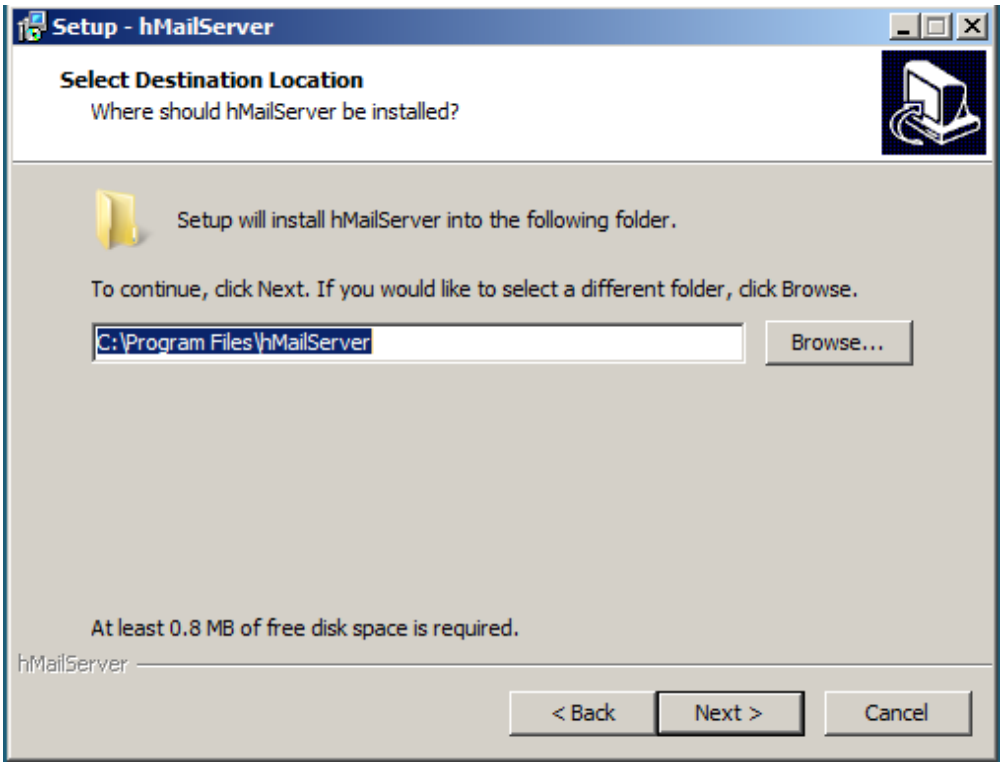
- Card mạng thứ nhất: có địa chỉ IP là 192.168.100.100/24 là card mạng đại diện cho mạng LAN để máy chủ hMailServer kết nối với các máy tính trong mạng LAN có cùng lớp địa chỉ mạng 192.168.100.0/24.
- Card mạng thứ hai: có địa chỉ IP để động là card mạng đại diện cho lớp mạng bên ngoài để máy chủ hMailServer kết nối được ra internet.

Sau khi tải về, chúng ta tiến hành cài đặt bằng cách chạy file "hMailServer-5.6.5-B2367.exe" và lần lượt thực hiện theo các bước như sau:



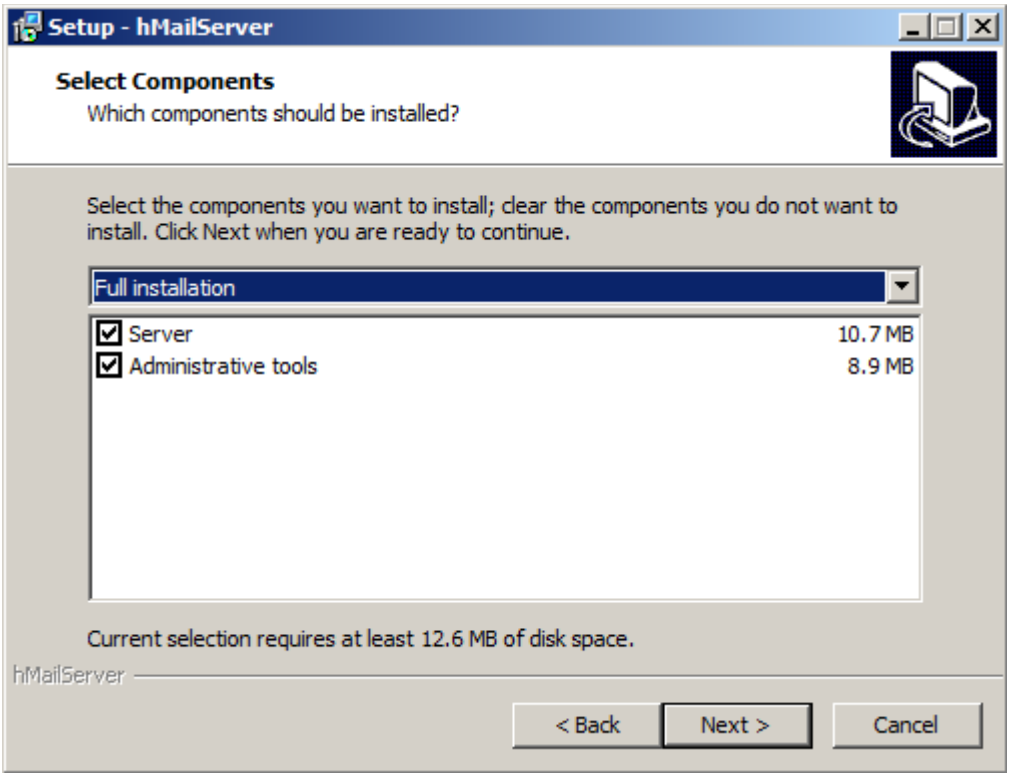
Hình 3.12. Bắt đầu tiền hành cài đặt hMailServer

Chọn Next, chọn đường dẫn cài đặt hMailServer:



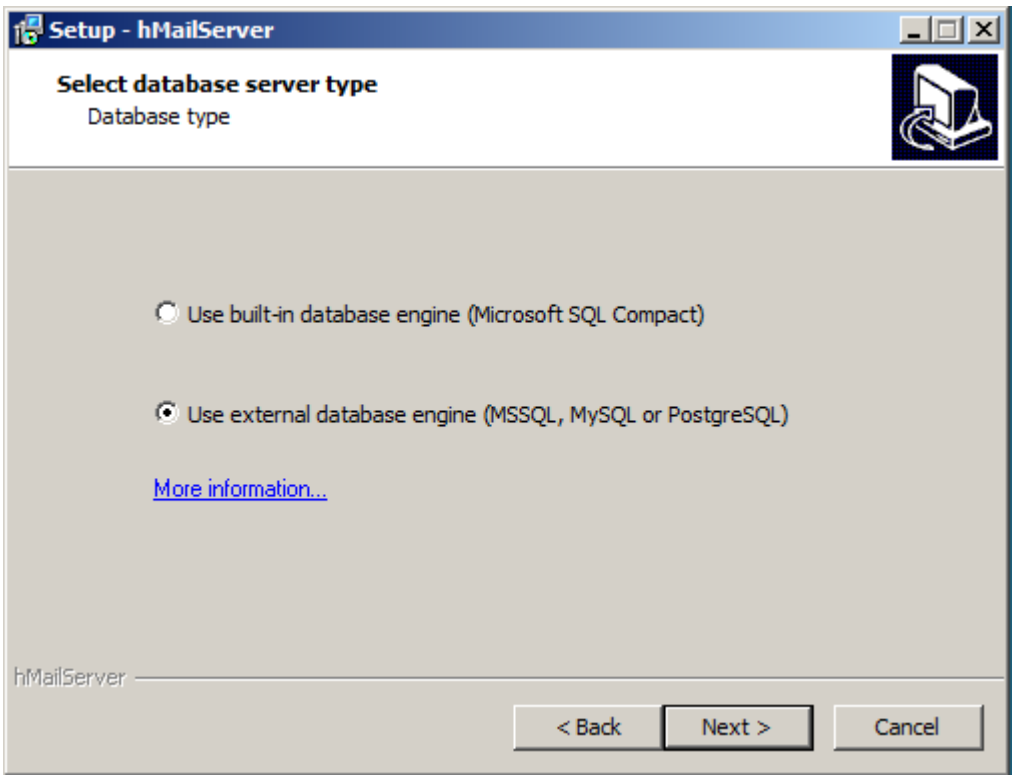
Hình 3.13. Chọn đường dẫn cài đặt hMailServer

Chọn Next, lựa chọn các thành phần cài đặt gồm chương trình và công cụ quản trị hMailServer:



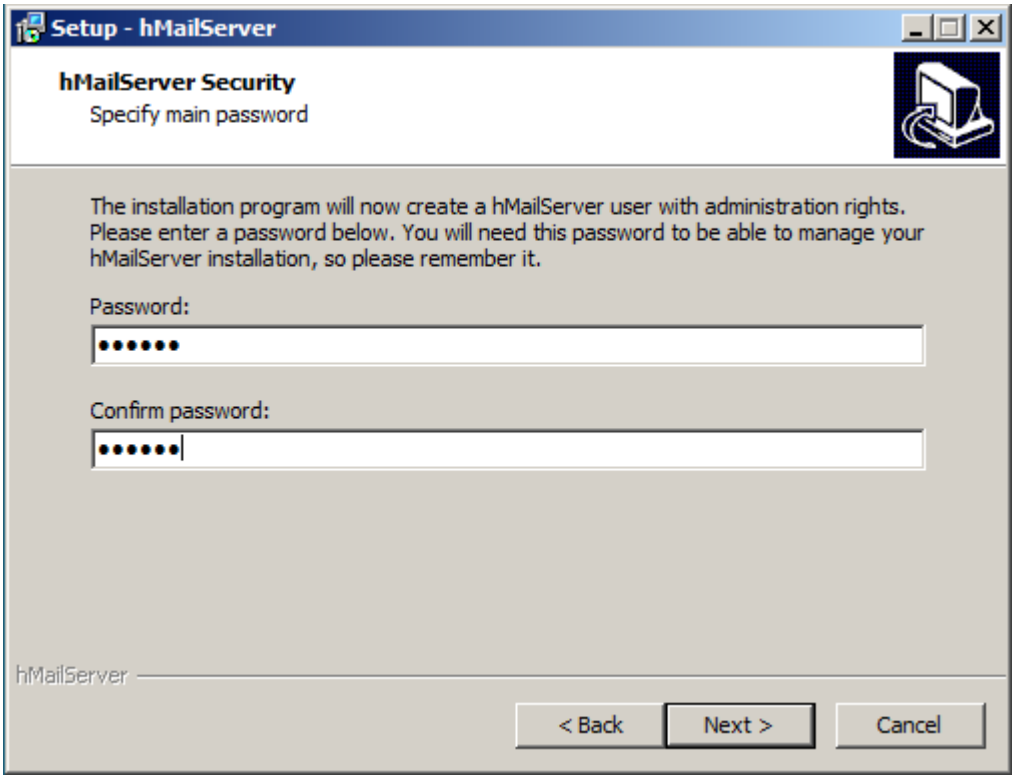
Hình 3.14. Chọn các thành phần để cài đặt cho hMailServer

Chọn Next, lựa chọn sử dụng hệ quản trị cơ sở dữ liệu Microsoft SQL Server để lưu trữ dữ liệu của hMailServer.



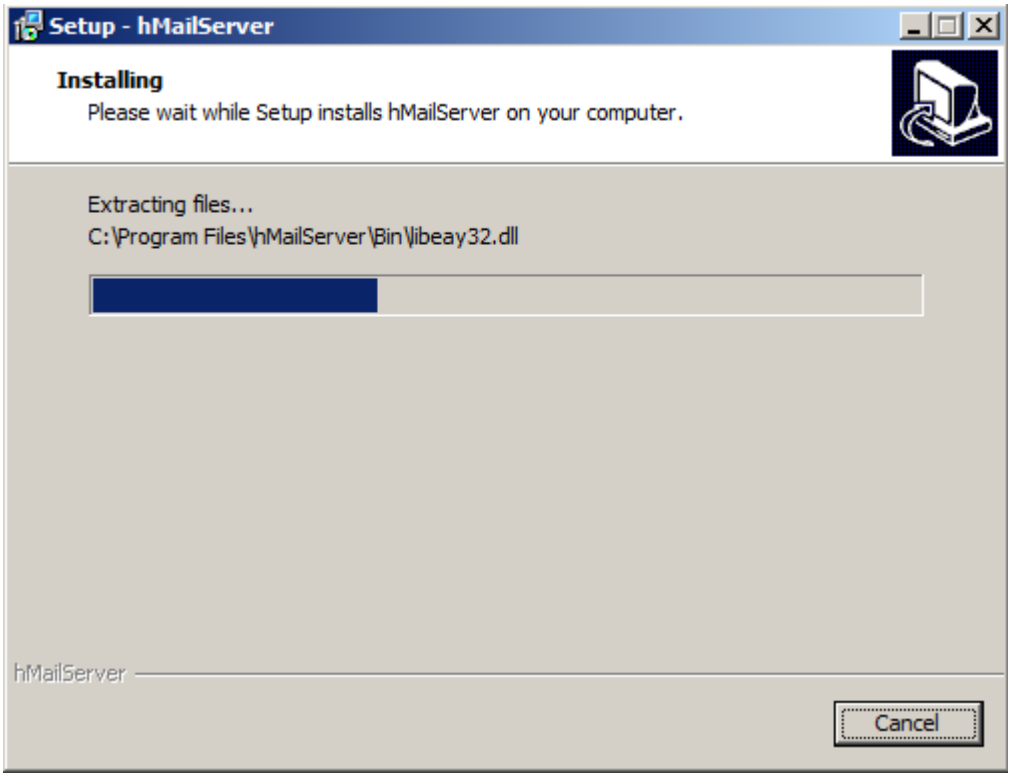
Hình 3.15. Tùy chọn cơ sở dữ diệu để sử dụng cho hMailServer

Chọn Next, tạo ra mật khẩu để quản trị hMailServer.



