**ĐẠI HỌC QUỐC GIA TP.HCM**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH**

****

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**NGHIÊN CỨU THỰC NGHIỆM**

**BỘ ĐỆM CHO VIDEO**

**HỘI ĐỒNG : HỆ THỐNG THÔNG TIN**

**GVHD : TS BÙI HOÀI THẮNG**

**GVPB : Th.S ĐẶNG TRẦN TRÍ**

**---o0o---**

**SVTH : ĐẶNG XUÂN HIẾU (50700731)**

TP. HỒ CHÍ MINH, THÁNG 01/2012

# LỜI CAM ĐOAN

*Ngoài các tài liệu tham khảo được liệt kê tôi xin cam đoan toàn bộ phần nghiên cứu và trình bày luận văn đều được tôi thực hiện dưới sự hướng dẫn của thầy Bùi Hoài Thắng, không có bất kỳ sự sao chép từ các nguồn tài liệu cũng như các công trình nghiên cứu khác.*

*Nếu có bất kỳ sai phạm nào, chúng tôi xin chịu trách nhiệm trước Hội đồng bảo vệ, Ban chủ nhiệm khoa và Ban giám hiệu Nhà trường.*

*Tp.Hồ Chí Minh, tháng 01 năm 2012*

# LỜI CẢM ƠN

*Sau mười bốn tuần thực hiện đề tài, bên cạnh sự nỗ lực của cá nhân, chúng tôi còn được nhận sự hỗ trợ nhiệt tình từ thầy cũng như các thành viên khác trong lớp. Điều này đã giúp tôi rất nhiều trong việc bắt kịp tiến độ đã đề ra và hoàn thiện hơn đề tài của mình.*

*Tôi xin được gửi lời cám ơn chân thành nhất đến thầy Bùi Hoài Thắng – giảng viên hướng dẫn trực tiếp đề tài - là người đã luôn theo dõi sát sao, cung cấp tài liệu, cũng như góp ý, sửa chữa những sai sót của chúng tôi trong suốt quá trình thực hiện.*

*Tôi xin cảm ơn quý thầy cô và khoa Khoa Học & Kỹ Thuật Máy Tính đã tận tình giảng dạy và tạo điều kiện thuận lợi cho chúng tôi học tập, nghiên cứu và thực hiện đề tài này.*

*Những lời cảm ơn sau cùng xin dành cho gia đình và bạn bè, những người đã cho tôi sự động viên, quan tâm và giúp đỡ kịp thời nhất để vượt qua những khó khăn trong cuộc sống và trong quá trình hoàn thành luận văn.*

*Tp.Hồ Chí Minh, tháng 01 năm 2012*

# TÓM TẮT LUẬN VĂN

Sự phát triển nhanh chóng của các ngành khoa học, đặc biệt là ngành máy tính đã góp phần rất lớn trong sự phát triển của nhiều lĩnh vực kinh tế, xã hội, văn hóa… trong đó có sự phát triển của giáo dục. Với sự phát triển của lĩnh vực công nghệ thông tin thì hình thức học trực tuyến đang dần được quan tâm và phát triển vì chi phí ít, tính cơ động, tiện lợi cao. Mặc dù trên thế giới, hình thức này đã phổ biến ở nhiều nơi nhưng ở Việt Nam vẫn còn rất hạn chế do các vấn đề về công nghệ, cơ sở hạ tầng kỹ thuật,...

Đối với hình thức học trực tuyến, có rất nhiều loại tài nguyên học tập được sử dụng trong quá trình đào tạo, trong đó có những tài liệu định dạng video. Việc quản lý, phân phối và bảo vệ bản quyền video đang là một thách thức lớn cho các hệ thống đào tạo trực tuyến ngày nay.

Cho đến nay, đa số các dịch vụ trình chiếu video trực tuyến đều sử dụng tập tin tạm của hệ thống như một công cụ trung gian để trình chiếu video phía máy khách khiến cho các chương trình hỗ trợ download như Internet Download Manager (IDM), Flashget, Orbit Download Manager, … có thể dễ dàng ghép nối và lưu trữ tập tin video đó lại trên máy tính. Người xem dễ dàng phân phát lại video cho bạn bè hoặc chia sẻ thông qua các hình thức chia sẻ tập tin khác mà chủ nhân thật sự của video hoàn toàn không thể quản lý được.

Mục tiêu của đề tài là dùng bộ nhớ đệm (buffer) ớ phía máy khách làm công cụ trung gian để trình chiếu tập tin video. Những đoạn video sau khi được chạy qua sẽ được xóa khỏi bộ đệm để ngăn cản các chương trình hỗ trợ download có thể lưu trữ lại. Về phía server, sẽ đưa kỹ thuật video caching vào hỗ trợ máy chủ đáp ứng với số kết nối lớn, giải quyết vấn đề quá tải băng thông mạng và quá tải máy chủ, để tăng tốc độ truyền tải video đến người dùng mà không hạn chế về dung lượng hay đường truyền.

# MỤC LỤC

[LỜI CAM ĐOAN ii](#_Toc312645169)

[LỜI CẢM ƠN iii](#_Toc312645170)

[TÓM TẮT LUẬN VĂN iv](#_Toc312645171)

[MỤC LỤC v](#_Toc312645172)

[MỤC LỤC HÌNH viii](#_Toc312645173)

[MỤC LỤC BẢNG ix](#_Toc312645174)

[CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 1](#_Toc312645175)

[1. Đặt vấn đề 2](#_Toc312645176)

[2. Mục tiêu đề tài 2](#_Toc312645177)

[3. Phương pháp thực hiện 3](#_Toc312645178)

[4. Những công trình liên quan 3](#_Toc312645179)

[4.1. Những giải pháp trình chiếu video hiện tại 3](#_Toc312645184)

[4.2. Những vấn đề còn tồn tại 4](#_Toc312645185)

[5. Bố cục luận văn 4](#_Toc312645186)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ 5](#_Toc312645187)

[1. Cơ sở lý thuyết 6](#_Toc312645188)

[1.1. Streaming Video 6](#_Toc312645189)

[1.2. Cache 8](#_Toc312645193)

[1.3. Các loại tập tin video 9](#_Toc312645194)

[1.4. Giải thuật Cache 12](#_Toc312645195)

[2. Công nghệ 14](#_Toc312645196)

[2.1. Java 16](#_Toc312645197)

[2.2. FFmpeg 19](#_Toc312645198)

[CHƯƠNG III: GIẢI QUYẾT BÀI TOÁN 21](#_Toc312645199)

[1. Phân tích và thiết kế hệ thống 22](#_Toc312645200)

[2.1. Mô hình tổng quan 22](#_Toc312645203)

[2.2. Xác lập tính năng hệ thống 22](#_Toc312645204)

[2.3. Kiến trúc hệ thống 23](#_Toc312645205)

[2. Giải pháp kỹ thuật 26](#_Toc312645206)

[2.1. Server 26](#_Toc312645209)

[2.2. Client 27](#_Toc312645210)

[3. Thực hiện 27](#_Toc312645211)

[3.1. Server 27](#_Toc312645213)

[3.2. Client 29](#_Toc312645219)

[3.3. Giao diện 31](#_Toc312645224)

[CHƯƠNG IV: KẾT QUẢ 33](#_Toc312645227)

[1. Những việc đã làm được 34](#_Toc312645228)

[2. Hạn chế 34](#_Toc312645229)

[3. Kiến thức thu được: 35](#_Toc312645230)

[4. Đánh giá hiệu suất: 35](#_Toc312645231)

[4.1. Server 35](#_Toc312645233)

[4.2. Client 37](#_Toc312645234)

[5. Hướng dẫn cài đặt và sử dụng 38](#_Toc312645235)

[5.1. Cài đặt 38](#_Toc312645237)

[5.2. Sử dụng 38](#_Toc312645240)

[CHƯƠNG V: TỔNG KẾT 41](#_Toc312645243)

[1. Hiệu quả giáo dục - kinh tế - xã hội 42](#_Toc312645244)

[1.1. Đối với giáo dục và đào tạo 42](#_Toc312645245)

[1.2. Đối với kinh tế - xã hội 42](#_Toc312645246)

[2. Phạm vi ứng dụng 42](#_Toc312645247)

[3. Hướng phát triển 42](#_Toc312645248)

[TÀI LIỆU THAM KHẢO 43](#_Toc312645249)

[PHỤ LỤC 44](#_Toc312645250)

# MỤC LỤC HÌNH

Hình 2.1 : Flowchart của giải thuật LRU 12

Hình 3.1: Kiến trúc hệ thống 23

Hình 3.2: Lược đồ hoạt động tổng quát 24

Hình 3.3: Lược đồ hoạt động trên Server 25

Hình 3.4 : Lược đồ hoạt động quản lý Buffer phía Client 25

Hình 3.5 : Giải pháp kỹ thuật ở Server 26

Hình 3.6 : Giải pháp kỹ thuật ở Client 27

Hình 3.7 : Giao diện Server khi khởi động chương trình 31

Hình 3.8 : Giao diện Server khi Server “turn on” 31

Hình 3.9 : Giao diện Client khi khởi động chương trình 32

Hình 3.10 : Giao diện của Player 32

Hình 4.1 : Biểu đồ theo dõi tốc độ xử lý dữ liệu của Server 35

Hình 4.2 : Biểu đồ so sánh tốc độ xử lý dữ liệu trong 2 trường hợp: có cache

và không có cache 36

Hình 4.3 : Biểu đồ theo dõi hiệu suất cache 36

Hình 4.4 : Biểu đồ theo dõi tốc độ nhận dữ liệu của Client 37

Hình 4.5 : Giao diện Server khi khởi động chương trình 38

Hình 4.6 : Giao diện Server khi Server “turn on” 39

Hình 4.7 : Shortcut của chương trình 39

Hình 4.8 : Giao diện Client khi khởi động chương trình 39

Hình 4.9 : Giao diện của Player 40

# MỤC LỤC BẢNG

Bảng 2.1 : Các kiểu tập tin video phổ biến 11

# CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

## Đặt vấn đề

Ngày nay, với tốc độ phát triển hạ tầng mạng cùng với trình độ tiếp cận công nghệ thông tin của người dân, việc chia sẻ, trình chiếu video trên các website trở nên rất phổ biến và dễ dàng.

Ý tưởng sử dụng video bài giảng điện tử cho chương trình đào tạo từ xa giúp nâng cao chất lượng và hiệu quả đào tạo trong khi chi phí bỏ ra ít, người dạy và người học chủ động hơn về thời gian và địa điểm đã được rất nhiều tổ chức đào tạo hiện thực và đưa vào sử dụng, đạt được những kết quả cao.

Tuy nhiên, việc trình chiếu video trực tuyến hiện nay vẫn còn khá nhiều hạn chế:

* Mặc dù ngày nay băng thông mạng đã được mở rộng nhưng chất lượng đường truyền vẫn không đồng đều tại các vị trí khác nhau, không thể đáp ứng được nhu cầu sử dụng ngày càng tăng về lưu lượng truy cập, cũng như chất lượng video. Cần một hệ thống không chỉ mạnh mà còn “thông minh”, tự nhận biết được chất lượng, băng thông đường truyền của khách và có chế độ gởi nhận đặc biệt để đảm bảo tốc độ tải, chạy của video không quá lâu, video ít bị đứng hình khi xem.
* Ngoài ra, vấn đề bản quyền của video cũng là điều không được đảm bảo, vì người dùng dễ dàng sử dụng các chương trình tự động kết nối và tải video về. Việc xây dựng hệ thống hỗ trợ trình chiếu và quản lý để bảo vệ bản quyền của video tuân thủ đúng luật áp dụng bảo vệ trách nhiệm cung cấp của các tổ chức uy tín là thực sự cần thiết.

## Mục tiêu đề tài

Với tốc độ phát triển công nghệ thông tin tại Việt Nam, việc trình chiếu video trực tuyến đang rất được quan tâm để đáp ứng nhu cầu học tập và giải trí của người dùng. Cũng như việc tích hợp bài giảng video trực tuyến vào những hệ thống phục vụ chương trình đào tạo từ xa của các tổ chức đào tạo ngày càng trở nên phổ biến hơn.

Mục tiêu của công trình là nghiên cứu và hiện thực một hệ thống ứng dụng video caching theo mô hình client-server để hỗ trợ việc trình chiếu video trực tuyến có tốc độ đáp ứng cao với chi phí bỏ ra thấp.

Ba vấn đề chính được quan tâm trong lĩnh vực trình chiếu video là bảo vệ bản quyền video cho người cung cấp, nâng cao chất lượng trình chiếu video cho người sử dụng và giảm tải cho máy chủ.

Khi nghiên cứu và xây dựng công trình này, chúng tôi mong muốn thành quả của công trình sẽ góp phần vào cuộc chiến chống vi phạm bản quyền trong lĩnh vực công nghệ thông tin, cụ thể là bảo vệ bản quyền các video. Bên cạnh đó, việc áp dụng một giải pháp mới (video caching) nhằm đáp ứng được số lượng người dùng lớn hơn tại một thời điểm, cũng như nâng cao chất lượng trình chiếu video và đáp ứng tốt hơn nhu cầu học tập và giải trí của người dùng.