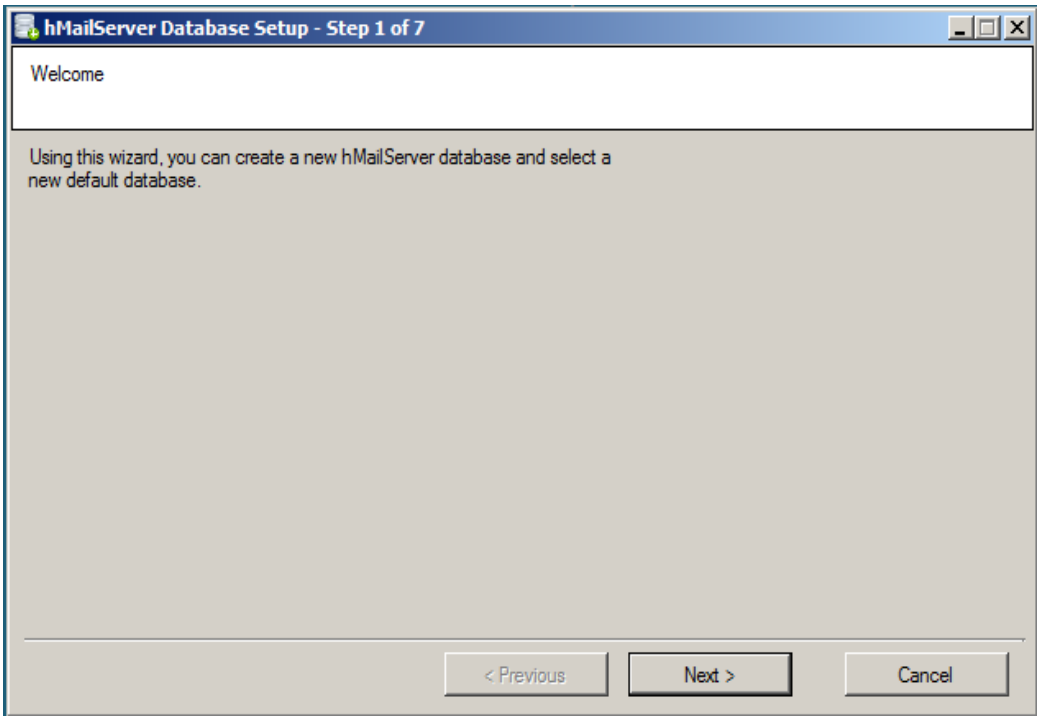
Hình 3.16. Tạo ra mật khẩu để quản trị hMailServer

Chọn Next, chương trình cài đặt được tiến hành.



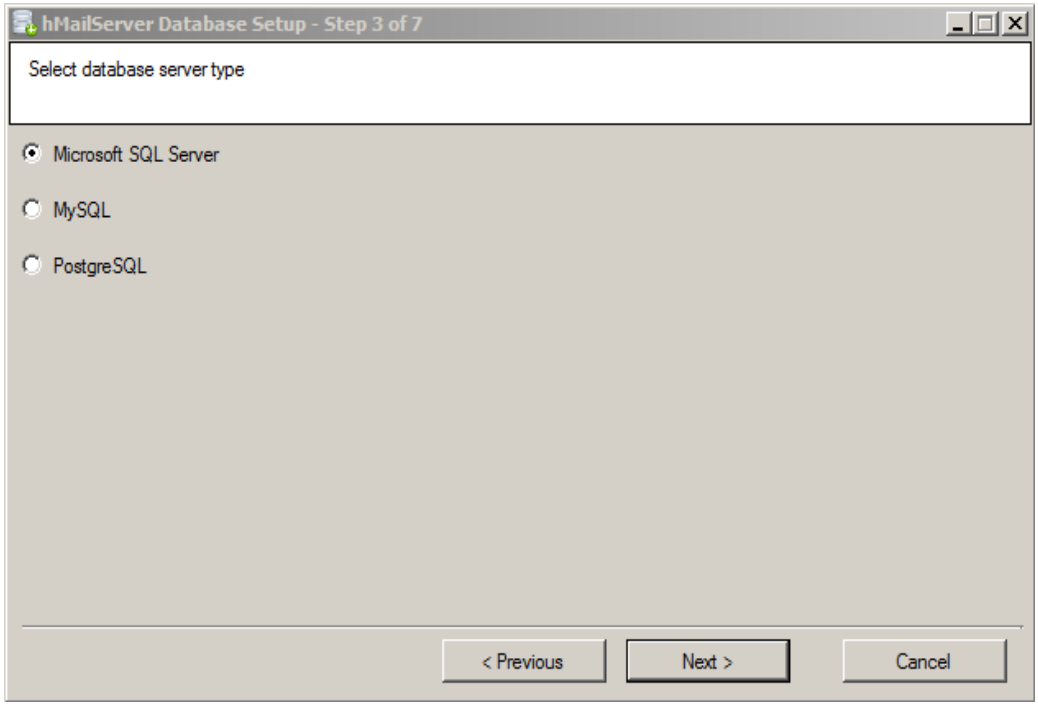
Hình 3.17. Quá trình cài đặt hMailServer được diễn ra

Sau khi cài đặt hoàn tất, tiến hành cấu hình cơ sở dữ liệu cho hMailServer bằng cách sử dụng công cụ được tích hợp sẵn trong quá trình cài đặt.



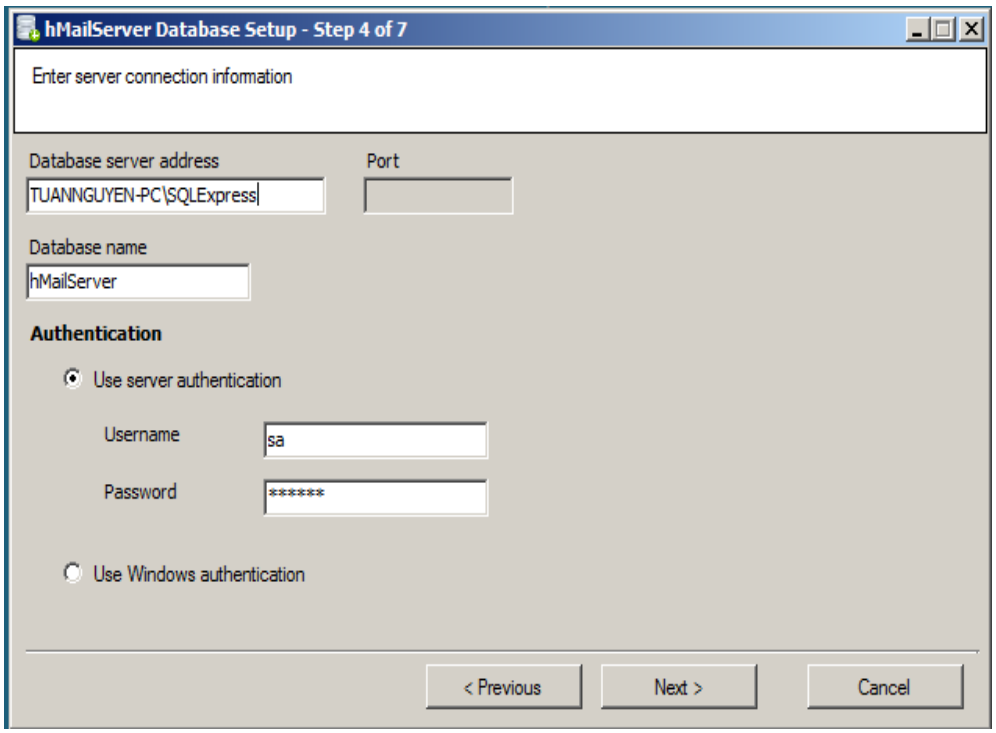
Hình 3.18. Cấu hình kết nối cơ sở dữ liệu cho hMailServer

Lựa chọn cơ sở dữ liệu sử dụng cùng hMailServer.



Hình 3.19. Cấu hình kết nối cơ sở dữ liệu cho hMailServer

Cấu hình các tham số để tạo cơ sở dữ liệu cho hMailServer



Hình 3.20. Cấu hình tham số để tạo cơ sở dữ liệu cho hMailServer

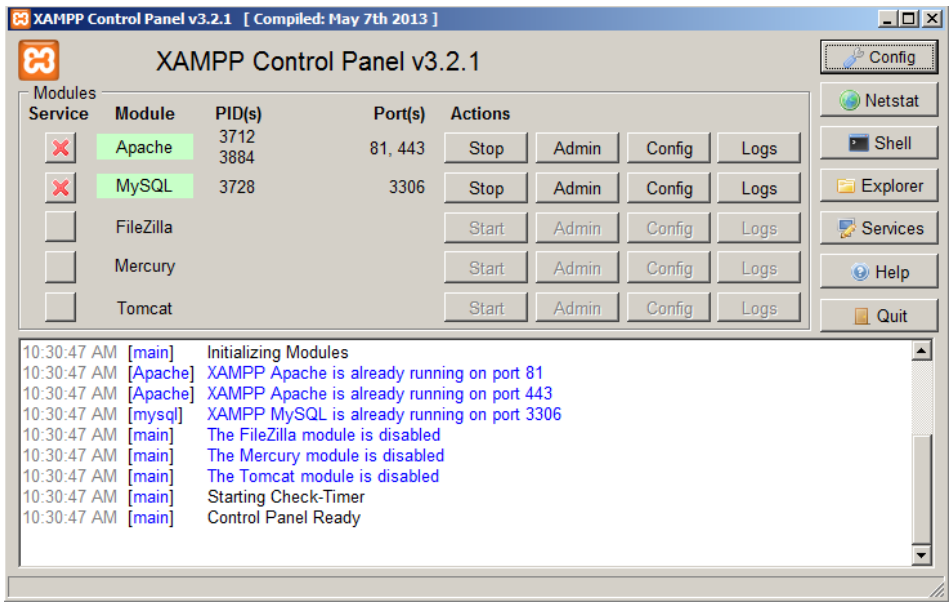
Chọn Next và Finish để hoàn tất quá trình cài đặt.

3.2.3.2. Cài đặt bộ quản trị WebAdmin và WebMail

Trong khi WebAdmin là công cụ quản trị máy chủ hMailServer thường được người quản trị sử dụng thì WebMail là công cụ truy cập email thường được sử dụng bởi người dùng, cả hai công cụ này đều được chạy thông qua môi trường web. Để cài đặt,

chúng ta cần phải có máy chủ web. Để đơn giản, chúng ta sẽ sử dụng Xampp để tạo máy chủ web.

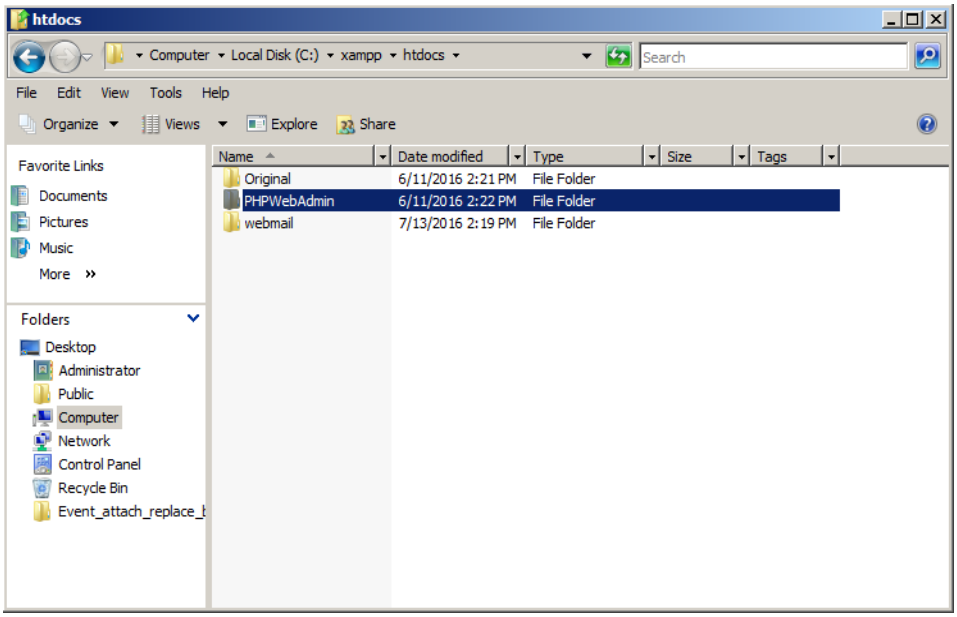
Lựa chọn phiên bản Xampp V3.2.1 để tiến hành cài đặt [24]. Sau khi cài đặt thành công, khởi chạy dịch vụ web apache:



Hình 3.21. Khởi chạy dịch vụ Apache trên Xampp v3.2.1

Để cài đặt bộ quản trị WebAdmin, chúng ta tiến hành theo các bước sau:

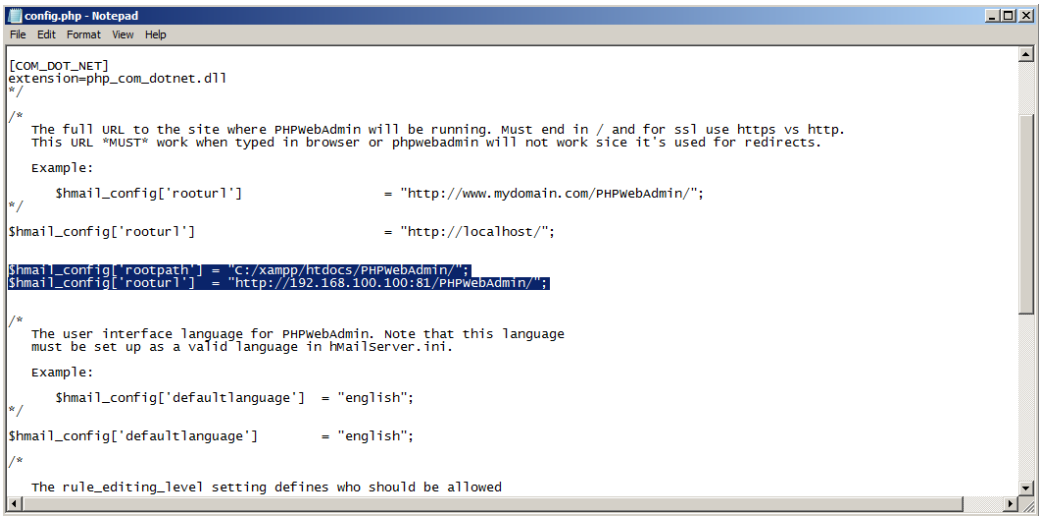
- Bước 1: sao chép thư mục PHPWebAdmin trong thư mục cài đặt hMailServer vào trong thư mục C:\xampp\htdocs (thư mục lưu trữ website của Xampp)



Hình 3.22. Cài đặt WebAdmin – sao chép thư mục PHPWebAdmin

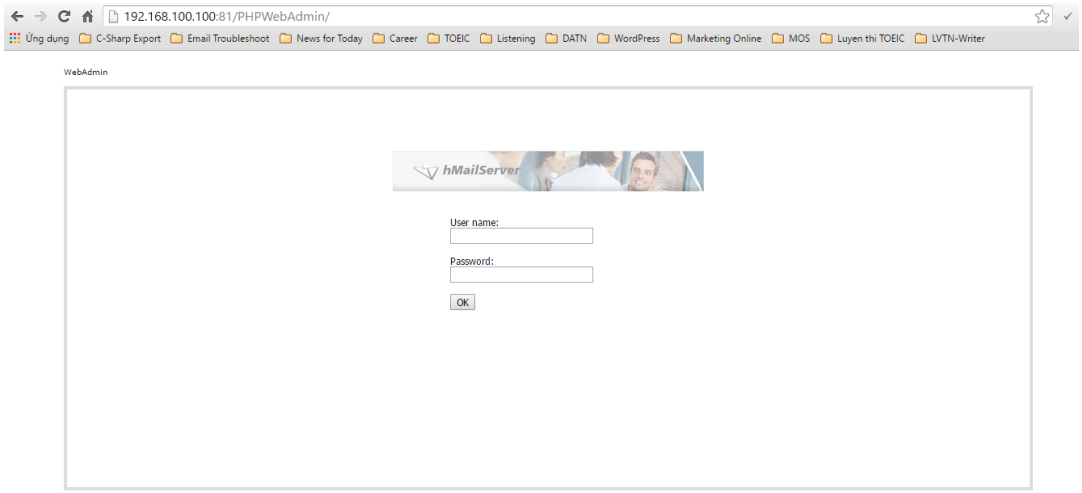
- Bước 2: Đổi tên tệp tin config-dist.php trong thư mục PHPWebAdmin vừa sao chép ở bước 1 thành config.php, sau đó thêm hai dòng lệnh sau:

```
$hmail_config['rootpath'] = "C:/xampp/htdocs/PHPWebAdmin/";
$hmail_config['rooturl'] = "http://192.168.100.100:81/PHPWebAdmin/";
```



Hình 3.23. Cài đặt WebAdmin – chỉnh sửa file config.php

- Bước 3: Sử dụng WebAdmin bằng cách mở trình duyệt web và truy cập theo địa chỉ: http://192.168.100.100:81/PHPWebAdmin/



Hình 3.24. Cài đặt WebAdmin – giao diện đăng nhập WebAdmin

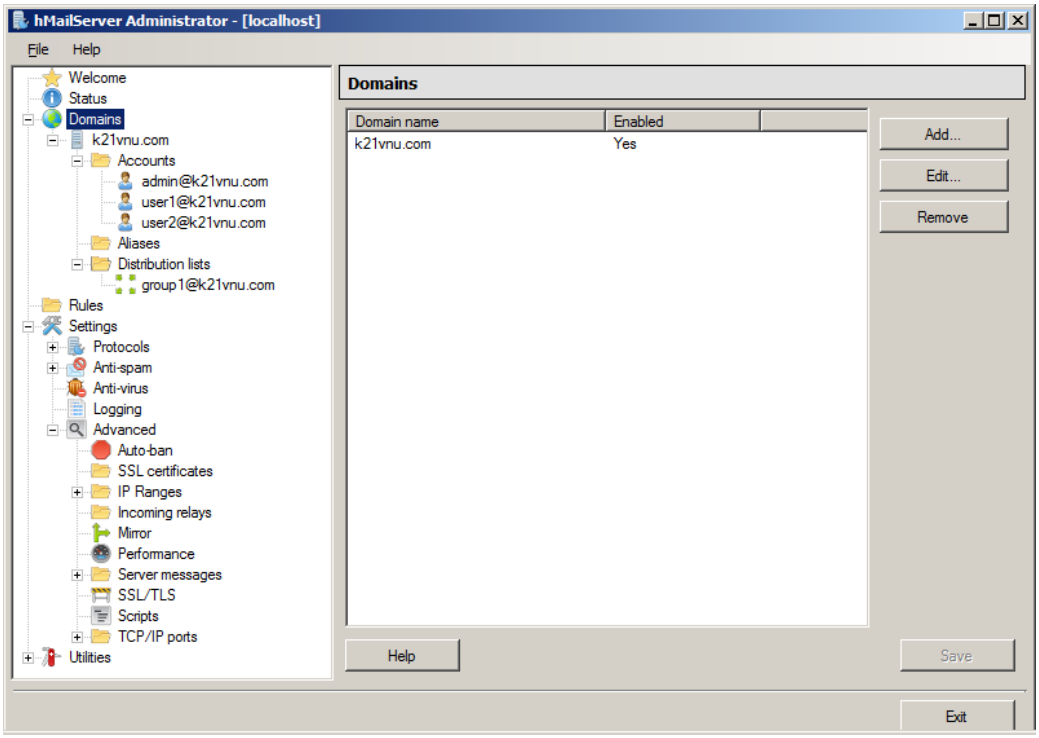
Để cài đặt chương trình WebMail sử dụng SquirrelMail, chúng ta tiến hành theo các bước sau:

- Bước 1: Tải về SquirrelMail tại địa chỉ: https://squirrelmail.org/download.php. Chúng ta sử dụng phiên bản mới nhất là squirrelmail-webmail-1.4.22.zip. [25]
- Bước 2: Giải nén và sao chép toàn bộ gói SquirrelMail vào trong thư mục đặt tên là webmail nằm trong thư mục: C:\xampp\htdocs\webmail.
- Bước 3: Trong thư mục config của thư mục webmail, đổi tên tệp tin config_default.php thành config.php, sau đó mở tệp tin này và thực hiện chỉnh sửa một số nội dung như tên tổ chức, logo, tên domain cho phù hợp với tổ chức sử dụng.
- Bước 4: Sử dụng webmail bằng cách mở trình duyệt web và truy cập theo địa chỉ: http://192.168.100.100:81/webmail

Hình 3.25. Cài đặt WebMail – giao diện đăng nhập WebMail

3.2.3.3. Cấu hình tên miền và tài khoản người dùng

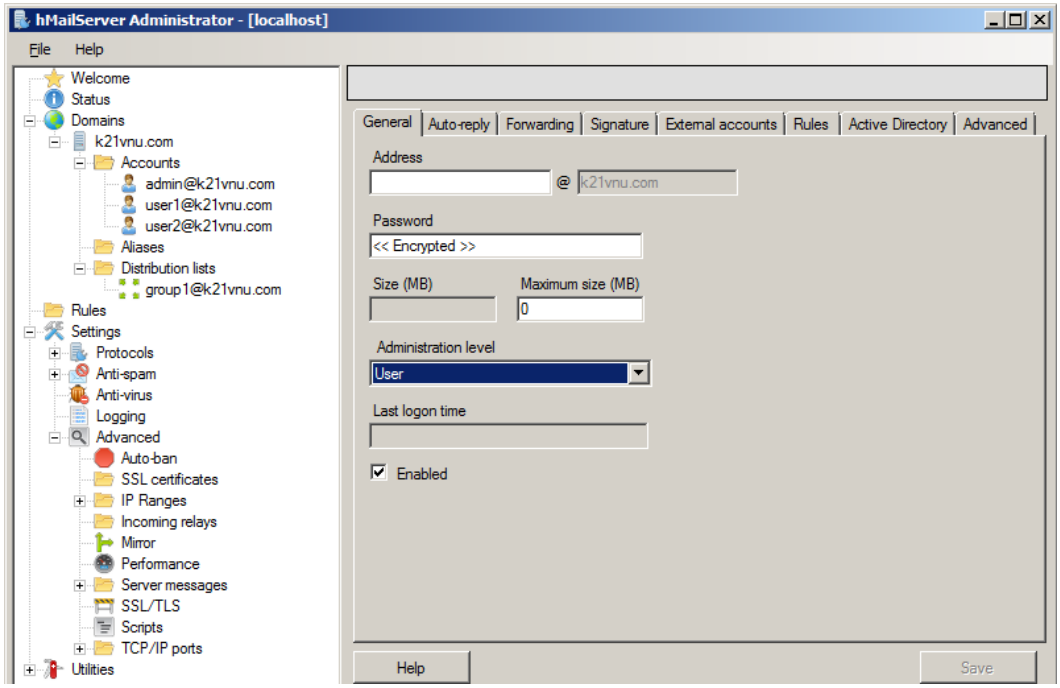
Để sử dụng hMailServer, chúng ta cần khai báo tên miền (Domain) sử dụng. Chúng ta có thể khai báo thông qua WebAdmin hoặc thông qua công cụ quản trị được cài đặt trên máy chủ hMailServer. [26]



Hình 3.26. Tạo Domain sử dụng trong hMailServer

Sau khi có Domain, chúng ta tạo các tài khoản người dùng tương ứng với Domain đã được tạo. Có ba cấp độ tài khoản trong hMailServer:

- Administration level = User: chức năng của level này áp dụng cho người dùng
- Administration level = Domain: chức năng của level này cung cấp các quyền cho người quản trị trên Domain đó
- Administration level = Server: chức năng của level này cung cấp các quyền cho người quản trị trên tất cả các Domain.