## Phương pháp thực hiện

Đầu tiên, để hiểu về mô hình hoạt động của các ứng dụng trình chiếu video, chúng tôi tiến hành tìm hiểu và thử nghiệm một số bộ giải mã (decoder) video mã nguồn mở. Hiện thực tác vụ đọc, chia nhỏ các tập tin video có dung lượng nhỏ đưa vào bộ nhớ đệm, sau đó thử nghiệm việc trình chiếu tập tin lấy từ bộ nhớ đệm dựa trên các bộ giải mã đó, để chọn ra một bộ giải mã phù hợp với mục tiêu của đề tài.

Nghiên cứu việc truyền nhận dữ liệu thông qua đường truyền mạng, để từ đó tiếp tục mở rộng hệ thống theo mô hình client-server trên bộ giải mã được chọn. Hiện thực bộ quản lý dữ liệu và việc truyền tải dữ liệu ở cả hai phía hệ thống.

Thử nghiệm việc trình chiếu tập tin video ở phía client bắt đầu với một kết nối đến server, sau đó tăng dần số kết nối lên. Nhận biết và giải quyết các xung đột xảy ra. Tiến hành phân tích kết quả trên dữ liệu thử nghiệm, để đề xuất số lượng kết nối phù hợp với từng chất lượng tập tin video.

Hiện thực và thử nghiệm các giải pháp caching trên server cho việc quản lý và phân phối gói dữ liệu về máy khách, khi có nhiều yêu cầu từ phía máy khách.

## Những công trình liên quan

4. 1. **Những giải pháp trình chiếu video hiện tại**

Hiện nay, có rất nhiều chuẩn công nghệ video streaming để hỗ trợ trình chiếu video được phát triển bởi các tổ chức khác nhau. Cũng như có nhiều định dạng dữ liệu riêng với các chuẩn công nghệ đó. Các định dạng video streaming chỉ giới hạn bởi các công ty dẫn đầu trong công nghệ streaming: Các hãng này đều cung cấp các bộ công cụ trọn gói gồm video Streaming Server (lưu trữ, truyền phát dữ liệu theo các giao thức hỗ trợ...), video Player (hiển thị dữ liệu tại phía người dùng), và công cụ tìm kiếm tạo dữ liệu với các chuẩn nén.

Đa số các dịch vụ trình chiếu video trực tuyến hiện nay đều sử dụng tập tin tạm của hệ thống làm công cụ trung gian để trình chiếu video trực tuyến ở phía máy khách.

Ngoài ra, những dịch vụ như tổ chức họp trực tuyến, chat video… theo chế độ thời gian thực cũng rất phát triển.

Trong thời gian nghiên cứu của luận văn, tôi không tìm thấy được đề tài nghiên cứu nào nói về việc ứng dụng cache để hỗ trợ trình chiếu video.

* 1. **Những vấn đề còn tồn tại**

Với những hệ thống trình chiếu video hiện tại (ví dụ: Youtube, Megafun,…), dữ liệu video được tải trực tiếp về máy dưới dạng tập tin tạm trước khi được ghép lại để trình chiếu, nên với những người dùng ở những nơi có tốc độ mạng thấp (dial up) khoảng thời gian chờ đợi này khá dài khiến việc xem video bị lắc (lag), chất lượng video kém.

Bên cạnh đó, người dùng dễ dàng sử dụng những phần mềm hỗ trợ (ví dụ: Internet Download Manager, FlashGet,…) để tải video về máy để xem, sau đó lại đem video đó đăng tải lên mạng để chia sẻ bất hợp pháp. Như vậy, hệ thống không thể kiểm soát được bản quyền của video và giảm tính tương tác liên tục của người dùng với hệ thống (nhất là trong chương trình đào tạo từ xa, cần theo dõi thời gian hoạt động của người dùng).

Ngoài ra, việc quản lý và phân phối dữ liệu video hiệu quả trở thành một thách thức rất lớn. Theo thời gian, số lượng dữ liệu và số lượng người sử dụng ngày càng tăng có thể dẫn đến vấn đề quá tải băng thông mạng và quá tải máy chủ.

## Bố cục luận văn

Nội dung của luận văn bao gồm 5 chương. Tên và nội dung chính của các chương như sau:

1. *Chương I - Giới thiệu đề tài*

Giới thiệu sơ lược về luận văn, giải thích tại sao đề tài này quan trọng, cần làm và nêu ra các nhiệm vụ phải làm trong luận văn.

1. *Chương II : Cơ sở lý thuyết và công nghệ*

Trình bày cơ sở lý thuyết về các vấn đề Streaming video, Cache và các loại tập tin video cùng với các công nghệ được sử dụng để làm trong luận văn.

1. *Chương II : Giải quyết bài toán*

Trình bày cách giải quyết đề tài.

1. *Chương IV : Kết quả*

Trình bày các kết quả đạt được và đánh giá.

1. *Chương V: Tổng kết*

Một số kết luận về luận văn và hướng phát triển của đề tài.

*Tài liệu tham khảo*

*Phụ lục*

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

## Cơ sở lý thuyết

* 1. **Streaming Video**
     1. **Streaming Video**

Streaming Video là một phương thức để máy chủ đặc biệt cung cấp video thông qua Internet. Streaming Video thường được sử dụng trong lĩnh vực giải trí hoặc dạy học, dùng để lưu trữ các tuyển tập các tập tin video hoặc các bài học, cung cấp cho người dùng các tiện ích như tìm kiếm, liệt kê, và khả năng hiển thị hoặc hiển thị lại các dữ liệu video theo yêu cầu.

Với các định dạng tập tin video truyền thống, dữ liệu chỉ có thể hiển thị khi đã được tải về (download) toàn bộ, vì vậy đối với các tập tin video chất lượng cao có dung lượng lớn sẽ tiêu tốn rất nhiều thời gian để chờ thực hiện công việc này.

* Streaming Video tiết kiệm thời gian cho người dùng bằng cách sử dụng các công nghệ giải nén kết hợp với player hiển thị dữ liệu đồng thời trong lúc vẫn tiếp tục download. Quá trình này được gọi là buffering:
  + Thay vì được gửi một lần duy nhất, dữ liệu streaming video sẽ được truyền đi thành các gói nhỏ.
  + Ban đầu player sẽ lấy về một phần chia nhỏ đó của dữ liệu video trước khi hiển thị, đồng thời trong lúc hiển thị các gói dữ liệu còn lại sẽ lần lượt được lấy về để kịp cho việc hiển thị tiếp theo.
* Streaming Video được thể hiện dưới hai dạng:
  + Video theo yêu cầu (on demand): là các dữ liệu video được lưu trữ trên multimedia server và được truyền đến người dùng khi có yêu cầu, người dùng có toàn quyền để hiển thị cũng như thực hiện các thao tác (tua, dừng, qua …) với các đoạn dữ liệu này.
  + Video thời gian thực (live event): là các dữ liệu video được phát trực tiếp từ các nguồn cung cấp dữ liệu theo thời gian thực (máy camera, microphone, thiết bị phát dữ liệu video…).
    1. **Quá trình Video Streaming**

Video Streaming là thuật ngữ dùng để nói về quá trình máy chủ cung cấp video thông qua Internet. Với Streaming, máy tính của người sử dụng không phải tải toàn bộ video về cùng một lúc. Để thực hiện việc tăng tốc độ lần tải về, tại bất kỳ thời điểm nào máy tính chỉ cần tải về một phần thông tin của video.Với kỹ thuật này, chúng ta cần một máy chủ đặc biệt kiểm soát việc cung cấp các nội dung video. Để bất kỳ phần nào của đoạn video được lưu trữ trên server có thể được truy cập bất kỳ lúc nào, hơn là cần nó chờ để tải về trước khi truy cập nó.

* Video Streaming:
  + True Streaming: tín hiệu video đến theo thời gian thực và hiển thị ngay lập tức cho người xem.
  + Download and Play: Tải toàn bộ video về, sau đó phát video.
  + Progressive Download and Play: Đây là công nghệ lai giữa hai công nghệ trên. Trong công nghệ này, video sẽ được chia nhỏ ra thành nhiều phần nhỏ; chương trình phát video (player) sẽ hiện thị ngay những đoạn (segment) vừa được tải về và lưu trữ dữ liệu đó ở ổ đĩa cục bộ của máy. Khi tua lại những đoạn video mà đã tải về, Player sẽ lấy dữ liệu từ ổ đĩa cục bộ để tải lên.

Video Streaming sử dụng các giao thức RTP, RTMP, MMS, HTTP,.. để truyền dữ liệu theo dạng streaming qua mạng Internet, đồng thời sử dụng các chuẩn nén để giảm dung lượng dữ liệu, cung cấp khả năng nén dữ liệu tại nhiều mức nén, nhiều kích thước hiển thị để có thể phù hợp với độ rộng băng thông của nhiều mạng truyền dẫn để tối ưu hoá việc truyền dữ liệu qua mạng. Cũng chính vì vậy việc truyền các Streaming video qua mạng sẽ phụ thuộc rất nhiều vào các sản phẩm phần mềm video Streaming Server.

* + 1. **Kiến trúc hệ thống Streaming**

Khi hiện thực một hệ thống Streaming trên một mạng IP (hoặc một vài kỹ thuật khác), chúng ta cần đầu tư cho hệ thống một Streaming Server (có nhiệm vụ phân phối Stream cho mỗi thiết bị người dùng).

Streaming Server hoạt động sẽ lấy nội dung (những tập tin video hoặc những tập tin khác trong hệ thống) và tạo Stream cho mỗi yêu cầu người dùng gửi đến. Những Stream này có thể hiện thực theo hai phương thức (mỗi phương thức được điều khiển bởi cơ chế khác nhau) là unicast và multicast. Phổ biến nhất hiện nay là cơ chế unicast.

Chức năng thường được tích hợp trong Streaming Server là lưu trữ và phục hồi nội dung. Công việc xử lý quan trọng của Streaming Server là tạo một gói tin cho mỗi Stream theo thời gian thực. Ngoài ra, Streaming Server còn phải giải quyết các yêu cầu về mã hóa gói dữ liệu để đảm bảo tính bảo mật, nén dữ liệu để gửi đến những kênh Stream có tùy chọn kết nối với tốc độ khác nhau. Một vấn đề quan trọng khác trong hệ thống là phân tích dữ liệu và đưa ra giải thuật giải quyết vấn đề quản lý thiết bị Cache một cách hiệu quả trong Streaming Server.

* 1. **Cache**

Với những hệ thống máy chủ trình chiếu video trực tuyến hoặc cung cấp cho việc tải về theo yêu cầu người dùng đang được sử dụng hiện nay, thì với mỗi yêu cầu của người dùng đến - sẽ được máy chủ xử lý thông tin đảm bảo yêu cầu đó là hợp lệ để thiết lập kết nối; tiếp sau đó, nó sẽ truy xuất đến ổ cứng (hoặc thiết bị lưu trữ cục bộ) để truy xuất tập tin yêu cầu đó và gửi về cho người dùng. Như chúng ta đều biết - tốc độ truy xuất trên ổ cứng thường là khá chậm, đặc biệt khi có nhiều yêu cầu truy cập ngẫu nhiên. Bên cạnh đó, có những tập tin được nhiều người dùng cùng quan tâm trong một thời điểm (sai khác nhỏ về thời gian), việc truy xuất ổ cứng liên tục để truy xuất cùng một tập tin như cách thức làm việc hiện tại gây ra tình trạng “thắt cổ chai” vì phải chờ tác vụ đọc từ ổ cứng là rất lãng phí và tốn kém cho máy chủ.

Để giải quyết vấn đề này, ta đưa ra giải pháp sử dụng Cache. Cache là một cơ chế lưu trữ tốc độ cao đặc biệt. Cache – được hiểu theo nghĩa thường là nơi lưu trữ dữ liệu nằm chờ ứng dụng hoặc phần cứng xử lý, nhằm mục đích tăng tốc độ xử lý. Cache có thể là một vùng lưu trữ của bộ nhớ chính hoặc một thiết bị lưu trữ tốc độ cao độc lập. Có rất nhiều loại Cache được biết đến hiện nay với những chức năng khác nhau như: Cache của CPU, Caching của Internet Browser, Caching của Oracle… Cụ thể ở đây ta quan tâm đến một cơ chế xử lý video Caching ngay trên Server thông qua bộ nhớ chính thay thế cho Web Caching quá tải hiện nay. Việc đầu tư cho thiết bị Cache cộng với việc có một giải pháp quản lý Cache hiệu quả sẽ giảm tải rất nhiều cho máy chủ.

* Cách thức hoạt động thông qua Cache:
  + Dữ liệu được truy xuất từ ổ cứng được đẩy lên Cache.
  + Với mỗi yêu cầu từ người nhận gửi tới, máy chủ sẽ tìm kiếm dữ liệu đã tồn tại trên Cache chưa để thực hiện tác vụ đẩy dữ liệu đó lên Cache; nếu đã tồn tại thì dữ liệu từ Cache - thông qua quá trình xử lý dữ liệu sẽ được truyền trực tiếp đến người dùng thông qua một Stream (bỏ qua giai đoạn truy xuất từ ổ cứng). Tốc độ truy xuất Cache lớn hơn nhiều lần so với ổ cứng vì vậy đây chính là điểm mấu chốt giúp server đáp ứng nhanh hơn các hệ thống cũ hiện tại gấp nhiều lần.
  + Dựa vào việc phân tích, khai phá thông tin truy xuất dữ liệu, cùng với những thử nghiệm những số liệu thực tế để đưa ra một cơ chế để quản lý dữ liệu trên Cache, quyết định dữ liệu nào được đưa vào Cache, hoặc loại bỏ ra khỏi Cache. Khắc phục nhược điểm giới hạn về dung lượng của Cache.
  1. **Các loại tập tin video**

Các tập tin video thường thuộc 2 loại: codec, không codec. Một số định dạng tập tin, chỉ cần nhìn vào đuôi tập tin ta biết được loại codec mà tập tin đó sử dụng như: WMV, RM, MOV… Hoặc các loại đuôi tập tin không chỉ định chính xác loại codec tập tin nào như: MP4, AVI, MKV… Các loại tập tin trên đều là các container để chứa nội dung, chúng ta thường phải sử dụng các trình media player để mở và đọc phần header của các tập tin đó mới biết được tập tin đó sử dụng loại codec nào để mã hóa.

Xem xét một số loại định dạng tập tin mà các website và các nhà cung cấp lớn thường hỗ trợ ta có thể thống kê một vài định dạng video phổ biến như bảng sau:

|  |  |
| --- | --- |
| **Định dạng** | **Giới thiệu** |
| 3GP –  3GPP Multimedia | * Định dạng 3GP là định dạng video được phát triển bởi dự án 3rd Generation Partnership dựa trên chuẩn MPEG-4. Được sử dụng cho các tập tin đa phương tiện trên mạng không dây 3G tốc độ cao, sử dụng phổ biến trên điện thoại có hỗ trợ quay phim. |
| ASF –  Advanced Systems Format | * Định dạng ASF được Microsoft phát triển. Định dạng này được dùng để truyền tải các tập tin đa phương tiện chứa văn bản, đồ họa, âm thanh, video và hoạt họa. Tập tin ASF chủ yếu là tập tin Windows Media Audio và Windows Media video. * Tập tin ASF chỉ đặc tả cấu trúc của tập tin audio hay âm thanh được truyển tải mà không chỉ rõ phương pháp mã hóa. |
| AVI –  Audio video Interleave | * Định dạng AVI cũng được phát triển bởi Microsoft và chứa dữ liệu được mã hóa theo các codec khác nhau( có thể dùng các codec như DivX để mã hóa tập tin AVI). Định dạng AVI bao gồm hai loại định dạng chính DivX, Xvid. Đây là một định dạng rất phổ biển hiện nay. * AVI là một sản phẩm của RIFF. Dữ liệu của tập tin được chia trong các khối, mỗi khối được xác định bởi một nhãn. Khối đầu tiên chứa dữ liệu về video, những thông tin về chiều dài, chiều rộng, tỷ lệ khung. Khối thứ 2 chứa dữ liệu thật sự của tập tin AVI. Khối thứ 3 chứa các chỉ mục của các khối dữ liệu trong tập tin. |
| FLV –  Flash video | * Flash video là một định dạng video thường được dùng cho những website chia sẻ như Youtube hoặc những website chiếu phim trực tuyến, được phát triển bởi Adobe System( Macromedia). Hỗ trợ 2 loại định dạng FLV và F4V. * Định dạng này có chất lượng chấp nhận được, trong khi kích thước tập tin nhỏ, nên rất hiệu quả trong môi trường Internet. Phù hợp với hệ thống trình chiếu video trực tuyến nói chung và hệ thống dạy học bằng video nói riêng. |
| MP4 –  MPEG-4 video | * Định dạng MPEG-4 là một tiêu chuẩn được phát triển bởi Moving Picture Experts Group, được sử dụng trên rất nhiều điện thoại và các thiết bị chơi video (còn gọi là thiết bị chơi MP4). |
| MOV –  Apple QuickTime Movie | * QuickTime Movie là định dạng được Apple phát triển. Cũng là một định dạng thường được sử dụng trên Internet. |
| RM –  Real Media | * Real Media là định dạng được phát triển bởi RealNetworks. Real Media chứa cả thông tin về video (Real video) và âm thanh (Real Audio). Thường được dùng để truyền tải các tập tin đa phương tiện thông qua internet. |
| VOB –  DVD video Object | * Định dạng VOB liên quan đến DVD video Movie File. Một tập tin VOB thường chứa các luồng đa công (multiplex) gồm: video, âm thanh và phụ đề. |
| WMV-  Windows Media video | * Windows Media video là một định dạng tập tin thường gặp nhất cũng được phát triển bởi Microsoft. Tập tin Windows Media chứa video được mã hóa theo bộ codec Windows Media video và âm thanh được mã hóa theo codec Windows Media Audio codec. * Tập tin WMV được thiết kế dùng cho ứng dụng luồng (thread) trên Internet. Dung lượng tập tin WMV khá thấp nên dễ dàng truyền tải trên mạng, tuy nhiên chấp lượng hình ảnh và âm thanh không thực sự tốt. |

*Bảng 2.1 : Các kiểu tập tin video phổ biến*

* 1. **Giải thuật Cache**

Thuật toán bộ nhớ cache (cache algorithms) thường được gọi là giải thuật thay thế (replacement algorithms) hoặc chính sách thay thế (replacement policies).

Thuật toán bộ nhớ cache bao gồm các hướng dẫn tối ưu hóa ( optimizing instruction) và các thuật toán để hướng dẫn các chương trình máy tính và phần cứng duy trì một cấu trúc để quản lý bộ nhớ cache của những thông tin được lưu trữ trên máy tính. Khi bộ nhớ cache đầy, thì giải thuật sẽ chọn lựa thành phần bỏ ra khỏi cache để nhường vị trí cho thành phần mới.

Thời gian tham khảo bộ nhớ trung bình là:

T = m \* Tm + Th + E

Trong đó:

* T = thời gian tham khảo bộ nhớ trung bình
* m = tỷ lệ bỏ lỡ = 1 - (hit tỷ lệ)
* Tm = thời gian cho một truy cập bộ nhớ chính khi có một bỏ lỡ (hoặc với bộ nhớ cache đa cấp thì đó là thời gian tham khảo bộ nhớ trung bình cho bộ nhớ cache cấp thấp hơn tiếp theo)
* Th = độ trễ thời gian để tham khảo bộ nhớ cache khi có hit
* E = hiệu ứng thứ cấp, ví dụ như hiệu ứng xếp hàng trong hệ thống đa tiến trình

Có 2 thành phần chính trong công thức [1] là: độ trễ và tỷ lệ hit (hit rate) của bộ nhớ cache. Ngoài ra còn có một số yếu tố thứ cấp ảnh hưởng đến hiệu năng của bộ nhớ cache.

* Tỉ lệ Hit (Hit rate) mô tả mức độ thường xuyên của việc tìm kiếm mà tìm thấy trong bộ nhớ cache. Chính sách thay thế hiệu quả theo dõi thông tin nào được sử dụng nhiều để cải thiện tỷ lệ hit (cho một kích thước bộ nhớ cache nhất định).
* Độ trễ (Latency) mô tả khoảng thời gian bao lâu sau khi yêu cầu một mục mong muốn, bộ nhớ cache có thể trả lại mục đó (khi có hit). Chiến lược thay thế nhanh hơn sẽ thường theo dõi những thông tin ít ​​sử dụng hoặc trong trường hợp bộ nhớ cache ánh xạ trực tiếp hoặc không có thông tin để làm giảm lượng thời gian cần thiết để cập nhật thông tin đó.
* Mỗi chiến lược thay thế sẽ giải quyết một cách thỏa hiệp 2 thông số này.

Đo tỷ lệ hit thường được thực hiện trên các ứng dụng chuẩn. Trên thực tế, tỷ lệ trúng rất khác nhau trong các ứng dụng khác nhau. Đặc biệt, video và các ứng dụng truyền âm thanh thường có một tỷ lệ trúng gần như bằng không, bởi vì mỗi bit dữ liệu trong dòng được đọc một lần cho thời gian đầu tiên (có bắt buộc), được sử dụng, và sau đó không bao giờ đọc hoặc ghi lại một lần nữa. Thậm chí tệ hơn, nhiều thuật toán bộ nhớ cache (đặc biệt là LRU) cho phép luồng dữ liệu này lấp đầy bộ nhớ cache và đẩy ra các thông tin bộ nhớ cache mà nó sẽ sớm được sử dụng lại (cache pollution).

* + 1. **Giải thuật “Lâu nhất chưa sử dụng” (Least-recently-used LRU)**

Giải thuật “lâu nhất chưa sử dụng” là một trong những giải thuật về bộ đệm (cache) được sử dụng phổ biến. Nó loại bỏ những mục chưa sử dụng lâu nhất. Giải thuật này yêu cầu lưu trữ những gì được sử dụng khi nào, nó rất tốn kém nếu muốn chắc rằng thuật toán luôn luôn loại bỏ những mục chưa sử dụng lâu nhất.

Kỹ thuật chung để hiện thực giải thuật này là dùng một biến tuổi cho khối dữ liệu và theo dõi khối dữ liệu lâu nhất chưa sử dụng dựa trên biến tuổi đó.

Lưu đồ của giải thuật LRU được biểu diễn trong hình 1.



*Hình 2.1 : Flowchart của giải thuật LRU*

Với giải thuật LRU, mỗi tập tin trong bộ đệm cơ sở dữ liệu (cache database) sẽ có một dấu thời gian (time-stamp) được gán giá trị khi đưa tập tin lên bộ đệm hoặc tìm thấy trong bộ đệm. Điều này làm tránh làm đầy bộ nhớ đệm (A) trong thời gian ngắn.

Hiện thực giải thuật LRU không tạo ra một quyết định tại B (tập tin có muốn ở trong bộ nhớ đệm). Nó lựa chọn một tâp tin để loại bỏ tại C bằng cách tìm tập tin cũ nhất trong bộ nhớ đệm sử dụng dấu thời gian.

Mã giả của giải thuật LRU

if FILENAME in cache

value(FILENAME) = timestamp

EXIT

while cachefree < filesize(FILENAME)

find and remove candidate in cache with oldest timestamp

cachefree = cachefree + filesize(candidate)

insert value(FILENAME) = timestamp

* + 1. **Một số giải thuật Cache khác**

*Thuật toán Bélády:* thuật toán bộ nhớ đệm hiệu quả nhất luôn loại bỏ các thông tin không được cần thiết cho thời gian dài nhất trong tương lai. Kết quả tối ưu này thường nói đến thuật toán Bélády hoặc các thuật toán tiên tri . Thông thường, không thể dự đoán thời gian mà các thông tin sẽ được dùng đến trong tương lai, các thuật toán này không thể hiện thực được. Khoảng thời gian thực tế nhỏ nhất chỉ có thể được tính toán bằng thực nghiệm và thông tin này được sử dụng đễ so sánh độ hiệu quả với giải thuật thật sự được chọn.

*Least Recently Used (LRU):* có nghĩa là loại bỏ các mục ít sử dụng gần đây nhất đầu tiên. Thuật toán này đòi hỏi phải theo dõi lưu giữ những gì đã được sử dụng khi nào, rất tốn kém nếu muốn chắc chắn rằng thuật toán luôn luôn loại bỏ các mục được sử dụng gần đây nhất. Hiện thực tổng quát của thuật toán này đòi hỏi phải giữ "bit tuổi" cho bộ nhớ cache-line và theo dõi bộ nhớ cache-line được sử dụng gần đây nhất dựa vào bit tuổi. Với cách tiếp cận này, mỗi lần một dòng bộ nhớ cache được sử dụng, tuổi của tất cả các bộ nhớ cache-line khác cũng thay đổi. LRU là họ của các thuật toán bao gồm 2Q phát minh bởi Theodore Johnson và Dennis Shasha và LRU/K phát minh bởi Pat O'Neil, Betty O'Neil và Gerhard Weikum

*Most Recently Used (MRU):* loại bỏ các phần tử sử dụng nhiều nhất gần đây đầu tiên. Khi một tập tin được quét nhiều lần trong một mô hình tham chiếu, MRU là thuật toán thay thế tốt nhất. Với các mẫu truy cập ngẫu nhiên và quét lặp đi lặp lại trên bộ dữ liệu lớn (đôi khi được gọi là các mẫu truy cập theo chu kỳ) MRU có số truy cập nhiều hơn LRU do xu hướng của họ để giữ lại dữ liệu cũ. MRU rất hữu ích trong các tình huống mà tồn tại người ta thường xuyên truy cập vào các mục cũ.

*Pseudo-LRU:* Đối với bộ nhớ cache có sự kết hợp lớn (hơn 4 đường). Với một mô hình sử dụng với hình thức loại bỏ những mục ít sử dụng nhất gần đây, giải thuật PLRU có thể sử dụng để giải quyết 1 phần cache làm việc. Có thể cache cho vị trí trên bộ nhớ.

*Random Replacement (RR):* chọn và loại bỏ 1 mục trong cache để lấy không gian cho mục khác một cách ngẫu nhiên. Thuật toán này không yêu cầu lưu trữ bất kỳ thông tin nào về lịch sử truy cập.