Hình 3.27. Giao diện tạo tài khoản người dùng trong hMailServer

Khi tạo người dung, chúng ta lựa chọn tab Active Directory (giao hiện như hình 4.27) để thiết lập tên miền (Domain) và tài khoản người dùng thuộc miền đó để gắn kết với tài khoản email được tạo. hMailServer khi đó sẽ ánh xạ một tài khoản email với một tài khoản người dùng trong Active Directory và sẽ sử dụng mật khẩu của tài khoản người dùng trong Active Directory làm mật khẩu cho tài khoản email.

3.2.3.4. Hoạt động gửi / nhận email trong hMailServer

Thư được gửi trong hệ thống hMailServer sẽ được lưu tại hòm thư của người nhận bên trong thư mục cài đặt hMailServer trên máy chủ.

Ví dụ: với các thông tin cài đặt hMailServer như ở phần trước thì email gửi đến cho user1 sẽ được lưu tại: C:\Program Files\hMailServer\Data\k21vnu.com\user1 (trong đó k21vnu.com là tên Domain)

Ngoài ra, các thông tin khác liên quan đến mỗi email giao dịch trong hMailServer sẽ được lưu trữ chi tiết trong cơ sở dữ liệu. Như trường hợp trên, mỗi email lưu trong hòm thư của user1 sẽ có các thông tin chi tiết kèm theo như: người gửi, ngày gửi, kích thước,… được lưu trữ trong cơ sở dữ liệu. Chúng ta có thể thực hiện các truy vấn (query) để xem danh sách các email được lưu trong hòm thư của người dùng:

SQLQuery1.sql - 19...ilServer (sa (53))*

```
USE [hMailServer] -- or whatever database name you use for hMailServer
GO

SELECT
    [messagecreatetime],
    [messagefilename],
    [messagefrom],
    [messagesize]
FROM
    [dbo].[hm_messages] m INNER JOIN
    [dbo].[hm_accounts] a ON a.[accountid] = m.[messageaccountid]
WHERE
    (a.accountaddress = 'user2@k21vnu.com')
ORDER BY messagecreatetime
```

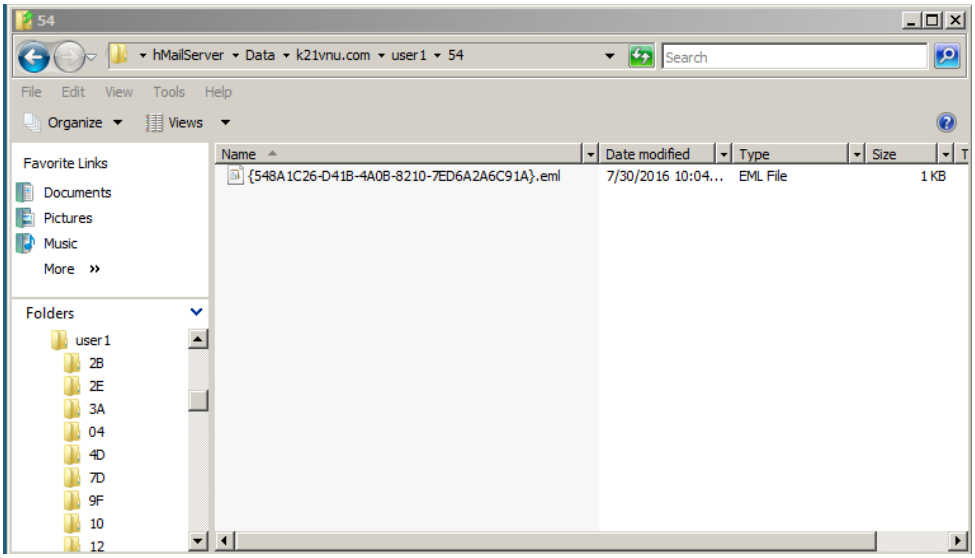
Results | Messages

	messagecreatetime	messagefilename	messagefrom	messagesize
1	2016-06-20 15:34:21.000	{955CA5A9-160E-4153-8593-80DEB0863ED8}.eml	admin@k21vnu.com	915947
2	2016-07-07 15:56:16.000	{9C66487D-974D-40BC-B8E2-F5248DE65B10}.eml	user1@k21vnu.com	1682822
3	2016-07-07 16:26:05.000	{9B14C191-5AAE-45E0-A3C8-FB870C183DB6}.eml	user1@k21vnu.com	1682797
4	2016-07-07 16:38:07.000	{EE816F88-1F51-42E8-B27D-C4BE1CB3B1BA}.eml		1131996
5	2016-07-13 14:54:44.000	{1AF2314E-C83F-433B-B298-721F4687E64B}.eml	user1@k21vnu.com	1600
6	2016-07-13 15:06:56.000	{6F52E5D1-25DA-40CC-A6B1-78CB7B703C1D}.eml	user1@k21vnu.com	1648
7	2016-07-13 15:26:18.000	{CC37B474-BEC6-40BE-9221-1C4F36048219}.eml	user1@k21vnu.com	1606
8	2016-07-13 15:36:57.000	{401EF8D4-D32F-4877-9D72-62E9423CF578}.eml	user1@k21vnu.com	1522
9	2016-07-14 14:52:37.000	{7056CEBD-7405-4EA1-B0A3-10F126C8DF30}.eml	user1@k21vnu.com	1896
10	2016-07-14 15:02:14.000	{162831D5-8F34-4E68-9144-77640A85CD39}.eml	user1@k21vnu.com	1923
11	2016-07-14 15:13:07.000	{FC656950-3A12-4EAC-8C76-E2055A391F20}.eml	user1@k21vnu.com	1743

Hình 3.28. Sử dụng truy vấn SQL để xem danh sách các email của người dùng

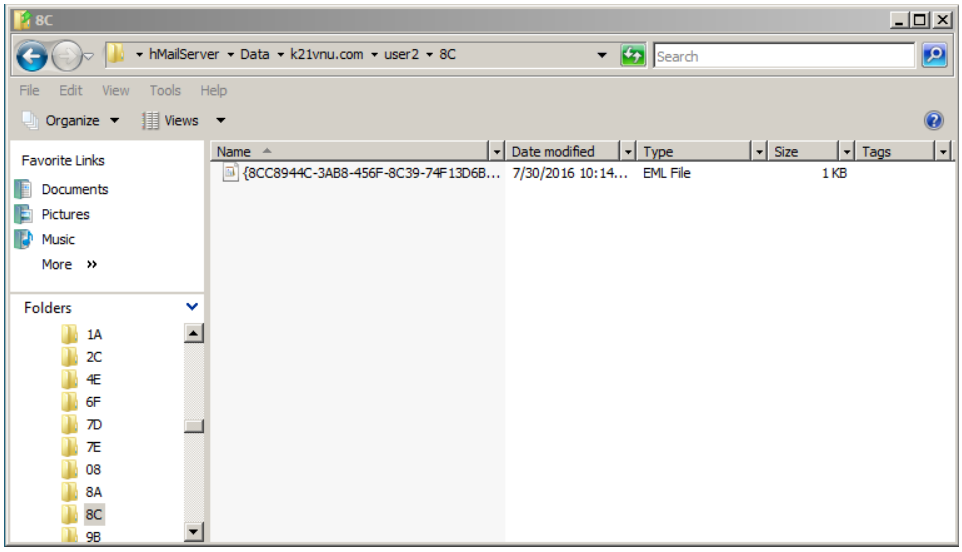
Chúng ta sẽ tiến hành kiểm tra hoạt động gửi / nhận email trong hMailServer bằng việc thực hiện quá trình gửi / nhận như sau:

- **Trường hợp 1:** Người dùng User1 sẽ gửi email cho Người dùng User2
  - Email gửi đi được lưu trong hòm thư User1



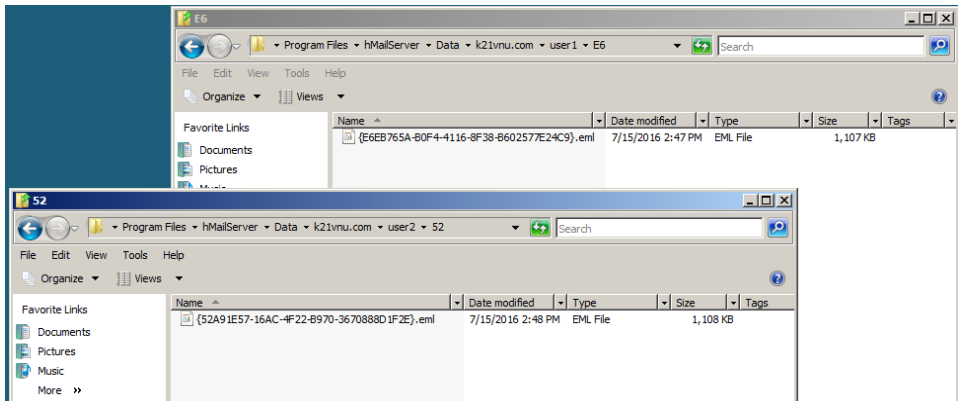
Hình 3.29. Email gửi đi được lưu trong hòm thư của User1

  - Email gửi đi được lưu trong hòm thư của User2



Hình 3.30. Email gửi đi được lưu trong hòm thư của User2

- **Trường hợp 2:** Người quản trị Admin sẽ gửi email cho Nhóm người dùng Group1 (trong hMailServer nhóm được gọi là distributions list – danh sách phân phối), Group1 gồm hai người dùng là User1 và User2. Email gửi đi sẽ được phân phát vào hòm thư của User1 và User2 với nội dung giống nhau:



Hình 3.31. Cùng một email gửi đi được lưu trong hòm thư của User1 và User2

3.2.4. Nhận xét về khả năng chống trùng lặp dữ liệu của hMailServer

Sau khi quan sát quá trình gửi / nhận email trong hMailServer có thể thấy rằng trường hợp email được gửi cho nhiều người nhận cùng lúc, nội dung email gửi đi giống nhau là được lưu riêng biệt tại mỗi hộp thư đến của người nhận. Điều này có nghĩa là các email giống nhau được lưu tại nhiều nơi khác nhau trong cùng hệ thống hMailServer.

Trong thực tế, khi sử dụng hMailServer một thời gian, quá trình gửi email cho nhiều nhóm người dùng hoặc nhiều email có nội dung giống nhau được gửi từ nhiều người khác nhau cho cùng một người nhận diễn ra hàng ngày có thể làm đĩa cứng trên máy chủ hMailServer nhanh đầy và dẫn đến tốc độ truy xuất email chậm hơn.

Đây là một trong những hạn chế liên quan đến khả năng chống trùng lặp dữ liệu mà phiên bản hiện tại của hMailServer chưa được cung cấp tính năng này.

3.3. Tích hợp tính năng deduplication trong hMailServer

3.3.1. Xây dựng kịch bản triển khai

Nhằm mục đích tích hợp tính năng Data Deduplication trong hMailServer để giảm tải bộ nhớ lưu trữ trên máy chủ và tiết kiệm nhiều nguồn tài nguyên, chúng ta có thể xây dựng kịch bản để áp dụng cho hệ thống hMailServer như sau:

- Bước 1: Kiểm tra email gửi đi trong trường hợp được gửi cho một hoặc nhiều nhóm người dùng (gồm nhiều người nhận)
- Bước 2: Thực hiện tách tệp tin đính kèm trong email gửi đi (trong trường hợp email có chứa tệp tin đính kèm) – do tệp đính kèm (nếu có) thường là thành phần chiếm nhiều dung lượng nhất trong email gửi đi.
- Bước 3: Lưu tệp tin đính kèm vào một thư mục xác định trên máy chủ hMailServer.
- Bước 4: Tạo đường link đến tệp tin vừa được lưu trữ, chèn đường link này vào trong email để thay thế tệp tin đính kèm và cuối cùng gửi email đến địa chỉ của người nhận.

Bằng việc triển khai kịch bản trên, khi một email gửi đi có chứa tệp đính kèm được gửi cho nhiều người nhận thì tệp tin đính kèm sẽ được lưu trữ một lần duy nhất và sẽ được thay thế trong email gửi đi bằng một đường link đến tệp đính kèm đã lưu trữ. Khi đó, tệp đính kèm sẽ không phải lưu trữ tại mỗi hòm thư của người nhận và dung lượng đĩa cứng lưu trữ trên hMailServer sẽ giảm đi đáng kể.