*Segmented LRU:* Một bộ nhớ cache được chia thành hai phân đoạn (khi PLRU quá chậm). Một đoạn tập sự( probationary segment) và một đoạn bảo vệ. Mỗi dòng trong một đoạn được sắp xếp theo thứ tự sử dụng nhiều nhất gần đây. Dữ liệu bị bỏ lỡ được thêm vào cuối phân đoạn tập sự.

*2-way set associative:* cho những CPU tốc độ cao. Địa chỉ của 1 mục được sử dụng để tính toán vị trí cho phép đi trong bộ nhớ cache. Nó đòi hỏi 1 bit trên 1 cặp cache-line để cho biết ít sử dụng gần đây.

*Direct-mapped cache:* cho những CPU tốc độ cao nhất (khi 2-way quá chậm). Các địa chỉ của những mục mới được sử dụng để tính toán vị trí trong bộ nhớ cache, nơi nó được phép đi.

*Least Frequently Used (LFU):* đếm số lần sử dụng của mỗi mục cần thiết. Loại bỏ đi những mục ít sử dụng thường xuyên.

*Adaptive Replacement Cache (ARC):* cân bằng liên tục giữa LRU và LFU và cải thiện kết quả sự kết hợp này. ARC cải thiện SLRU bằng cách sử dụng thông tin về thời gian loại bỏ một mục trong bộ nhớ cache để tự động điều chỉnh kích thước của đoạn bảo vệ và tập sự, hỗ trợ làm việc tốt nhất trong một không gian bộ nhớ cache có sẵn xác định.

*Clock with Adaptive Replacement (CAR):* kết hợp ARC với CLOCK/ CAR có hiện suất tối hơn ARC, LRU và CLOCK.

## Công nghệ

* 1. **Java**
* Java:
  + Java là một công nghệ xây dựng các ứng dụng phần mềm được coi là công nghệ mang tính cách mạng và khả thi nhất trong việc tạo ra các ứng dụng có khả năng chạy thống nhất trên nhiều nền tảng mà chỉ cần biên dịch một lần. Java vừa là ngôn ngữ lập trình được giới thiệu bởi Sun MicroSystems, còn là một nền tảng phát triển và triển khai ứng dụng trong đó máy ảo Java, bộ thông dịch có vai trò trung tâm. Java được xây dựng trên nền tảng của C và C++. Do vậy nó sử dụng các cú pháp của C và các đặc trưng hướng đối tượng của C++. Java là ngôn ngữ lập trình hướng đối tượng. Java là ngôn ngữ vừa biên dịch vừa thông dịch. Đầu tiên mã nguồn được biên dịch bằng công cụ JAVAC để chuyển thành dạng ByteCode(file class). Sau đó được thực thi trên từng loại máy cụ thể nhờ chương trình thông dịch. Có nghĩa là khi Compile một chương trình Java bạn sẽ có một tập tin lớp(file class) và bạn không thể chạy nó bằng cách nhấp đôi vào nó được mà bạn phải dùng trình thông dịch(Interpreter) Java để chạy.Mục tiêu của các nhà thiết kế Java là cho phép người lập trình viết chương trình một lần nhưng có thể chạy trên bất cứ phần cứng cụ thể. Ngày nay, Java được sử dụng rộng rãi để viết chương trình chạy trên Internet. Nó là ngôn ngữ lập trình hướng đối tượng độc lập thiết bị, không phụ thuộc vào hệ điều hành.
  + Java chia làm ba bộ phận: J2SE, J2EE, J2ME gồm các đặc tả, công cụ, API để phát triển các ứng dụng.
  + Java có các đặc trưng: đơn giản, hướng đối tượng, độc lập với phần cứng và hệ điều hành, bảo mật, đa luồng, phân tán, động. Vì vậy, Java là một ngôn ngữ được rất nhiều người sử dụng hiện nay.
* Các kiểu chương trình Java:
  + Applets: Đây là chương trình chạy trên Internet thông qua các trình duyệt hỗ trợ Java như IE hay Netscape. Bạn có thể dùng các công cụ của Java để xây dựng Applet. Applet được nhúng bên trong trang Web hoặc tập tin HTML. Khi trang Web hiển thị trong trình duyệt, Applet sẽ được nạp và thực thi.
  + Ứng dụng thực thi qua dòng lệnh: Các chương trình này chạy từ dấu nhắc lệnh và không sử dụng giao diện đồ họa. Các thông tin nhập xuất được thể hiện tại dấu nhắc lệnh.
  + Ứng dụng đồ họa: Đây là các chương trình Java chạy độc lập cho phép người dùng tương tác qua giao diện đồ họa.
  + Servlet: Java thích hợp để phát triển ứng dụng nhiều lớp. Applet là chương trình đồ họa chạy trên trình duyệt tại máy trạm. Ở các ứng dụng Web, máy trạm gửi yêu cầu tới máy chủ. Máy chủ xử lý và gửi ngược kết quả trở lại máy trạm. Các chương trình Java API chạy trên máy chủ giám sát các quá trình tại máy chủ và trả lời các yêu cầu của máy trạm. Các chương trình Java API chạy trên máy chủ này mở rộng khả năng của các ứng dụng Java API chuẩn. Các ứng dụng trên máy chủ này được gọi là các Servlet. hoặc Applet tại máy chủ. Các xử lý trên Form của HTML là cách sử dụng đơn giản nhất của Servlet. Chúng còn có thể được dùng để xử lý dữ liệu, thực thi các transaction và thường được thực thi qua máy chủ Web.
  + Ứng dụng cơ sở dữ liệu: Các ứng dụng này sử dụng JDBC API để kết nối tới cơ sở dữ liệu. Chúng có thể là Applet hay ứng dụng, nhưng Applet bị giới hạn bởi tính bảo mật.
* Máy ảo Java (JVM- Java Virtual Machine ): Máy ảo là một phần mềm dựa trên cơ sở máy tính ảo. Nó có tập hợp các lệnh logic để xác định các hoạt động của máy tính. Người ta có thể xem nó như một hệ điều hành thu nhỏ. Nó thiết lập các lớp trừu tượng cho: Phần cứng bên dưới, hệ điều hành, mã đã biên dịch. Trình biên dịch chuyển mã nguồn thành tập các lệnh của máy ảo mà không phụ thuộc vào phần cứng cụ thể. Trình thông dịch trên mỗi máy sẽ chuyển tập lệnh này thành chương trình thực thi.
* Bộ công cụ phát triển JDK (Java Development Kit) bao gồm: Java Plug-In, chúng cho phép chạy trực tiếp Java Applet hay JavaBean bằng cách dùng JRE thay cho sử dụng môi trường thực thi mặc định của trình duyệt. JDK chứa các công cụ:
  + Trình biên dịch javac
  + Trình thông dịch java
  + Trình dịch ngược javap
  + Công cụ sinh tài liệu javadoc
  + Chương trình tìm lỗi (debug) jdb
* Môi trường phát triển tích hợp - Integrated Development Environment(IDE):
  + IDE lfà một loại phần mềm máy tính có công dụng giúp đỡ các lập trình viên trong việc phát triển phần mềm.
  + IDE thường bao gồm:
    - Một trình soạn thảo mã (source code editor): dùng để viết mã.
    - Trình biên dịch (compiler) và/hoặc trình thông dịch (interpreter).
    - Công cụ xây dựng tự động: khi sử dụng sẽ biên dịch (hoặc thông dịch) mã nguồn, thực hiện liên kết (linking), và có thể chạy chương trình một cách tự động.
    - Trình gỡ lỗi (debugger): hỗ trợ dò tìm lỗi.
    - Ngoài ra, còn có thể bao gồm hệ thống quản lí phiên bản và các công cụ nhằm đơn giản hóa công việc xây dựng giao diện người dùng đồ họa (GUI).
    - Nhiều môi trường phát triển hợp nhất hiện đại còn tích hợp trình duyệt lớp (class browser), trình quản lí đối tượng (object inspector), lược đồ phân cấp lớp (class hierarchy diagram),... để sử dụng trong việc phát triển phần mềm theo hướng đối tượng.
  + Một số IDE như: Eclipse, NetBean…
* Để bắt đầu phát triển ứng dụng Java ta cần chuẩn bị môi trường lập trình: cài đặt JDK, IDE, thiết lập biến môi trường. Cũng như tìm hiểu kiến thức về lập trình hướng đối tượng. Java là ngôn ngữ phổ biến hiện nay, nên tài liệu hỗ trợ cho người lập trình cũng rất đa dạng và đầy đủ.

**LinkedHashMap**

Trong khoa học máy tính, bảng băm là một cấu trúc dữ liệu sử dụng hàm băm để ánh xạ từ giá trị xác định, được gọi là khóa (ví dụ như tên của một người), đến giá trị tương ứng (ví dụ như số điện thoại của họ). Do đó, bảng băm là một mảng kết hợp. Hàm băm được sử dụng để chuyển đổi từ khóa thành chỉ số (giá trị băm) trong mảng lưu trữ các giá trị tìm kiếm.

Trong trường hợp lý tưởng, hàm băm luôn chuyển đổi các khóa khác nhau đến các chỉ số khác nhau. Tuy nhiên trong thực tế, điều này hiếm khi xảy ra (trừ khi các khóa là cố định: không có thêm khóa mới nào được bổ sung vào bảng sau khi tạo bảng). Thay vào đó, hầu hết các thiết kế bảng băm đều giả sử các khóa khác nhau có thể được băm vào cùng một giá trị (gọi là va chạm băm), và cung cấp cách giải quyết va chạm.

Trong một bảng băm có kích thước đủ lớn, thời gian trung bình cho mỗi lần tra cứu là độc lập với số lượng phần tử trong bảng. Nhiều thiết kế bảng băm cũng cho phép chèn thêm và xóa bỏ tùy ý của cặp khóa-giá trị trong thời gian trung bình (hoặc trừ dần) không đổi cho mỗi thao tác.[1][2]

Trong nhiều trường hợp, bảng băm có hiệu quả hơn so với cây tìm kiếm hoặc bất kỳ cấu trúc dữ liệu tìm kiếm nào. Vì lý do này, chúng được sử dụng rộng rãi trong nhiều loại phần mềm máy tính, đặc biệt là cho mảng kết hợp, lập chỉ mục cơ sở dữ liệu, tổ chức bộ nhớ đệm, và cấu trúc dữ liệu tập hợp.

* 1. **FFmpeg**

FFmpeg là hệ thống đa phương tiện mã nguồn mở hàng đầu, để giải mã, mã hóa, chuyển mã, mux, demux, stream, lọc và chơi được khá nhiều những tập tin giải trí số mà con người và máy móc đã tạo ra. Nó hỗ trợ từ những định dạng xưa nhất đến những định dạng hiện đại nhất hiện nay kể cả nó được thiết kế bởi một số ủy ban tiêu chuẩn, cộng đồng hoặc tổ chức nào đó. Nó chứa libavcodec, libavutil, libavformat, libavdevice, libswscale và libswresample có thể được sử dụng bởi các ứng dụng khác như là một thư viện của hệ thống. Cũng như ffmpeg, ffserver, ffplay và ffprobe có thể được sử dụng bởi người dùng cuối để chuyển mã, stream và chạy tập tin.

FFmpeg cố gắng cung cấp các giải pháp tốt nhất về mặt kỹ thuật cho các nhà phát triển ứng dụng và người dùng cuối. Để đạt được điều này, FFmpeg kết hợp các tùy chọn của các phần mềm miễn phí tốt nhất hiện có. FFmpeg cố gắng để giữ sự phụ thuộc vào thư viện khác thấp nhất có thể và để tối đa hóa việc chia sẻ mã giữa các phần của FFmpeg. Bất cứ nơi nào không thể trả lời câu hỏi về "tốt nhất" FFmpeg hỗ trợ cả hai (hoặc nhiều) lựa chọn để người dùng có thể lựa chọn.

# CHƯƠNG III: GIẢI QUYẾT BÀI TOÁN

## Phân tích và thiết kế hệ thống

1. 1. **Mô hình tổng quan**

Hiện nay, chúng ta có rất nhiều hệ thống Website hỗ trợ những bài giảng trực tuyến. Elearning là một trong những ứng dụng khá điển hình đã được xây dựng và vận hành trên Internet. Với mô hình đào tạo này, dữ liệu bài giảng trực tuyến thường là Slide, video bài giảng, hướng dẫn được lưu trữ trên server, thông tin về bài giảng được lưu trữ trong cơ sở dữ liệu và được hiện thị theo dạng danh sách trên Website đào tạo trực tuyến.

Người sử dụng truy cập vào Website, lựa chọn bài giảng trong danh sách các bài giảng đó và bắt đầu học tập theo chương trình đào tạo trực tuyến.

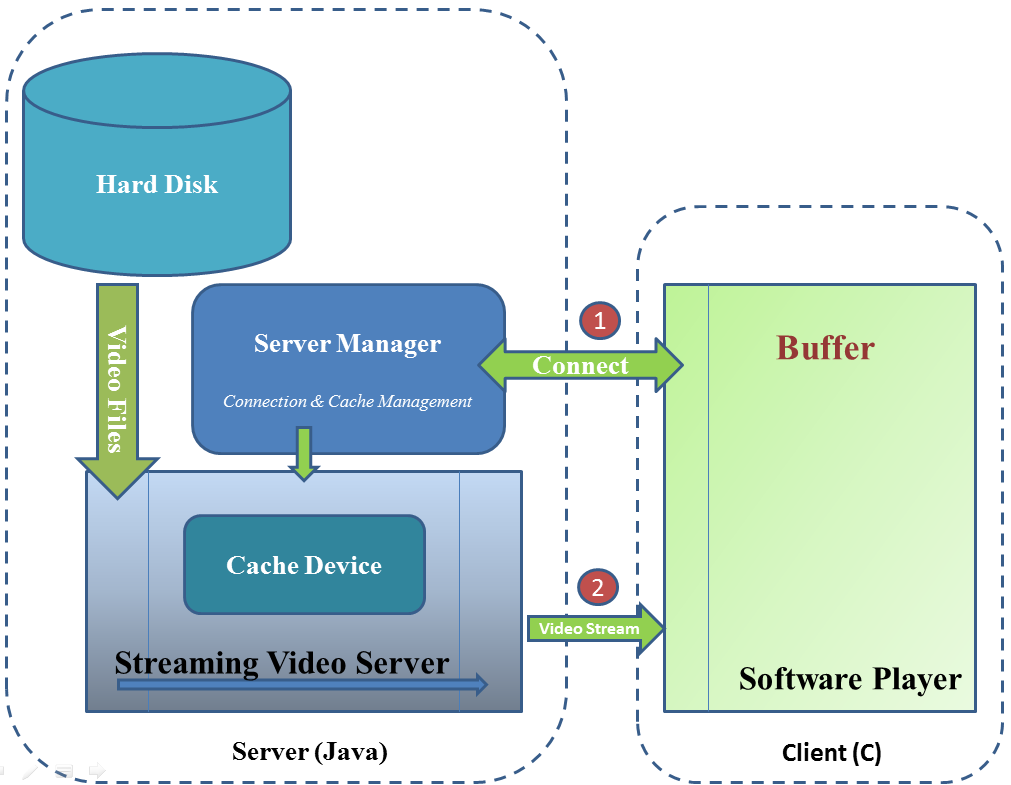
* 1. **Xác lập tính năng hệ thống**

Như đã giới thiệu ở trên, tầm vực của đề tài không phải là thiết kế toàn bộ hệ thống trình chiếu video, cũng như các chức năng quản lý cơ sở dữ liệu và quản lý chức năng của người dùng mà chỉ tập trung vào các chức năng sau:

Những chức năng quản lý kết nối, quản lý tiến trình Streaming video, quản lý Cache trên một Server có hỗ trợ video caching để tăng tốc độ truyền tải video đến người dùng.

Với việc lựa chọn những công nghệ phát triển khác nhau, tùy thuộc vào player hỗ trợ ta có thể phát triển software hoặc plug-in để quản lý dữ liệu bên phía người dùng.

* 1. **Kiến trúc hệ thống**



*Hình 3.1: Kiến trúc hệ thống*

Hệ thống trong hình gồm 2 thành phần:

* + - * Server(S):
        + Server là một máy chủ gồm có: Hard disk (D) chứa dữ liệu, và thành phần quản lý máy chủ(SM) quản lý quá trình kết nối của các máy trạm (thông qua một cổng nhất định).
        + Hard Disk(D): Lưu trữ tập tin video của nhà cung cấp.
        + Server Manager(SM):

Tạo Server với port được lựa chọn, chờ người dùng kết nối tới.

Quản lý người dùng kết nối dựa trên IP.

Quản lý những tập tin video đã được đưa vào cache.

Quản lý việc lựa chọn gửi phần nào của tập tin video về cho người dùng.

* + - * + Streaming video Server (SV):

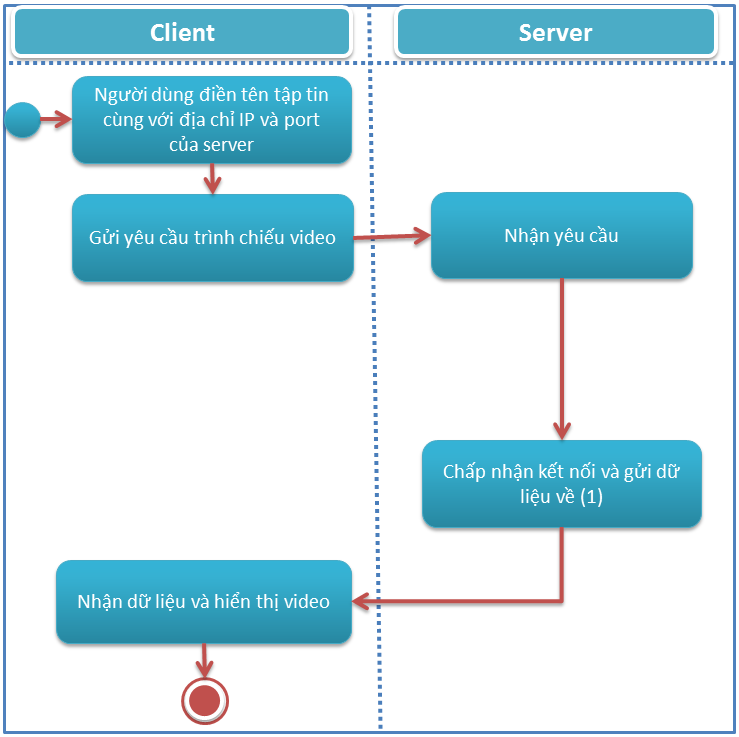
Khi nhận được yêu cầu từ phía người dùng chạy một gói (packet) của một tập tin video nào đó, Streaming video Server tìm gói đó ở trong Cache Device, nếu nó đã có trong Cache Device thì sẽ được gởi về cho người dùng, trong trường hợp chưa có trong cache thì Streaming video Server sẽ tìm gói đó dưới ổ cứng (D)và đẩy vào Caching Devide.

* + - * Client (C):
        + Điền tên tập tin video cần chạy cùng với địa chỉ IP và cổng của server để thực hiện việc gởi yêu cầu đối với máy chủ (1’).
        + Nhận một số thông tin cơ bản của tập tin và bắt đầu gởi yêu cầu về dữ liệu của tập tin.
        + Player software nhận những gói (2’) dữ liệu hiển thị video lên cho người dùng.

**Tóm lại:**

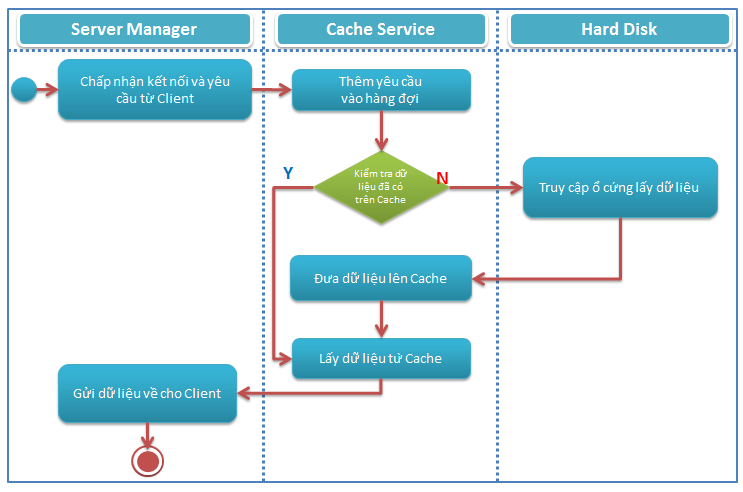
* + - * Server: lưu trữ tập tin video, gởi tập tin video được yêu cầu về người dùng.
      * Client: kết nối đến server và thực hiện việc truyền nhận các gói dữ liệu của tập tin video cần chạy

**Mô hình hoạt động tổng quát của hệ thống:**



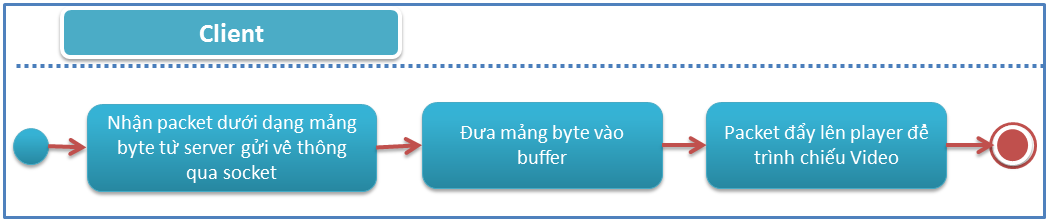
*Hình 3.2: Lược đồ hoạt động tổng quát*

**Từ (1) ta có mô hình hoạt động chi tiết bên phía Server**



*Hình 3.3: Lược đồ hoạt động trên Server*

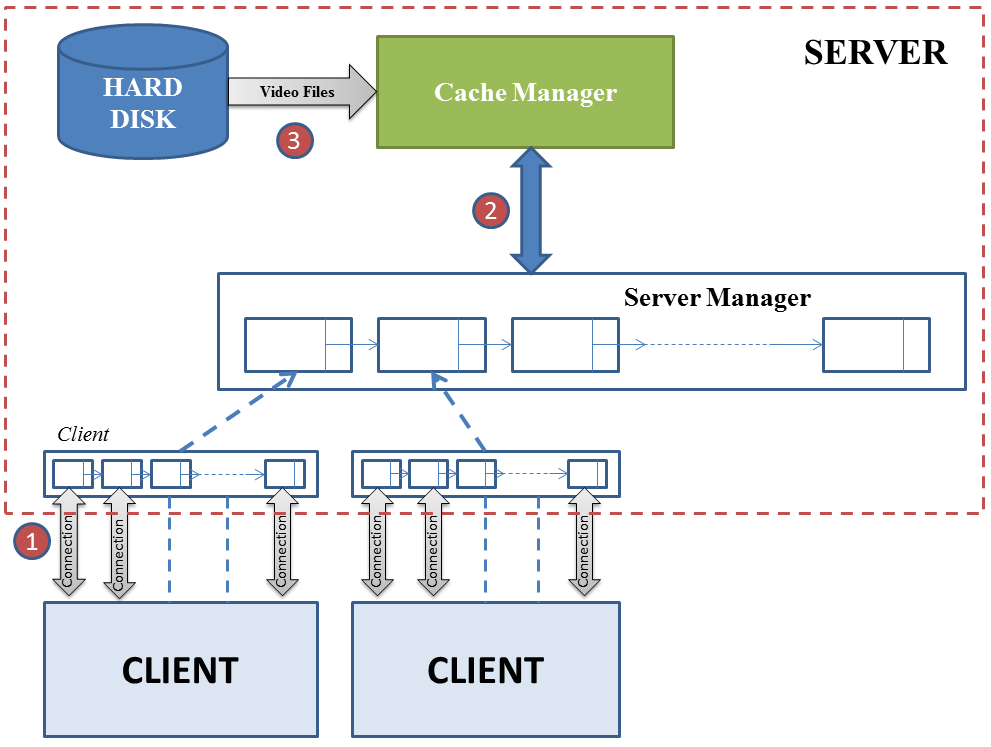
**Từ (1) ta có mô hình hoạt động về phía client**



*Hình 3.4 : Lược đồ hoạt động quản lý Buffer phía Client*

## Giải pháp kỹ thuật

2. 1. **Server**



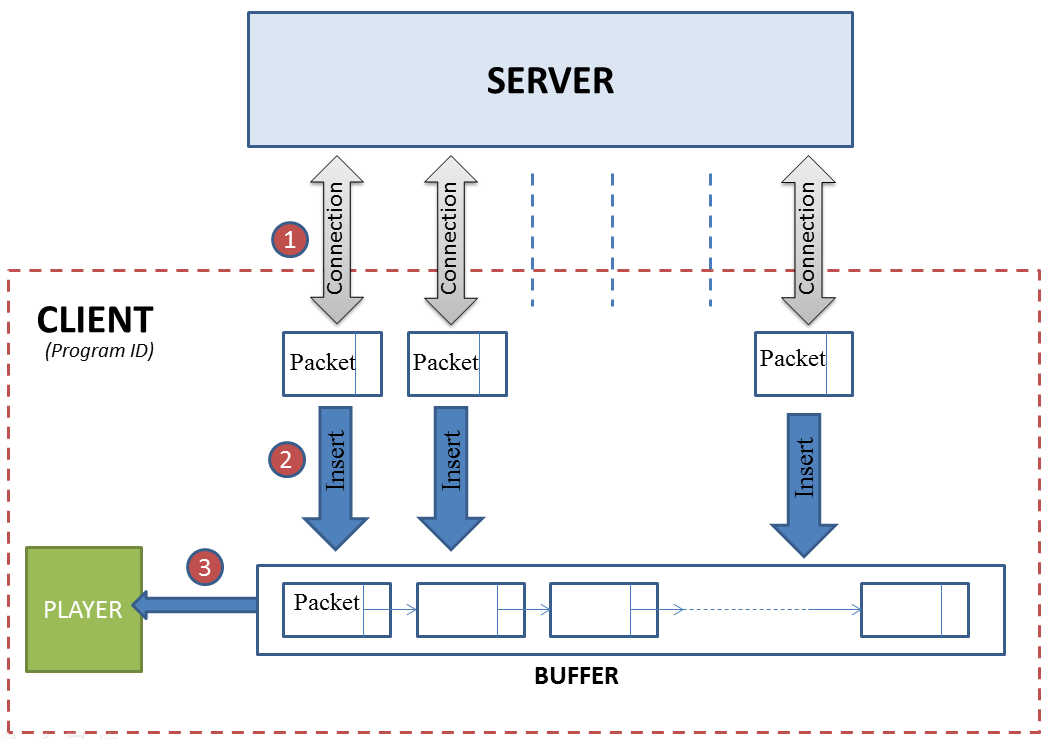
*Hình 3.5 : Giải pháp kỹ thuật ở Server*

Server Manager là phần quản lý chính của toàn bộ server. No quản lý danh sách các client đang kết nối với server, và mỗi client sẽ quản lý những kết nối của client đó.