3.3.2. Cài đặt kịch bản

Chúng ta sẽ viết kịch bản bằng ngôn ngữ VBScript để thực hiện theo các bước được mô tả trong mục 4.3.1. Dựa trên phân loại và các yếu tố kỹ thuật về chống trùng lặp dữ liệu như đã phân tích ở chương 2, phạm vi của kịch bản được áp dụng như sau:

- Kịch bản xử lý quá trình chống trùng lặp dữ liệu ở mức độ file-level (mức độ tệp tin): xử lý gồm hai mức độ là dựa trên thuộc tính của tệp tin và dựa trên hàm băm (dùng thuật toán MD5) để so sánh các chuỗi tệp tin (nếu tệp tin sau có chuỗi MD5 trùng với tệp tin lưu trữ trước đó thì hai tệp tin này là giống nhau, khi đó tệp tin kiểm tra sau sẽ không được lưu và đường link trong email gửi đi sẽ được tham chiếu đến tệp tin đính kèm đã lưu trước đó)
- Kịch bản áp dụng kỹ thuật chống trùng lặp dữ liệu ở mức độ Source-Based (chống trùng lặp dữ liệu tại nguồn): dữ liệu sẽ được xử lý trùng lặp trước khi được lưu vào máy chủ email của người nhận.

Trong quá trình thực hiện luận văn, tôi đã thực hiện viết hai scripts với nội dung chi tiết như sau:

*) Script thứ nhất: sẽ tiến hành tách tệp tin đính kèm trong email gửi đi, lưu trữ vào máy chủ email, gắn đường link tham chiếu đến tệp tin vừa lưu trữ vào trong email gửi đi và cuối cùng gửi email đến người nhận. Ưu điểm của Script này là giúp tăng cường hiệu năng của hệ thống khi không phải so sánh hàm băm (chuỗi MD5) giữa các tệp tin đính kèm được lưu trữ trên máy chủ email. Tuy nhiên, khi triển khai có nhược

điểm là hệ thống sẽ không kiểm tra được sự trùng lặp khi các email gửi đến sau có tệp tin đính kèm trùng với tệp tin đính kèm của email đã được gửi trước đó.

```
Function DetachAttachments(oMessage)
' xác định đường dẫn chứa file đính kèm và đường link để truy cập
  PathName = "C:\inetpub\wwwroot\hmailserver\attachfiles\"
  UrlName = "http://k21vnu.com:8080/attachfiles/"
  aText = "Email này chứa file đính kèm được gán ở link sau:"
' kiểm tra email gửi đi có chứa file đính kèm
  If oMessage.attachments.count > 0 Then
   max=1000000
   min=1
   Randomize
   dem=0
' duyệt lần lượt từng file đính kèm có trong email
   for i = 1 to oMessage.attachments.count ' attachment
 ' xử lý kiểu file
num_fileExt = (InStr(1,StrReverse(oMessage.Attachments(i-1).Filename),".")) - 1
fileExt = Right(LCase(oMessage.Attachments(i-1).Filename),num_fileExt)
    ' kiểm tra file đính kèm cần xử lý
        If (oMessage.attachments.item(i-1).size > 20) and (fileExt <> "eml") Then
            'xác định đường dẫn tạm thời chứa file đính kèm theo kiểu file
             newTempfolderpath = PathName & fileExt
             newTempUrlName = UrlName & fileExt
             'kiểm tra folder theo kiểu file đã có chưa, nếu chưa có thì tạo mới
             set filesys=CreateObject("Scripting.FileSystemObject")
             If Not filesys.FolderExists(newTempfolderpath) Then
                    Set newfolder = filesys.CreateFolder(newTempfolderpath)
             End If
             'lưu file vào máy chủ
             NewName = (Int((max-min+1)*Rnd+min))
             aPath = newTempfolderpath & "\" & NewName & "." & fileExt
             aUrl = newTempUrlName & "/" & NewName & "." & fileExt
             oMessage.attachments.item(i-1).saveAs(aPath)
             'đính kèm đường dẫn file đính kèm vào email và xóa file đính kèm
             oMessage.Body = aUrl & VBNewLine & VBNewLine &
oMessage.Body
             oMessage.HTMLBody = "<a href=" & Chr(34) & aUrl & Chr(34) &">"
& aUrl & "</a>" & "<br />" & "<br />" & oMessage.HTMLBody
             oMessage.attachments.item(i-1).delete()
             dem = 1
```

```
        End If
    Next
     'chèn text thông báo (aText) có file đính kèm
     If dem > 0 Then
      oMessage.Body = aText & VBNewLine & oMessage.Body
      oMessage.HTMLBody = aText & "<br />" & oMessage.HTMLBody
      oMessage.save
     End If
    End If
End Function
```

*) Script thứ hai: sẽ tiến hành tách tệp tin đính kèm trong email gửi đi, kiểm tra hàm băm (chuỗi MD5) giữa tệp tin đính kèm đang xử lý với các tệp tin đính kèm cùng kiểu đã được lưu trên hệ thống trước đó (nếu mã MD5 trùng nhau thì không lưu tệp tin đính kèm đang xử lý và ngược lại nếu mã MD5 không trùng thì lưu tệp tin đính kèm đang xử lý), gắn đường link tham chiếu đến tệp tin vừa lưu trữ vào trong email gửi đi và cuối cùng gửi email đến người nhận. Ưu điểm của Script này là giúp xử lý triệt để vấn đề chống trùng lặp dữ liệu giữa các tệp tin đính kèm được gửi đến máy chủ email tại các thời điểm khác nhau. Tuy nhiên, khi triển khai có nhược điểm sẽ tốn tài nguyên hệ thống do máy chủ email phải xử lý và so sánh chuỗi MD5 giữa các tệp tin đính kèm.

```
'trả về mã MD5 của một file bất kỳ
Private lngTrack
Private arrLongConversion(4)
Private arrSplit64(63)
Private Const OFFSET_4 = 4294967296
Private Const MAXINT_4 = 2147483647
Private Const S11 = 7
Private Const S12 = 12
Private Const S13 = 17
Private Const S14 = 22
Private Const S21 = 5
Private Const S22 = 9
Private Const S23 = 14
Private Const S24 = 20
Private Const S31 = 4
Private Const S32 = 11
Private Const S33 = 16
Private Const S34 = 23
Private Const S41 = 6
Private Const S42 = 10
Private Const S43 = 15
Private Const S44 = 21
'//=======================================================
Function DetachAttachments(oMessage)
'xác định đường dẫn chứa file đính kèm và truy cập từ web
```

```
PathName = "C:\inetpub\wwwroot\hmailserver\attachfiles\"
UrlName = "http://k21vnu.com:8080/attachfiles/"
aText = "Email nay chua file dinh kem theo cac link duoi day:"
'kiểm tra xem email gửi đi có file đính kèm không
If oMessage.attachments.count > 0 Then
  max=1000000
  min=1
  Randomize
'duyệt lần lượt từng file đính kèm có trong email
For i = 1 to oMessage.attachments.count
'xử lý kiểu file
num_fileExt = (InStr(1,StrReverse(oMessage.Attachments(i-1).Filename),".")) - 1
fileExt = Right(LCase(oMessage.Attachments(i-1).Filename),num_fileExt)
'kiểm tra file đính kèm cần xử lý
If (oMessage.attachments.item(i-1).size > 20) and (fileExt <> "eml") Then
'xác định đường dẫn tạm thời chưa file đính kèm
        newTempfolderpath = PathName & fileExt
        newTempUrlName = UrlName & fileExt
'lưu file, thực hiện hashing, sau đó xóa đi khi kết thúc so sánh
temp_current_attachfiles = "C:\xampp\htdocs\webmail\attachfiles\temp"
aTempPath = temp_current_attachfiles & "\" & "temp_file_" & i & "." & fileExt
oMessage.attachments.item(i-1).saveAs(aTempPath)
sHash = MD5FileHash(aTempPath) 'MD5 hashing current attach file
        dem = 0
'kiểm tra folder theo kiểu file đã có chưa, nếu chưa có thì tạo mới
set filesys=CreateObject("Scripting.FileSystemObject")
If Not filesys.FolderExists(newTempfolderpath) Then
        Set newfolder = filesys.CreateFolder(newTempfolderpath)
        NewName = (Int((max-min+1)*Rnd+min))
        aPath = newTempfolderpath & "\" & NewName & "." & fileExt
        aUrl = newTempUrlName & "/" & NewName & "." & fileExt
        oMessage.attachments.item(i-1).saveAs(aPath)
'nếu tồn tại folder theo kiểu file
        Else
'so sánh chuỗi MD5 giữa file đính kèm với các file đã lưu
Set AAA = CreateObject("Scripting.FileSystemObject")
Set BBB = AAA.GetFolder(newTempfolderpath).Files
        For Each CCC In BBB
        If (MD5FileHash(CCC) = sHash) Then
                dem = 1
                num_saved_fileName = (InStr(1,StrReverse(CCC),"\")) - 1
                savedfileName = Right(CCC,num_saved_fileName)
        Exit For
End If
Next
'kiểm tra kết quả so sánh
IF (dem = 0) Then
        New2Name = (Int((max-min+1)*Rnd+min))
```

```
        a2Path = newTempfolderpath & "\" & New2Name & "." & fileExt
        aUrl = newTempUrlName & "/" & New2Name & "." & fileExt
        oMessage.attachments.item(i-1).saveAs(a2Path)
Else
        aUrl = newTempUrlName & "/" & savedfileName
End If
End If
        oMessage.Body = aUrl & VBNewLine & VBNewLine & oMessage.Body
        oMessage.HTMLBody = "<a href=" & Chr(34) & aUrl & Chr(34) &">" & aUrl
& "</a>" & "<br />" & "<br />" & oMessage.HTMLBody
        oMessage.attachments.item(i-1).delete()
        counter = 1
End If
'xóa file đính kèm đã lưu tạm trước đó
Set fso = CreateObject("Scripting.FileSystemObject")
        fso.DeleteFile(aTempPath)
Next
'chèn text thông báo có file đính kèm
    If counter > 0 Then
     oMessage.Body = aText & VBNewLine & oMessage.Body
     oMessage.HTMLBody = aText & "<br />" & oMessage.HTMLBody
     oMessage.save
    End If
  End If
End Function

'//=========================================================
Public Function MD5FileHash(strFile)
        Dim strMD5 : strMD5 = ""
        Dim ofso : Set ofso = CreateObject("Scripting.FileSystemObject")
        If ofso.FileExists(strFile) then
                strMD5 = BinaryToString(ReadTextFile(strFile, ""))
                MD5FileHash = CalculateMD5(strMD5)
        Else
                MD5FileHash = strFile & VbCrLf & "Error: File not found"
        End if
End Function
' ------------------------------------
Function ReadTextFile(FileName, CharSet)
        Const adTypeText = 2
        Dim BinaryStream : Set BinaryStream = CreateObject("ADODB.Stream")
        BinaryStream.Type = adTypeText
        If Len(CharSet) > 0 Then
                BinaryStream.CharSet = CharSet
        End If
        BinaryStream.Open
        BinaryStream.LoadFromFile FileName
        ReadTextFile = BinaryStream.ReadText
```

```
End Function
' ----------------------------
Function BinaryToString(Binary)
Dim cl1, cl2, cl3, pl1, pl2, pl3
Dim L
	cl1 = 1
	cl2 = 1
	cl3 = 1
	L = LenB(Binary)
	Do While cl1<=L
		pl3 = pl3 & Chr(AscB(MidB(Binary,cl1,1)))
		cl1 = cl1 + 1
		cl3 = cl3 + 1
		If cl3>300 Then
			pl2 = pl2 & pl3
			pl3 = ""
			cl3 = 1
			cl2 = cl2 + 1
			If cl2>200 Then
				pl1 = pl1 & pl2
				pl2 = ""
				cl2 = 1
			End If
		End If
	Loop
	BinaryToString = pl1 & pl2 & pl3
End Function
' ------------------------------------------------------
Private Function MD5Round(strRound, a, b, C, d, X, S, ac)
	Select Case strRound
		Case "FF"
			a = MD5LongAdd4(a, (b And C) Or (Not (b) And d), X, ac)
			a = MD5Rotate(a, S)
			a = MD5LongAdd(a, b)
		Case "GG"
			a = MD5LongAdd4(a, (b And d) Or (C And Not (d)), X, ac)
			a = MD5Rotate(a, S)
			a = MD5LongAdd(a, b)
		Case "HH"
			a = MD5LongAdd4(a, b Xor C Xor d, X, ac)
			a = MD5Rotate(a, S)
			a = MD5LongAdd(a, b)
		Case "II"
			a = MD5LongAdd4(a, C Xor (b Or Not (d)), X, ac)
			a = MD5Rotate(a, S)
			a = MD5LongAdd(a, b)
	End Select
End Function
```

```
' ----------------------------------------
Private Function MD5Rotate(lngValue, lngBits)
	Dim lngSign
	Dim lngI
	lngBits = (lngBits Mod 32)
	If lngBits = 0 Then MD5Rotate = lngValue: Exit Function
	For lngI = 1 To lngBits
		lngSign = lngValue And &HC0000000
		lngValue = (lngValue And &H3FFFFFFF) * 2
		lngValue = lngValue Or ((lngSign < 0) And 1) Or (CBool(lngSign And
&H40000000) And &H80000000)
	Next
	MD5Rotate = lngValue
End Function
' --------------------
Private Function TRID()
	Dim sngNum, lngnum
	Dim strResult
	sngNum = Rnd(2147483648)
	strResult = CStr(sngNum)
	strResult = Replace(strResult, "0.", "")
	strResult = Replace(strResult, ".", "")
	strResult = Replace(strResult, "E-", "")
	TRID = strResult
End Function
' ------------------------------------------------
Private Function MD564Split(lngLength, bytBuffer())

	Dim lngBytesTotal, lngBytesToAdd
	Dim intLoop, intLoop2, lngTrace
	Dim intInnerLoop, intLoop3
	lngBytesTotal = lngTrack Mod 64
	lngBytesToAdd = 64 - lngBytesTotal
	lngTrack = (lngTrack + lngLength)
	If lngLength >= lngBytesToAdd Then
		For intLoop = 0 To lngBytesToAdd - 1
			arrSplit64(lngBytesTotal + intLoop) = bytBuffer(intLoop)
		Next
		MD5Conversion arrSplit64
		lngTrace = (lngLength) Mod 64
	For intLoop2 = lngBytesToAdd To lngLength - intLoop - lngTrace Step 64
		For intInnerLoop = 0 To 63
		arrSplit64(intInnerLoop) = bytBuffer(intLoop2 + intInnerLoop)
		Next

		MD5Conversion arrSplit64
		Next
		lngBytesTotal = 0
```

```
        Else
                intLoop2 = 0
        End If
        For intLoop3 = 0 To lngLength - intLoop2 - 1
        arrSplit64(lngBytesTotal + intLoop3) = bytBuffer(intLoop2 + intLoop3)
        Next
End Function
' ------------------------------------
Private Function MD5StringArray(strInput)
        Dim intLoop
        Dim bytBuffer()
        ReDim bytBuffer(Len(strInput))
        For intLoop = 0 To Len(strInput) - 1
                bytBuffer(intLoop) = Asc(Mid(strInput, intLoop + 1, 1))
        Next
        MD5StringArray = bytBuffer
End Function
' ----------------------------------
Private Sub MD5Conversion(bytBuffer())
        Dim X(16), a
        Dim b, C
        Dim d
        a = arrLongConversion(1)
        b = arrLongConversion(2)
        C = arrLongConversion(3)
        d = arrLongConversion(4)
        MD5Decode 64, X, bytBuffer
        MD5Round "FF", a, b, C, d, X(0), S11, -680876936
        MD5Round "FF", d, a, b, C, X(1), S12, -389564586
        MD5Round "FF", C, d, a, b, X(2), S13, 606105819
        MD5Round "FF", b, C, d, a, X(3), S14, -1044525330
        MD5Round "FF", a, b, C, d, X(4), S11, -176418897
        MD5Round "FF", d, a, b, C, X(5), S12, 1200080426
        MD5Round "FF", C, d, a, b, X(6), S13, -1473231341
        MD5Round "FF", b, C, d, a, X(7), S14, -45705983
        MD5Round "FF", a, b, C, d, X(8), S11, 1770035416
        MD5Round "FF", d, a, b, C, X(9), S12, -1958414417
        MD5Round "FF", C, d, a, b, X(10), S13, -42063
        MD5Round "FF", b, C, d, a, X(11), S14, -1990404162
        MD5Round "FF", a, b, C, d, X(12), S11, 1804603682
        MD5Round "FF", d, a, b, C, X(13), S12, -40341101
        MD5Round "FF", C, d, a, b, X(14), S13, -1502002290
        MD5Round "FF", b, C, d, a, X(15), S14, 1236535329
        MD5Round "GG", a, b, C, d, X(1), S21, -165796510
        MD5Round "GG", d, a, b, C, X(6), S22, -1069501632
        MD5Round "GG", C, d, a, b, X(11), S23, 643717713
        MD5Round "GG", b, C, d, a, X(0), S24, -373897302
        MD5Round "GG", a, b, C, d, X(5), S21, -701558691
```

```
        MD5Round "GG", d, a, b, C, X(10), S22, 38016083
        MD5Round "GG", C, d, a, b, X(15), S23, -660478335
        MD5Round "GG", b, C, d, a, X(4), S24, -405537848
        MD5Round "GG", a, b, C, d, X(9), S21, 568446438
        MD5Round "GG", d, a, b, C, X(14), S22, -1019803690
        MD5Round "GG", C, d, a, b, X(3), S23, -187363961
        MD5Round "GG", b, C, d, a, X(8), S24, 1163531501
        MD5Round "GG", a, b, C, d, X(13), S21, -1444681467
        MD5Round "GG", d, a, b, C, X(2), S22, -51403784
        MD5Round "GG", C, d, a, b, X(7), S23, 1735328473
        MD5Round "GG", b, C, d, a, X(12), S24, -1926607734
        MD5Round "HH", a, b, C, d, X(5), S31, -378558
        MD5Round "HH", d, a, b, C, X(8), S32, -2022574463
        MD5Round "HH", C, d, a, b, X(11), S33, 1839030562
        MD5Round "HH", b, C, d, a, X(14), S34, -35309556
        MD5Round "HH", a, b, C, d, X(1), S31, -1530992060
        MD5Round "HH", d, a, b, C, X(4), S32, 1272893353
        MD5Round "HH", C, d, a, b, X(7), S33, -155497632
        MD5Round "HH", b, C, d, a, X(10), S34, -1094730640
        MD5Round "HH", a, b, C, d, X(13), S31, 681279174
        MD5Round "HH", d, a, b, C, X(0), S32, -358537222
        MD5Round "HH", C, d, a, b, X(3), S33, -722521979
        MD5Round "HH", b, C, d, a, X(6), S34, 76029189
        MD5Round "HH", a, b, C, d, X(9), S31, -640364487
        MD5Round "HH", d, a, b, C, X(12), S32, -421815835
        MD5Round "HH", C, d, a, b, X(15), S33, 530742520
        MD5Round "HH", b, C, d, a, X(2), S34, -995338651
        MD5Round "II", a, b, C, d, X(0), S41, -198630844
        MD5Round "II", d, a, b, C, X(7), S42, 1126891415
        MD5Round "II", C, d, a, b, X(14), S43, -1416354905
        MD5Round "II", b, C, d, a, X(5), S44, -57434055
        MD5Round "II", a, b, C, d, X(12), S41, 1700485571
        MD5Round "II", d, a, b, C, X(3), S42, -1894986606
        MD5Round "II", C, d, a, b, X(10), S43, -1051523
        MD5Round "II", b, C, d, a, X(1), S44, -2054922799
        MD5Round "II", a, b, C, d, X(8), S41, 1873313359
        MD5Round "II", d, a, b, C, X(15), S42, -30611744
        MD5Round "II", C, d, a, b, X(6), S43, -1560198380
        MD5Round "II", b, C, d, a, X(13), S44, 1309151649
        MD5Round "II", a, b, C, d, X(4), S41, -145523070
        MD5Round "II", d, a, b, C, X(11), S42, -1120210379
        MD5Round "II", C, d, a, b, X(2), S43, 718787259
        MD5Round "II", b, C, d, a, X(9), S44, -343485551
        arrLongConversion(1) = MD5LongAdd(arrLongConversion(1), a)
        arrLongConversion(2) = MD5LongAdd(arrLongConversion(2), b)
        arrLongConversion(3) = MD5LongAdd(arrLongConversion(3), C)
        arrLongConversion(4) = MD5LongAdd(arrLongConversion(4), d)
End Sub
```