Thông qua Connection (1), Server Manager nhận yêu cầu từ Client và đưa qua Cache Manager (2).

Cache Manager nhận yêu cầu và phân tích yêu cầu đó, tiếp đến Cache Magager sẽ tìm kiếm gói dữ liệu được yêu cầu trên cache (được hiện thực là một bản đồ băm (hash map)), nếu gói dữ liệu đó không có sẵn trên cache thì Cache Manager sẽ lấy(2) từ Hard Disk lên để trả về cho Client. Sau khi yêu cầu được đáp ứng thì Cache Manager sẽ dựa trên yêu cầu đó đưa lên cache một số gói dữ liệu tiếp theo của gói được yêu cầu

* 1. **Client**



*Hình 3.6 : Giải pháp kỹ thuật ở Client*

Mỗi client sẽ có một ProgramID để server có thể quản lý các kết nối đến server thuộc về client nào và nó sẽ được nhận về từ server cho kết nối đầu tiên của client đó.

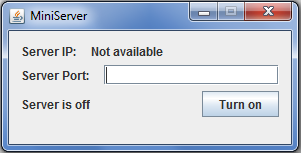
Mỗi Connection (1) là một thread, làm nhiệm vụ gửi yều cầu và nhận gói dữ liệu. Trong đó mỗi yêu cầu đến server sẽ bao gồm ProgramID, fileID và packetID. Gói dữ liệu sau khi nhận về sẽ được chèn (2) vào buffer dựa vào packetID của gói dữ liệu đó.

Thread chính của chương trình (Main Thread) sẽ lấy gói dữ liệu đầu tiền của buffer (3) để giải mã và hiển thị lên Player.

## Thực hiện

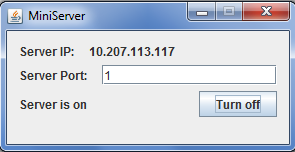
1. 1. **Server**
      1. **Quản lý chờ - tạo kết nối mới**

* Tập tin thực hiện: ServerControl.java
* Tập tin liên kết: MiniServerSkin.java, Client.java, Connection.java
* Thực hiện chức năng chờ client kết nối đến, kiểm tra sự hợp lệ và chấp nhận kết nối. Sau đó tạo ra mỗi thread tương ứng với mỗi kết nối đến
* Kiểm tra kết nối đến thuộc client nào thông qua thông số kết nối đó gửi lên. Nếu là client mới thì tạo ra 1 thread mới để quản lý các kết nối của nó. Nếu là client đã có và số kết nối của client đó chưa vượt số lượng cho phép thì thêm kết nối mới vào client đó. Ngược lại hủy kết nối.
  + 1. **Quản lý client cụ thể:**
* Tập tin thực hiện: Client.java
* Tập tin liên kết: Connection.java, ServerControl.java, RequestHandle.java, MoviePakage.java
* Thực hiện quản lý các kết nối từ client thông qua các tác vụ thêm, xóa kết nối.
* Cache tạm thời dữ liệu được hệ thống trả về để chờ gởi xuống client
* Thực hiện chức năng đa luồng trên server, nhằm đáp ứng số lượng kết nối lớn tới server. Mỗi thread ứng với mỗi client đang kết nối đến server.
  + 1. **Quản lý kết nối cụ thể:**
* Tập tin thực hiện: Connection.java
* Tập tin liên kết: ServerControl.java, Client.java, RequestHandle.java, MoviePakage.java
* Nhận yêu cầu dữ liệu của client, tạo tác vụ mới (Request) và thêm vào danh sách các tác vụ chờ xử lý.
* Nhận dữ liệu từ thread quản lý client, kiểm tra hợp lệ và gởi về cho client.
* Xác nhận dữ liệu đã gởi xong và chờ dữ liệu tiếp theo.
* Kiểm tra kết nối còn sống (alive) hay không, nếu kết nối bị ngắt, xóa toàn bộ tác vụ của kết nối đang chờ được xử lý, xóa toàn bộ các dữ liệu đang chờ gửi về client, xóa thread quản lý kết nối khỏi danh sách các thread của Client. Sau đó gọi hàm hủy thread.
  + 1. **Quản lý các dịch vụ đa luồng trên server**
* Tập tin thực hiện: ServerControl.java, CacheService.java, RequestProcess.java
* Tập tin liên kết: Requesthandle.java, MoviePakage.java, FileManager.java, FileStreamHandle.java, FileInfo.java
* Các thread con (CacheService, RequestProcess) được tạo ra và quản lý bởi 2 danh sách liên kết trên Server, Server nắm quyền tạo mới, hủy trên danh sách này.
* Tất cả các thread hoạt động độc lập dưới sự quản lý chung của Server, có các giải thuật để đồng bộ, đảm bảo giảm thiểu xung đột ở mức tối đa.
* RequestProcess quản lý toàn bộ các yêu cầu chờ xử lý, chuyển dần xuống cache để xử lý dữ liệu tuần tự, tránh tình trạng quá tải khi phải nhận cùng lúc quá nhiều yêu cầu. Hiện thực việc loại bỏ các yêu cầu trùng lặp.
* CacheService quản lý bộ đệm (Cache Database), cung cấp cho hệ thống 1 tác vụ truy cập duy nhất là getCachePkt. Trả về pkt nếu nó đã được cache lên bộ đệm hoặc đọc từ ổ cứng nếu pkt đó chưa được cache. Tự sinh các yêu cầu ngay sau yêu cầu đang xử lý để “cache trước” dữ liệu, tận dụng khả năng đọc file liên tục của hệ thống.
  + 1. **Quản lý đọc, cache tập tin trên hệ thống**
* Tập tin thực hiện: CacheService.java
* Tập tin liên kết: RequestHandle.java, MoviePakage.java, FileManager.java, FileStreamHandle.java, FileInfo.java
* Thực hiện chức năng đọc tập tin video từ đĩa cứng lên bộ đệm và trả về cho các tác vụ gọi nó.
* Hỗ trợ chức năng truy cập ngẫu nhiên (Random Access) cho các tác vụ yêu cầu đọc file ở các vị trí khác nhau.
* Quản lý các con trỏ giữ FileStream và thông tin file trên hệ thống.
* Hiện thực các giải thuật quản lý cache, cache trước dữ liệu để tăng tối đa hiệu suất của hệ thống.
  1. **Client**
     1. **Gửi yêu cầu và nhận dữ liệu từ server**
* Tập tin thực hiện: Connection.c
* Tập tin liên kết: Connection.h, myPlayer.c
* Thực hiện chức năng kết nối với server.
* Gửi thông tin về tập tin video cần xem lên server.
* Nhận dữ liệu từ socket và đưa vào buffer.
  + 1. **Đồng bộ các kết nối**
* Tập tin thực hiện : Mutex.c
* Tập tin liên kết : Mutex.h, myPlayer.c
* Thực hiện nhiệm vụ đồng bộ giữa các thread kết nối đến server, tránh trường hợp bị xung đột dữ liệu
  + 1. **Player**
* Tập tin thực hiện: myPlayer.c
* Tập tin liên kết: Connection.c, cmdutils.c
* Quản lý đa kết nối đến server.
* Nhận dữ liệu từ server và đưa vào thư viện libavformat.
* Thực hiện các chức năng của player như phát, dừng, phát lại tập tin video; tăng, giảm âm thanh.
  + 1. **Thư viện Ffmpeg**
* FFmpeg cung cấp các công cụ khác nhau:
  + Ffmpeg là một công cụ dòng lệnh để chuyển đổi định dạng các tập tin đa phương tiện.
  + Ffplay là một trình phát các tập tin đa phương tiện đơn giản dựa trên SDL và thư viện FFmpeg.
* Và các thư viện:
  + libavutil là một thư viện chứa các hàm lập trình đơn giản, bao gồm cả việc tạo số ngẫu nhiên, các cấu trúc dữ liệu, các công thức toán học, phần cốt lõi của các tiện ích đa phương tiện, và nhiều hơn nữa.
  + libavcodec là một thư viện chứa các bộ giải mã và mã hóa cho các bảng mã (codec) âm thanh và video.
  + libavformat là một thư viện có chứa demuxers và muxers cho các định dạng container đa phương tiện.
  + libavdevice là một thư viện chứa các thiết bị đầu vào và đầu ra cho việc lưu và hiển thị trên các khung (framework) phần mềm đa phương tiện phổ biến, bao gồm cả video4Linux, video4Linux2, VFW, và ALSA.
  + libavfilter là một thư viện chứa các bộ lọc.
  + libswscale là một thư viện tối ưu hóa việc chuyển đổi kích cỡ ảnh, màu sắc, định dạng của các điểm ảnh (pixel).
  + libswresample là một thư viện thực hiện tối ưu hóa cao trong resampling, rematrixing âm thanh và các thao tác chuyển đổi định dạng âm thanh.
* Phát triển thêm:
  + processInput.c, avformat.h: quản lý bộ nhớ đệm của chương trình.
  + file.c: đưa dữ liệu vào bộ nhớ đệm để chạy.
  + avformat-52.def: đưa dữ liệu vào và ra các thư viện động.
  1. **Giao diện**
     1. **Server**
* Khi khởi động chương trình dành cho server thì khung cửa sổ của MiniServer sẽ hiện ra cho người quản trị server chọn port để chờ kết nối.



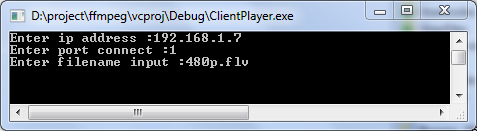
*Hình 3.7 : Giao diện Server khi khởi động chương trình*

* Sau khi Turn On thì server sẽ bắt đầu chạy và chờ client kết nối đến.



*Hình 3.8 : Giao diện Server khi Server “turn on”*

* + 1. **Client**
* Chương trình sẽ hiện ra giao diện console để người dùng nhập địa chỉ IP, port của server và tên tập tin video cần chạy



*Hình 3.9 : Giao diện Server khi khởi động chương trình*

* Sau khi kết nối với server, client sẽ nhận dữ liệu về tập tin video và phát cho người dùng xem.



*Hình 3.10 : Giao diện của Player*

# CHƯƠNG IV: KẾT QUẢ

## Những việc đã làm được

Máy chủ:

* Hiện thực được quy trình quản lý các tác vụ của server.
* Hiện thực được hệ thống xử lý đa luồng (multithread).
* Hiện thực được quy trình quản lý các máy khách truy cập.
* Hiện thực được quy trình quản lý các kết nối từ máy khách.
* Hiện thực được quy trình quản lý bộ đệm dữ liệu.
* Hiện thực được các tác vụ truy xuất file (tuần tự và ngẫu nhiên) trên hệ thống.
* Xây dựng được quy trình quản lý file, quản lý các tác vụ thao tác với database sau này (nếu được phát triển tiếp).

Máy khách:

* Chạy được tập tin video với dữ liệu từ bộ nhớ đệm.
* Tạo kết nối với máy chủ để gởi yêu cầu và nhận dữ liệu
* Xử lý được đa kết nối (multi connection) đến máy chủ để tăng tốc độ truyện nhận dữ liệu.

## Hạn chế

Máy chủ:

* Giao diện quản lý máy chủ còn thô sơ, chưa điều khiển được các tính năng cao cấp của hệ thống.
* Chưa hiện thực được hệ thống cơ sở dữ liệu quản lý file trên hệ thống.
* Chưa thực hiện được hệ thống cache đa tầng (multilevel-cache).
* Chưa xử lý các vấn đề liên quan đến bảo mật server (giải thuật chống hack, DDos).
* Chưa hiện thực được vấn đề truyền tải các phiên bản có chất lượng khác nhau cho các máy khách có chất lượng đường truyền khác nhau.
* Chưa hiện thực được cơ chế nén và mã hóa dữ liệu.

Máy khách:

* Chưa phát triển được phần giao diện cho máy khách
* Chưa kiểm tra được chất lượng dữ liệu nhận về.
* Chưa thay đổi được số lượng kết nối đến máy chủ trong khi đang chạy video.
* Chưa hiện thực được cơ chế giải nén và giải mã.

## Kiến thức thu được:

* Lập trình socket trên java và C.
* Lập trình multithread, các vấn đề về quản lý và đồng bộ trong java và C.
* Đọc, xử lý file trên java.
* Các nguyên lý, giải thuật cache.
* Tìm hiểu được kiến thức về cấu trúc file video.
* Tìm hiểu được kiến thức về giải mã video.
* Hiểu về cơ chế streaming video.

## Đánh giá hiệu suất:

1. 1. **Server**

*Hình 4.1 : Biểu đồ theo dõi tốc độ xử lý dữ liệu của Server*

*Hình 4.2 : Biểu đồ so sánh tốc độ xử lý dữ liệu trong 2 trường hợp có cache*

*và không có cache*

*Hình 4.3 : Biểu đồ theo dõi hiệu suất cache*

Mô tả biểu đồ:

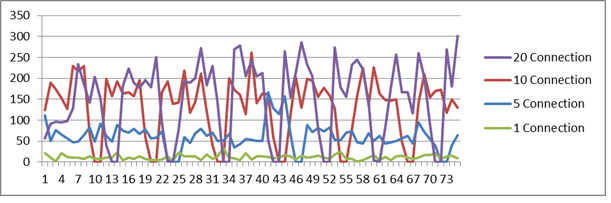
* Trục tung là số lượng video packet được gởi đi trong 1 giây
* Trục hoành là thời gian, tính bằng giây

Điều kiện thực hiện phép đo:

* Server phải phục vụ cho 25 máy khác chạy liên tục trong mạng LAN (100Mbps)
* Mỗi máy khách mở 10 kết nối đến server
* Server được cấu hình với kích thước Cache là 20.000 gói (tương đương 625MB)
* File được dùng để thử nghiệm có kích thước 2GB

Nhận xét kết quả thu được:

* Hiệu suất server chưa đạt mức tối đa do giới hạn đường truyện mạng LAN chỉ có 100Mbps (~10MB/s)
* Trong quá trình khởi tạo kết nối ban đầu, khi server đang khởi tạo Cache, tốc độ trả dữ liệu của server chậm hơn khi không sử dụng cache.
* Sau khi cache đã khởi tạo xong, tốc độ trả dữ liệu của server nhanh hơn hẳn và giữ ổn định hơn.
* Khi sử dụng Cache, số lần hệ thống phải đọc dữ liệu từ ổ cứng rất ít.
  1. **Client**



*Hình 4.4 : Biểu đồ theo dõi tốc độ nhận dữ liệu của Client*

Mô tả biểu đồ:

* Trục tung là số lượng gói dữ liệu.
* Trục hoành là thời gian.

Biểu đồ trên được đo trong điều kiện:

* Server có Cache
* Số lượng gói dữ liệu trong buffer tối đa là 1500(~47 mb).
* Dung lượng file yêu cầu là 2GB

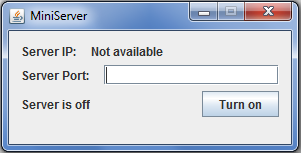
Từ biếu đồ ta có những nhận xét sau:

* Khi tăng số Connection từ 1 lên 5 rồi lên 10 thì số lượng gói dữ liệu tải về trên 1 giây sẽ tăng gần gấp đôi. Nhưng khi tăng từ 10 lên 20 Connection thì tỉ lệ gói dữ liệu tải về không đáng kể. Điều này có nghĩa là không phải càng nhiều Connection thì càng tốt, nó sẽ bị giới hạn ở một mức nào đó.
* Khi buffer bị đầy thì những Connection sẽ bị ngủ nên có sẽ những đoạn số lượng gói dữ liệu bằng 0.

## Hướng dẫn cài đặt và sử dụng

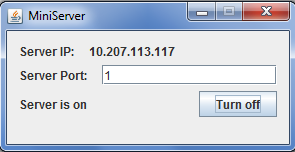
1. 1. **Cài đặt**
      1. **Server**

* Người sử dụng phải có Java trên hệ thống (có thể tải về từ địa chỉ : <http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
* Modem phải được mở port (port mà người sử dụng dành cho chương trình). Hướng dẫn mở port xem trong phần phụ lục.
* Muốn cache tập tin video nào, người dùng phải để đặt đường dẫn đến tập tin video đó trong tập tin ListFileOnServer.txt
  + 1. **Client**
* Người sử dụng double vào file “Client Player.msi”để tiến hành cài đặt chương trình. Hướng dẫn cài đặt xem trong phần phụ lục.
  1. **Sử dụng**
     1. **Server**
* Người dùng double click vào tập tin MovieServerJava.jar để khởi động chương trình.



*Hình 4.5 : Giao diện Server khi khởi động chương trình*

* Người dùng điền port đã được mở vào để bắt đầu chạy Server.



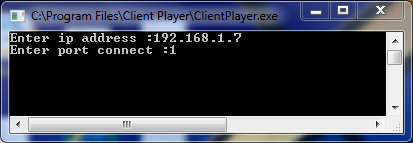
*Hình 4.6 : Giao diện Server khi Server “turn on”*

* + 1. **Client**
* Người dùng click vào shourtcut của chương trình ở màn hình desktop để khởi động chương trình.



*Hình 4.7 : Shortcut của chương trình*

* Sau đó điền địa chỉ IP và port muốn kết nối đến và nhấn Enter.



*Hình 4.8 : Giao diện Client khi khởi động chương trình*

* Trong danh sách file hiện ra, chọn ID của tập tin video mà người dùng muốn chạy và nhấn Enter. Cửa sổ player sẽ hiện ra để trình chiếu tập tin video mà người dùng chọn.



*Hình 4.9 : Giao diện của Player*

# CHƯƠNG V: TỔNG KẾT

## Hiệu quả giáo dục - kinh tế - xã hội

* 1. **Đối với giáo dục và đào tạo**

Chương trình đào tạo từ xa đang là một chương trình đào tạo đang được các trường đại học quan tâm, đăc biệt trong đó khoa KH&KT Máy tính chú trọng đầu tư và phát triển mạnh. Việc tích hợp ứng dụng vào chương trình giúp tăng hiệu quả của việc trình chiếu trực tuyến, tăng tính tương tác của các học viên với hệ thống. Ứng dụng giúp các học viên ở các tỉnh có điều kiện cơ sở hạ tầng thấp cũng có thể xem được video mà không cần tải xuống máy tính để xem, vừa có thể bảo vệ được bảo quyền của video và giảng viên có thể theo dõi quá trình học của các học viên ở xa.

* 1. **Đối với kinh tế - xã hội**

Ngày nay, việc chống xâm phạm bản quyền nói chung và của dữ liệu media nói riêng đang là một vấn đề được quan tâm. Hướng đi của đề tài là góp phần vào công cuộc bảo vệ tác quyền, tránh một khoảng thất thoát không nhỏ cho các nhà cung cấp dịch vụ.

## Phạm vi ứng dụng

Với ưu điểm là sản phẩm được xây dựng hỗ trợ trình chiếu đa số các định dạng video phổ biến hiện nay, nên dễ dàng tích hợp vào các hệ thống có nhu cầu trình chiếu video. Bên cạnh đó, sản phẩm đã giải quyết được bài toán chống vi phạm bản quyền video và vấn đề quá tải băng thông mạng cũng như máy chủ, mà các hệ thống thương mại điện tử trên lĩnh vực kinh doanh sản phẩm multi-media và truyền thông đang quan tâm hiện nay. Từ đó, chúng ta thấy được tính khả thi của đề tài, phạm vi ứng dụng của sản phẩm rất rộng rãi từ chương trình đạo tạo trực tuyến từ xa, những hệ thống chia sẻ video và các hệ thống truyền thông, giải trí thông qua internet.

## Hướng phát triển

Để hoàn thành hệ thống, đề tài sẽ tiếp tục được phát triển tiếp các bước sau:

* Thử nghiệm nhiều giải pháp caching trên server, tiến hành đo đạc và chọn ra giải pháp tối ưu nhất.
* Hiện thực giao diện cho máy khách dưới dạng một ứng dụng hoặc plugin cho web.
* Triển khai ứng dụng vào các hệ thống thực tế, bước đầu là tích hợp ứng dụng vào sakai để hỗ trợ chương trình đào tạo từ xa của Khoa KH&KT Máy tính, ĐH Bách Khoa Tp HCM.

# TÀI LIỆU THAM KHẢO

[1] Xuggle: <http://www.xuggle.com/xuggler/doc/>

[2] Neatbean: <http://netbeans.org/kb/docs/cnd/beginning-jni-linux.html>

[3] FLV: <http://www.adobe.com/devnet/f4v.html>

[4] Ffmpeg: <http://www.ffmpeg.org/>

[5] Java: [www.java.com](http://www.java.com)

[6]Caching: An Interactive video Delivery and Caching System Using video Summarization - Sung-Ju Lee, Wei-Ying Ma, and Bo Shen

[7] Socket: [http://www.sockets.com](http://www.sockets.com/)

[8] C & C++: [http://www.cplusplus.com](http://www.cplusplus.com/)

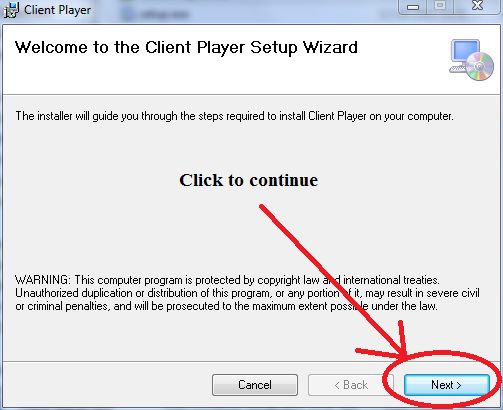
[9] A. van Hoof (2009), Top40 cache algorithm compared to LRU and LFU

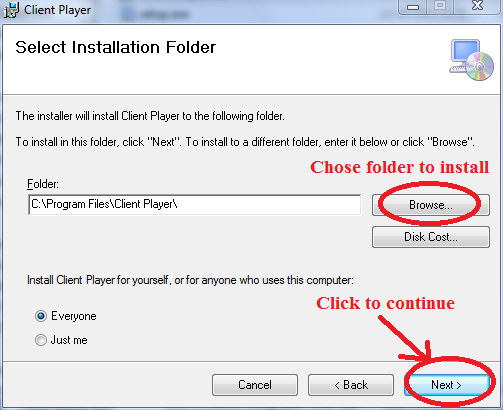
[10] Giải thuật bộ đệm : <http://en.wikipedia.org/wiki/Cache_algorithms>

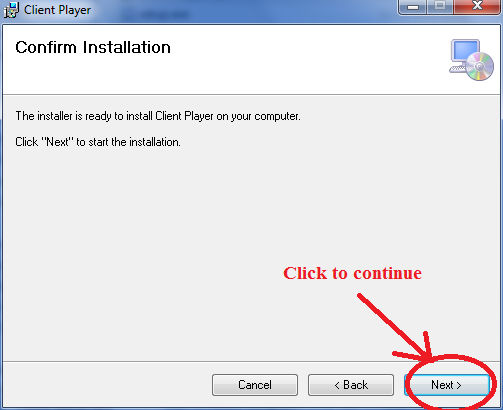
[11] Hướng dẫn mở port modem : <http://forum.hdvnbits.org/thao-luan-chung-ve-torrent/mo-port-cho-cac-modem-thuong-gap-6.html>