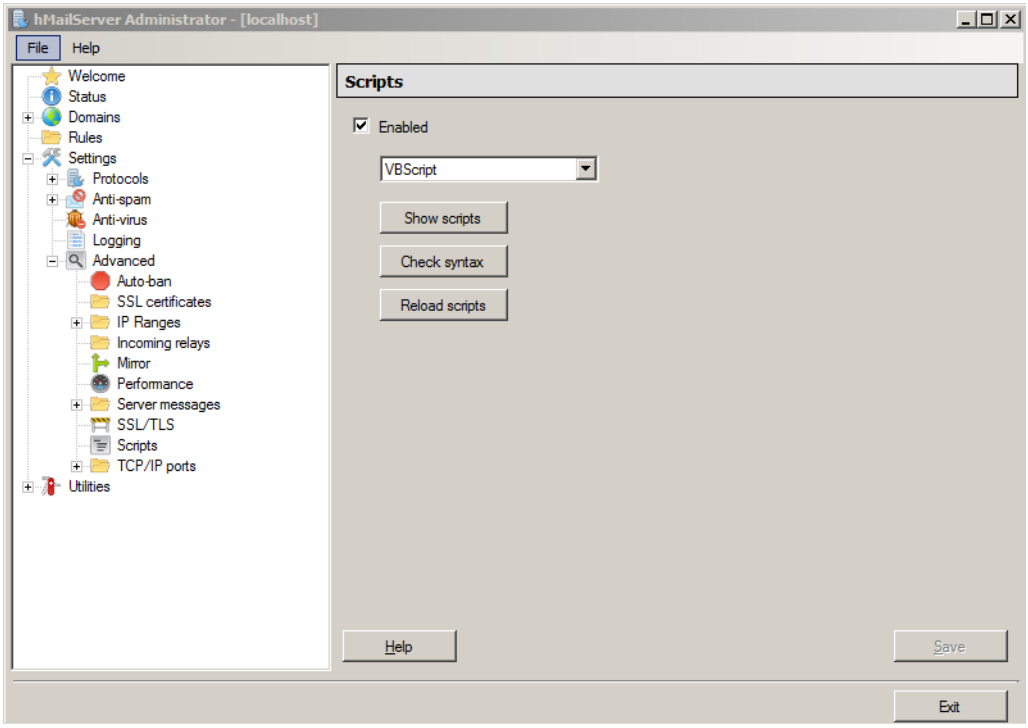
```
' ----------------------------------------
Private Function MD5LongAdd(lngVal1, lngVal2)
	Dim lngHighWord
	Dim lngLowWord
	Dim lngOverflow
	lngLowWord = (lngVal1 And &HFFFF&) + (lngVal2 And &HFFFF&)
	lngOverflow = lngLowWord \ 65536
	lngHighWord = (((lngVal1 And &HFFFF0000) \ 65536) + ((lngVal2 And &HFFFF0000) \ 65536) + lngOverflow) And &HFFFF&
	MD5LongAdd = MD5LongConversion((lngHighWord * 65536) + (lngLowWord And &HFFFF&))
End Function
' ----------------------------------------------------------
Private Function MD5LongAdd4(lngVal1, lngVal2, lngVal3, lngVal4)
	Dim lngHighWord
	Dim lngLowWord
	Dim lngOverflow
	lngLowWord = (lngVal1 And &HFFFF&) + (lngVal2 And &HFFFF&) + (lngVal3 And &HFFFF&) + (lngVal4 And &HFFFF&)
	lngOverflow = lngLowWord \ 65536
	lngHighWord = (((lngVal1 And &HFFFF0000) \ 65536) + ((lngVal2 And &HFFFF0000) \ 65536) + ((lngVal3 And &HFFFF0000) \ 65536) + ((lngVal4 And &HFFFF0000) \ 65536) + lngOverflow) And &HFFFF&
	MD5LongAdd4 = MD5LongConversion((lngHighWord * 65536) + (lngLowWord And &HFFFF&))
End Function
' ----------------------------------------------------------
Private Sub MD5Decode(intLength, lngOutBuffer(), bytInBuffer())
	Dim intDblIndex
	Dim intByteIndex
	Dim dblSum
	intDblIndex = 0
	For intByteIndex = 0 To intLength - 1 Step 4
		dblSum = bytInBuffer(intByteIndex) + bytInBuffer(intByteIndex + 1) * 256 + bytInBuffer(intByteIndex + 2) * 65536 + bytInBuffer(intByteIndex + 3) * 16777216
		lngOutBuffer(intDblIndex) = MD5LongConversion(dblSum)
		intDblIndex = (intDblIndex + 1)
	Next
End Sub
' ----------------------------------------
Private Function MD5LongConversion(dblValue)
	If dblValue < 0 Or dblValue >= OFFSET_4 Then Error 6
	If dblValue <= MAXINT_4 Then
		MD5LongConversion = dblValue
	Else
		MD5LongConversion = dblValue - OFFSET_4
	End If
```

```
End Function
' --------------------
Private Sub MD5Finish()
	Dim dblBits
	Dim arrPadding(72)
	Dim lngBytesBuffered
	arrPadding(0) = &H80
	dblBits = lngTrack * 8
	lngBytesBuffered = lngTrack Mod 64
	If lngBytesBuffered <= 56 Then
		MD564Split (56 - lngBytesBuffered), arrPadding
	Else
		MD564Split (120 - lngTrack), arrPadding
	End If
	arrPadding(0) = MD5LongConversion(dblBits) And &HFF&
	arrPadding(1) = MD5LongConversion(dblBits) \ 256 And &HFF&
	arrPadding(2) = MD5LongConversion(dblBits) \ 65536 And &HFF&
	arrPadding(3) = MD5LongConversion(dblBits) \ 16777216 And &HFF&
	arrPadding(4) = 0
	arrPadding(5) = 0
	arrPadding(6) = 0
	arrPadding(7) = 0
	MD564Split 8, arrPadding
End Sub
' ------------------------------------
Private Function MD5StringChange(lngnum)
	Dim bytA
	Dim bytB
	Dim bytC
	Dim bytD
	bytA = lngnum And &HFF&
	If bytA < 16 Then
		MD5StringChange = "0" & Hex(bytA)
	Else
		MD5StringChange = Hex(bytA)
	End If

	bytB = (lngnum And &HFF00&) \ 256
	If bytB < 16 Then
		MD5StringChange = MD5StringChange & "0" & Hex(bytB)
	Else
		MD5StringChange = MD5StringChange & Hex(bytB)
	End If

	bytC = (lngnum And &HFF0000) \ 65536
	If bytC < 16 Then
		MD5StringChange = MD5StringChange & "0" & Hex(bytC)
	Else
```

```
            MD5StringChange = MD5StringChange & Hex(bytC)
        End If
        If lngnum < 0 Then
            bytD = ((lngnum And &H7F000000) \ 16777216) Or &H80&
        Else
            bytD = (lngnum And &HFF000000) \ 16777216
        End If
        If bytD < 16 Then
            MD5StringChange = MD5StringChange & "0" & Hex(bytD)
        Else
            MD5StringChange = MD5StringChange & Hex(bytD)
        End If
End Function
' ------------------------
Private Function MD5Value()
        MD5Value = LCase(MD5StringChange(arrLongConversion(1)) &
MD5StringChange(arrLongConversion(2)) &
MD5StringChange(arrLongConversion(3)) &
MD5StringChange(arrLongConversion(4)))
End Function
' -------------------------------------------------
Public Function CalculateMD5(strMessage)
        Dim bytBuffer
        bytBuffer = MD5StringArray(strMessage)
        MD5Start
            MD564Split Len(strMessage), bytBuffer
        MD5Finish
        CalculateMD5 = MD5Value
End Function
' --------------------
Private Sub MD5Start()
        lngTrack = 0
        arrLongConversion(1) = MD5LongConversion(1732584193)
        arrLongConversion(2) = MD5LongConversion(4023233417)
        arrLongConversion(3) = MD5LongConversion(2562383102)
        arrLongConversion(4) = MD5LongConversion(271733878)
End Sub
```

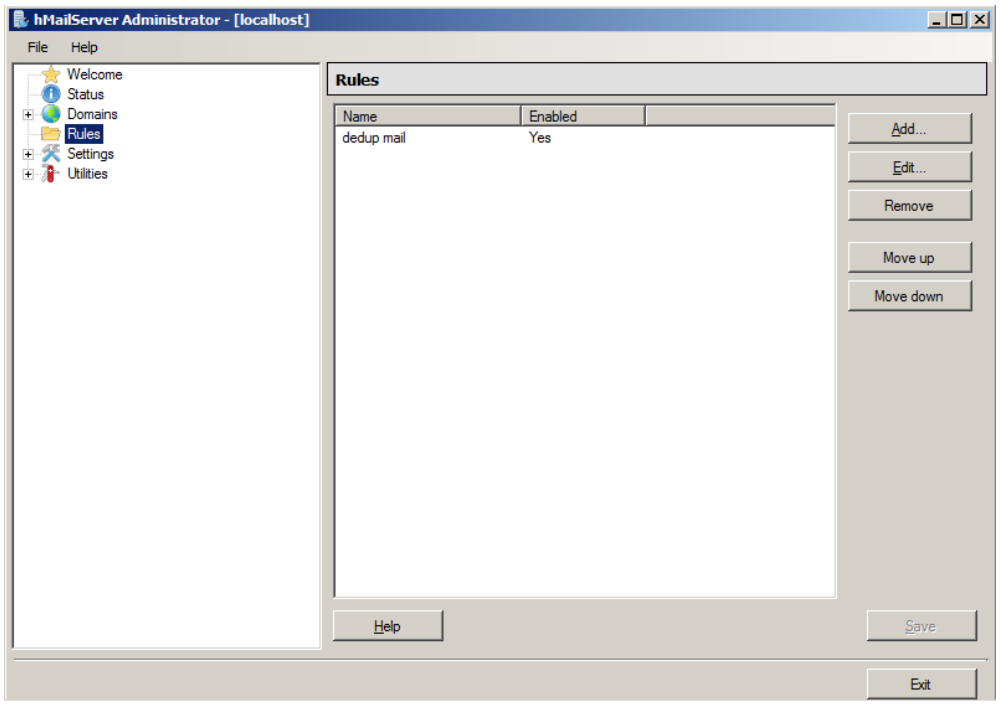
Để kịch bản hoạt động trong hMailServer cần tiến hành các bước như sau:

- Bước 1: Sao chép kịch bản vào trong tệp tin "EventHandlers.vbs" thuộc thư mục cài đặt chương trình hMailServer trên máy chủ, thông thường thư mục mặc định là: C:\Program Files\hMailServer\Events.
- Bước 2: Chạy chương trình quản trị hMailServer, chọn mục "Script" tại menu Setting, tiếp đó chọn "Enable" để kích hoạt Script và click chọn "Reload scripts" để cập nhật kịch bản mới nhất cho hMailServer, cuối cùng chọn "Save" để lưu lại:



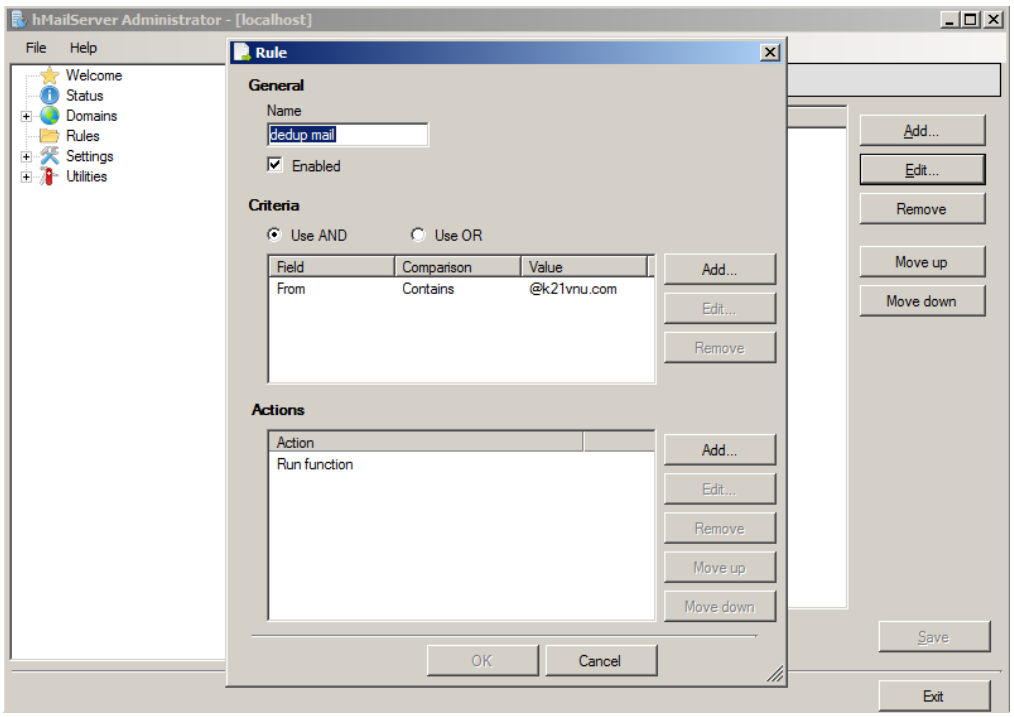
Hình 3.32. Cài đặt kịch bản tích hợp chức năng deduplication

- Bước 3: Tạo ra một Rule để áp dụng kịch bản trong một trường hợp nào đó. Mở chương trình quản trị hMailServer, chọn mục "Rule", sau đó tiến hành thêm mới một Rule cho tính năng Data Deduplication:



Hình 3.33. Tạo Rule để kích hoạt kịch bản

Cấu hình chi tiết Rule: khi email gửi đi sẽ được kiểm tra xem email đó có được gửi đến một nhóm email nào đó không, nếu có thì kích hoạt kịch bản được tạo ra ở trên:

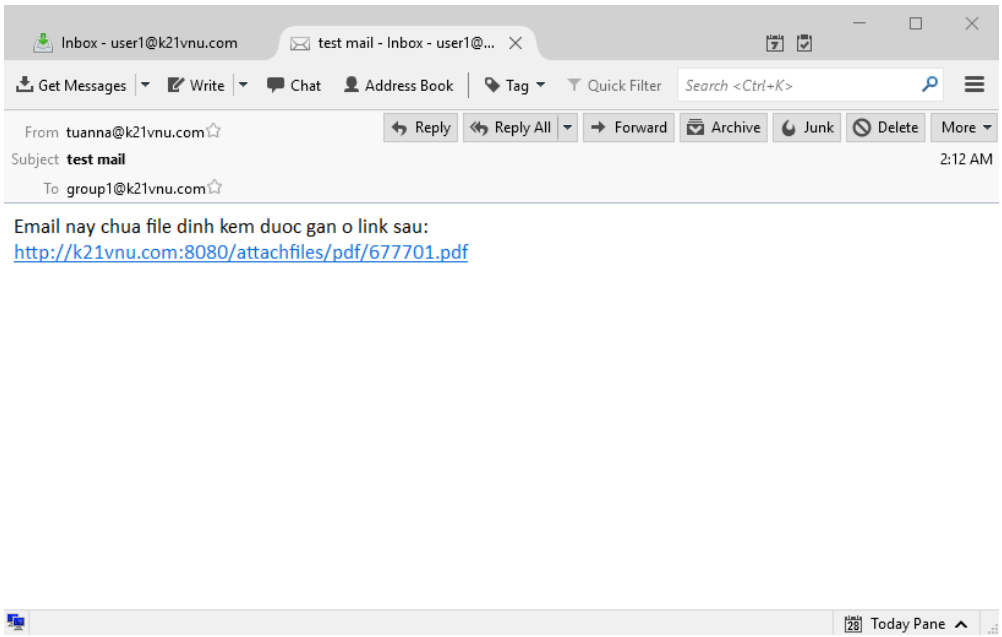


Hình 3.34. Chi tiết cấu hình Rule để kích hoạt kịch bản

3.3.3. Hoạt động của hMailServer trong trường hợp tích hợp Deduplication

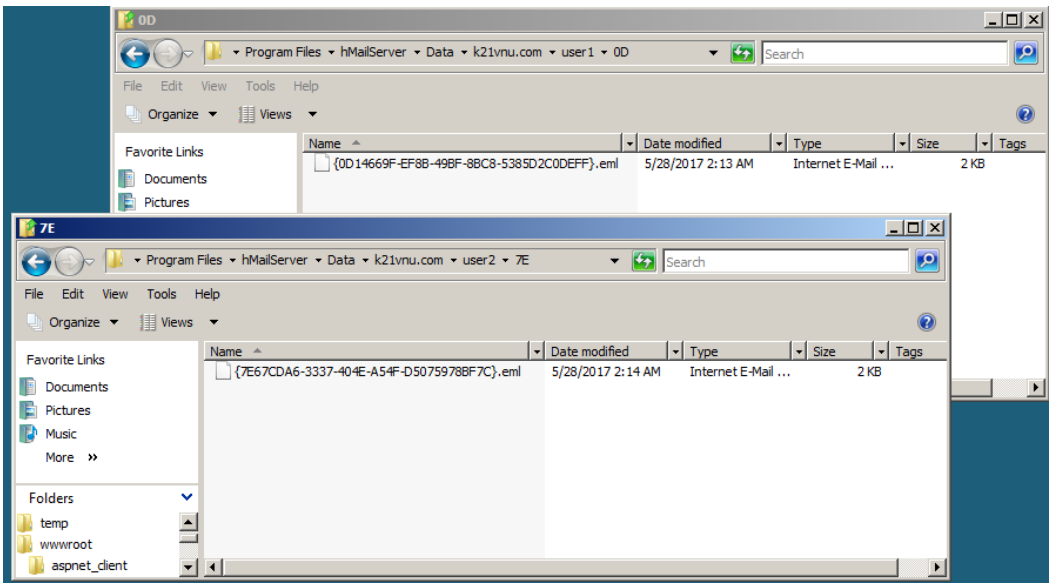
Khi tích hợp tính năng Data Deduplication, hoạt động gửi / nhận email có chứa tệp đính kèm cho nhóm người dùng của hMailServer thay đổi như sau:

- Người dùng thuộc các nhóm mail sẽ nhận được email có đường link trỏ đến tệp đính kèm được lưu trữ:



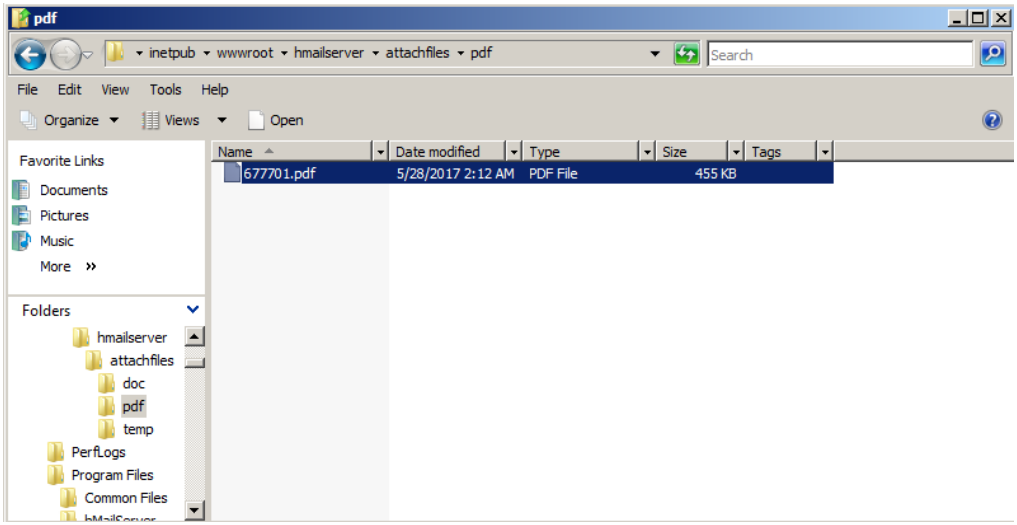
Hình 3.35. Người dùng nhận được email khi triển khai tính năng deduplication

- Tại hòm thư của mỗi người nhận, dung lượng tệp chứa mail được giảm đáng kể, gần giống như dung lượng của một email không có tệp đính kèm:



Hình 3.36. Email được lưu tại hòm thư của người nhận với dung lượng nhỏ

- Tệp đính kèm được lưu một bản duy nhất tại một thư mục được thiết lập trước trên máy chủ hMailServer:

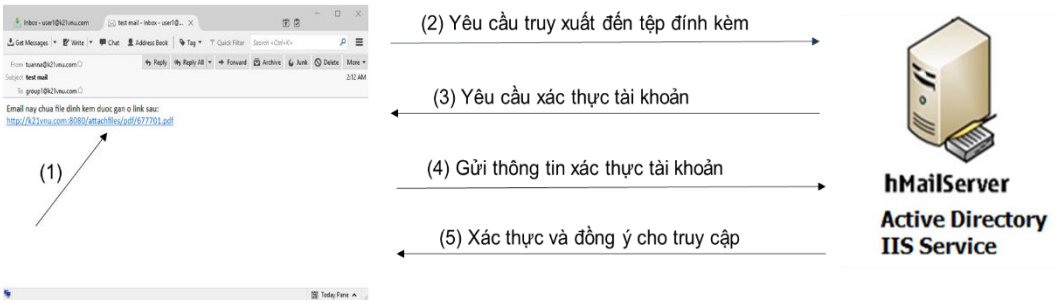


Hình 3.37. Tệp đính kèm được lưu chỉ một bản trên máy chủ hMailServer

3.3.4. Tính bảo mật của hệ thống

Tính bảo mật là một trong những yêu tố quan trọng cho bất kỳ ứng dụng nào khi được triển khai thực tế. Việc triển khai hệ thống đã đảm bảo được các yêu cầu như sau:

- Tệp tin đính kèm trong email được lưu trữ tập trung trên máy chủ mail, chỉ có người quản trị hệ thống hoặc những tài khoản được cấp phép truy cập vào máy chủ mới có quyền truy cập đến vùng chứa dữ liệu của các thư mục này. Việc phân quyền cho các thư mục chứa tệp tin đính kèm được thực hiện theo phân quyền dựa trên hệ điều hành windows.
- Khi người dùng nhận được email có chứa đường link tham chiếu đến tệp tin đính kèm, người dùng khi click vào đường link truy cập cần phải được hệ thống xác thực bằng việc cung cấp tài khoản chứng thực trùng với tài khoản trên cơ sở dữ liệu của dịch vụ Active Directory.



Hình 3.38. Mô tả quá trình chứng thực khi người dùng truy cập tệp tin đính kèm

3.4. So sánh kết quả thực nghiệm

Bằng việc triển khai tính năng Data Deduplication dựa trên các tệp đính kèm, chúng ta có thể thấy hMailServer sẽ tiết kiệm được không gian lưu trữ đáng kể cho máy chủ email Server.

Bảng 3.1. So sánh gần đúng kết quả khi sử dụng Data Deduplication

Dung lượng tệp đính kèm trong email (1)	Số lượng người nhận trong email gửi đi (2)	Dung lượng đĩa cứng dùng lưu trữ khi hMailServer chưa có tính năng deduplication (3)	Dung lượng đĩa cứng dùng lưu trữ khi hMailServer được tính hợp tính năng deduplication (4)	Dung lượng cần để lưu trữ (4) giảm so với (3)
1 MB	1	1 MB	1 MB	0 %
1 MB	10	10 MB	1 MB	90 %
10 MB	1	10 MB	10 MB	0 %
10 MB	10	100 MB	10 MB	90 %
10 MB	50	500 MB	10 MB	98 %
X (MB)	Y	X * Y (MB)	X (MB)	(1 – 1/Y) %

Như tính toán gần đúng ở bảng trên, dung lượng đĩa cứng trên máy chủ hMailServer dùng để lưu trữ sẽ tiết kiệm được (1-1/Y) % so với thông thường. Trong đó, Y là số lượng người nhận trong email gửi đi. Nếu số lượng người nhận càng lớn thì càng tiết kiệm không gian lưu trữ dữ liệu so với thông thường.

Việc tiết kiệm không giảm lưu trữ cho máy chủ sẽ giúp cho bất kỳ một tổ chức nào khi triển khai hệ thống email có thể tiết kiệm nhiều chi phí từ việc đầu tư thiết bị phần cứng, trang bị hạ tầng vật lý như thiết bị điện, không gian đặt thiết bị. Ngoài ra, việc vận hành, bảo trì, sao lưu hệ thống cũng được cải thiện do lượng dữ liệu truyền đi trên mạng được giảm thiểu đáng kể.

KẾT LUẬN

Như vậy, việc ứng dụng thành công kỹ thuật Data Deduplication trong hệ thống hMailServer nói riêng và các hệ thống lưu trữ dữ liệu nói chung chắc chắn sẽ đem lại một lợi ích to lớn cho người dùng và các nhà cung cấp dịch vụ. Tùy theo từng ứng dụng cụ thể trong thực tế mà chúng ta có thể lựa chọn các công nghệ phù hợp nhất để triển khai nhằm đem lại hiệu quả tối đa về chi phí và tăng hiệu năng hoạt động của hệ thống.

Luận văn đã thực hiện được các nội dung chính:

- Nắm được tổng quan về kỹ thuật Data Deduplication, tổng quan về email và mỗi tương quan giữa email với Data Deduplication.
- Các phương thức xử lý Data Deduplication nói chung và đề xuất giải pháp cho việc xử lý dữ liệu trùng lặp trong hệ thống email.
- Trình bày tổng quan về máy chủ hMailServer và mở rộng tính năng Data Deduplication cho hệ thống hMailServer.
- Đánh giá ở mức cơ bản về hiệu quả của kỹ thuật Data Deduplication khi triển khai cho hệ thống hMailServer so với hệ thống khi hoạt động thông thường.

Tuy nhiên, do khả năng tìm hiểu và kiến thức của bản thân có hạn nên bên cạnh những kết quả đạt được, luận văn vẫn còn có những mặt hạn chế nhất định:

- Kịch bản triển khai hiện chỉ xử lý được dữ liệu dư thừa ở mức cơ bản nhất, chưa xử lý được mọi vấn đề về chống trùng lặp dữ liệu trong hệ thống hMailServer.
- Trong trường hợp sử dụng máy chủ email không phải phần mềm hMailServer, cần phải có sự phân tích kỹ lưỡng để có được giải pháp chống trùng lặp dữ liệu phù hợp với mỗi máy chủ email. Kịch bản triển khai trong luận văn cho máy chủ hMailServer không thể áp dụng đồng nhất cho tất cả các máy chủ email khác.

TÀI LIỆU THAM KHẢO

1. Stephen J.Bigelow (2007), Data Deduplication Explained. *Storage Magazine.*
2. Jaspreet Singh. Understanding Data Deduplication. [online] Available at: http://www.druva.com/blog/understanding-data-deduplication/ [Accessed 28 July 2016].
3. Chris Poelker (2013). Data deduplication in the cloud explained. [online] Available at:http://www.computerworld.com/article/2474479/data-center/data-deduplication-in-the-cloud-explained--part-one.html [Accessed 24 July 2016]
4. Lauren Whitehouse. The pros and cons of file-level vs. block-level data deduplication technolog. [online] Available at: http://searchdatabackup.techtarget.com/tip/The-pros-and-cons-of-file-level-vs-block-level-data-deduplication-technology [Accessed 24 July 2016]
5. Todd Erickson. Deduplication best practices and choosing the best dedupe technology. [online] Available at: http://searchdatabackup.techtarget.com/Deduplication-best-practices-and-choosing-the-best-dedupe-technology [Accessed 28 July 2016]
6. Data deduplication technology review. [online] Available at: http://www.computerweekly.com/report/Data-deduplication-technology-review [Accessed 28 July 2016].
7. Data deduplication methods: File-level vs Block-level vs byte-level deduplication. [online] Available at: https://www.starwindsoftware.com/file-level-vs-block-level-vs-byte-level-deduplication [Accessed 05 August 2016].
8. Lauren Whitehouse. Data deduplication methods: Block-level versus byte-level dedupe. [online] Available at: http://searchdatabackup.techtarget.com/tip/Data-deduplication-methods-Block-level-versus-byte-level-dedupe [Accessed 05 August 2016]
9. Email - Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/Email [Accessed 05 August 2016]
10. Introduction To Email. CWS Community Workshop Series. *University of North Carolina at Chapel Hill Libraries | Chapel Hill Public Library | Carrboro Branch Library | Carrboro Cybrary | Durham Public Library.*
11. Sharanjeet Hundal, Tanveer Singh, Basavasai Konuru (2012). *A Final Project Presented to The Faculty of the Department of General Engineering*. San José State University.
12. Lawrence Hughes. *Internet E-mail: Protocols, Standards, and Implementation.* Artech House Telecommunications Library in London.
13. What is an Email Header?. [online] Available at:

http://whatismyipaddress.com/email-header [Accessed 05 August 2016].

14. MIME - Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/MIME [Accessed 05 August 2016]
15. Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/Comparison_of_mail_servers Accessed 05 August 2016]
16. GK_RAJ. Deduplication Internals – Source Side & Target Side Deduplication. [online] Available at: https://pibytes.wordpress.com/2013/03/09/deduplication-internals-source-side-target-side-deduplication-part-4/ [Accessed 28 July 2016]
17. Mark R. Coppock and Steve Whitner. *Data Deduplication for Dummies*, Quantum 2nd Special Edition), Wiley Publishing Inc, Indiana.
18. hMailServer – Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/HMailServer [Accessed 05 August 2016]
19. 19. hMailServer – Homepage. [online] Available at: https://www.hmailserver.com/ [Accessed 05 August 2016]
20. hMailServer – COM API. [online] Available at: https://www.hmailserver.com/documentation/latest/?page=com_objects [Acccessed 05 August 2016]
21. hMailServer – COM API: Examples. [online] Available at: https://www.hmailserver.com/documentation/latest/?page=com_examples [Accessed 05 August 2016]
22. VBA – Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/Visual_Basic_for_Applications [Accessed 05 August 2016]
23. SourceCode of hMailServer on GitHub. [online] Available at: https://github.com/hmailserver/hmailserver [Accessed 05 August 2016]
24. Download Xampp for Windows. [online] Available at: https://www.apachefriends.org/download.html [Accessed 05 August 2016]
25. Download Squirrelmail-Webmail for Windows. [online] Available at: https://squirrelmail.org/download.php [Accessed 05 August 2016]
26. hMailServer – Configuration. [online]. Available at: https://www.hmailserver.com/documentation/latest/?page=overview [Accesed 05 August 2016]