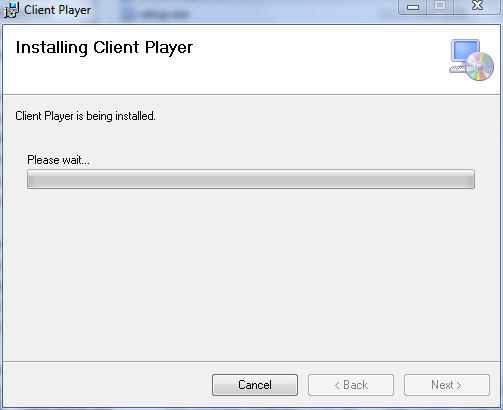
# PHỤ LỤC

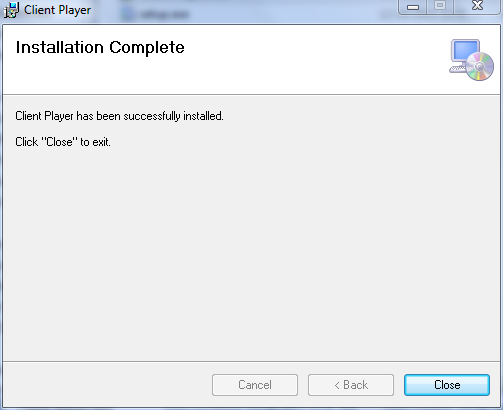
## Hướng dẫn cài đặt Client

Hướng dẫn cài đặt theo hình  






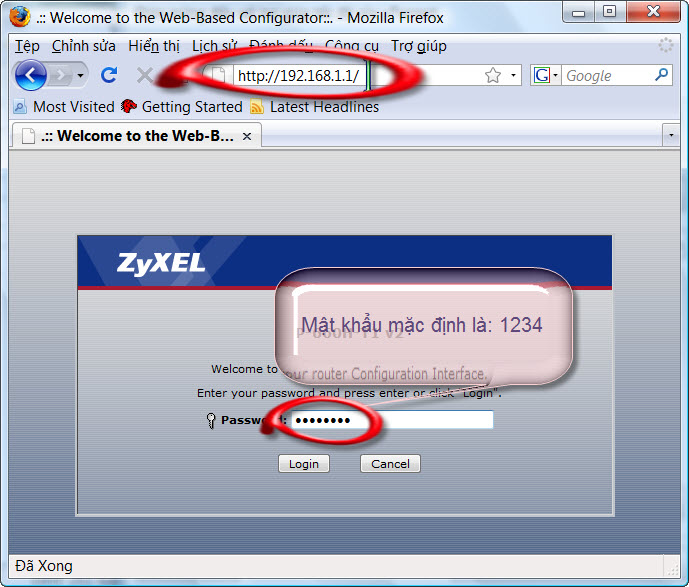


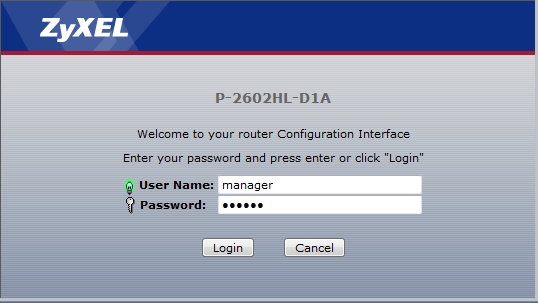


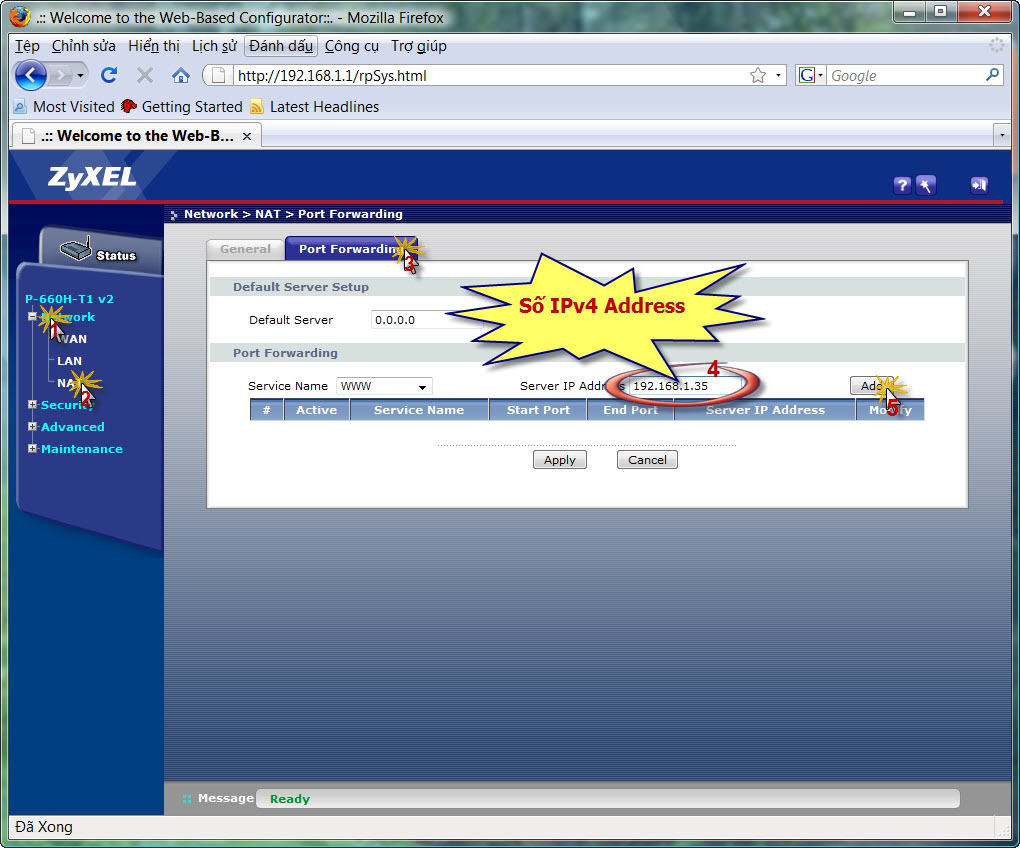
## Hướng dẫn mở port của modem (MODEM ZYXEL( P-660H-T1 và P-2602HL-D1A))

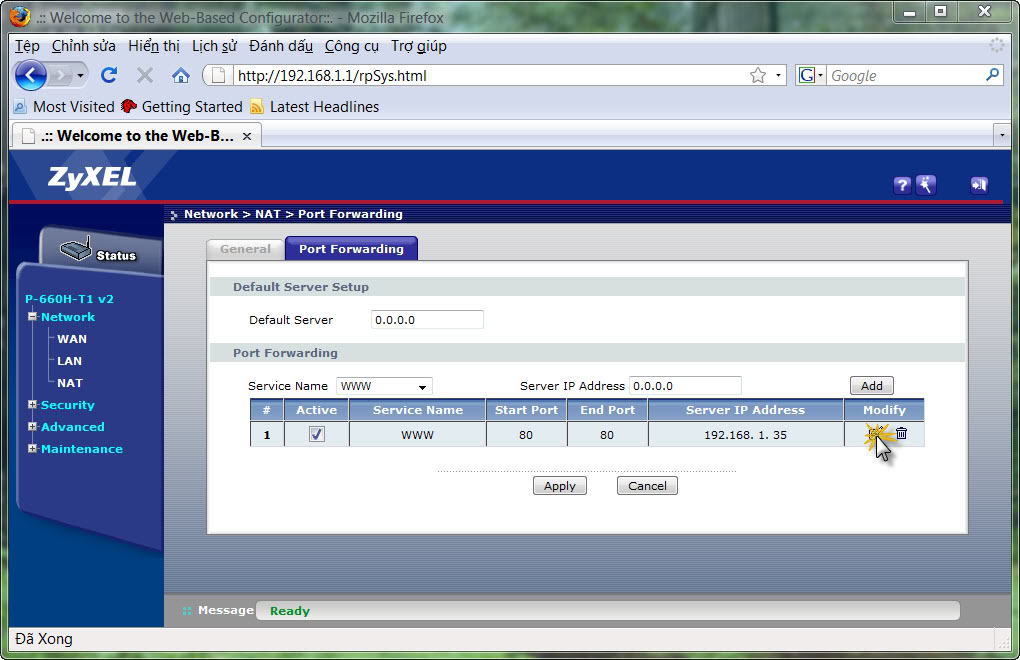
Mở trình duyệt web và gõ: http://192.168.1.1/

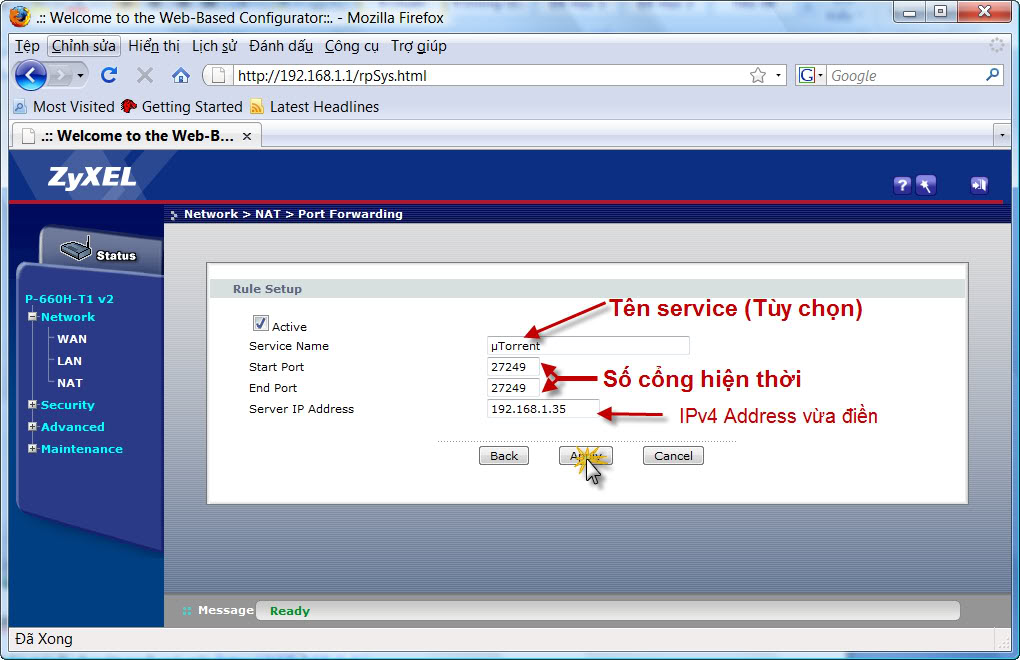
Mật khẩu mặc định là: 1234( P-660H-T1) . Lưu ý với modem loại mới(P-2602HL-D1A) thì user và pass đăng nhập đều là manager



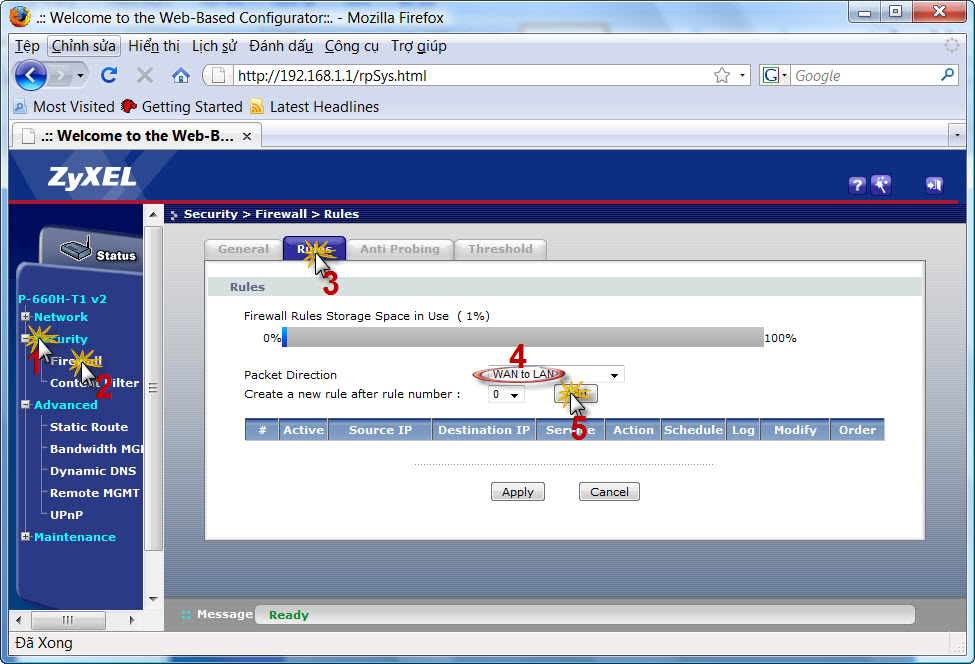


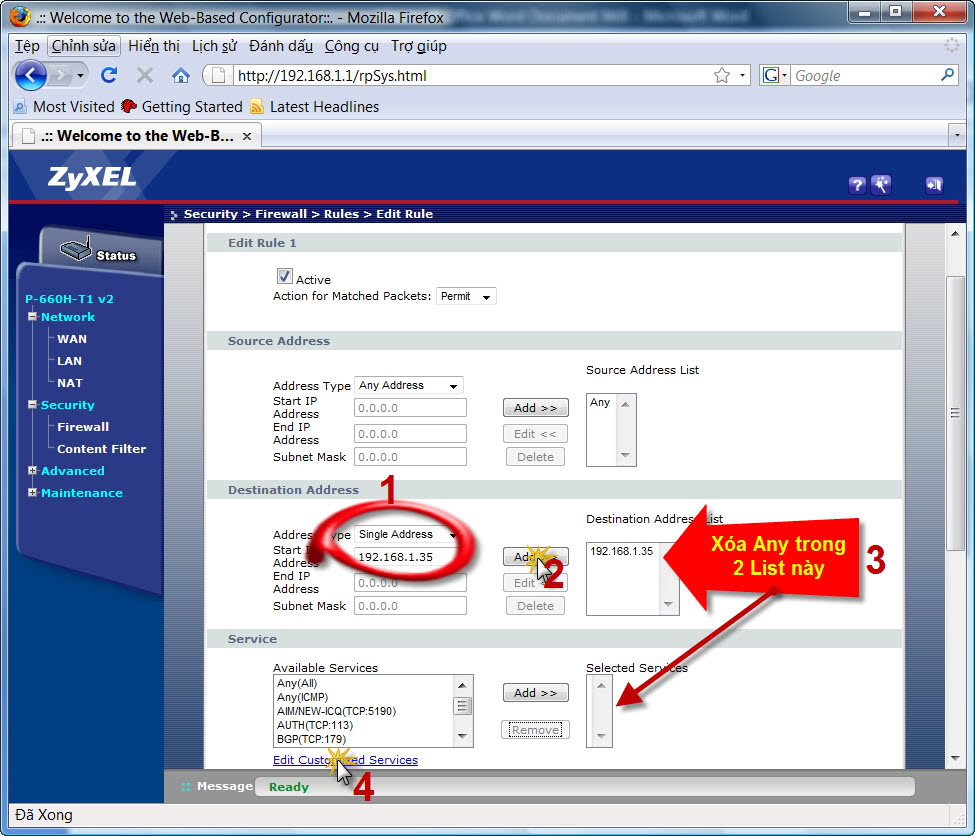


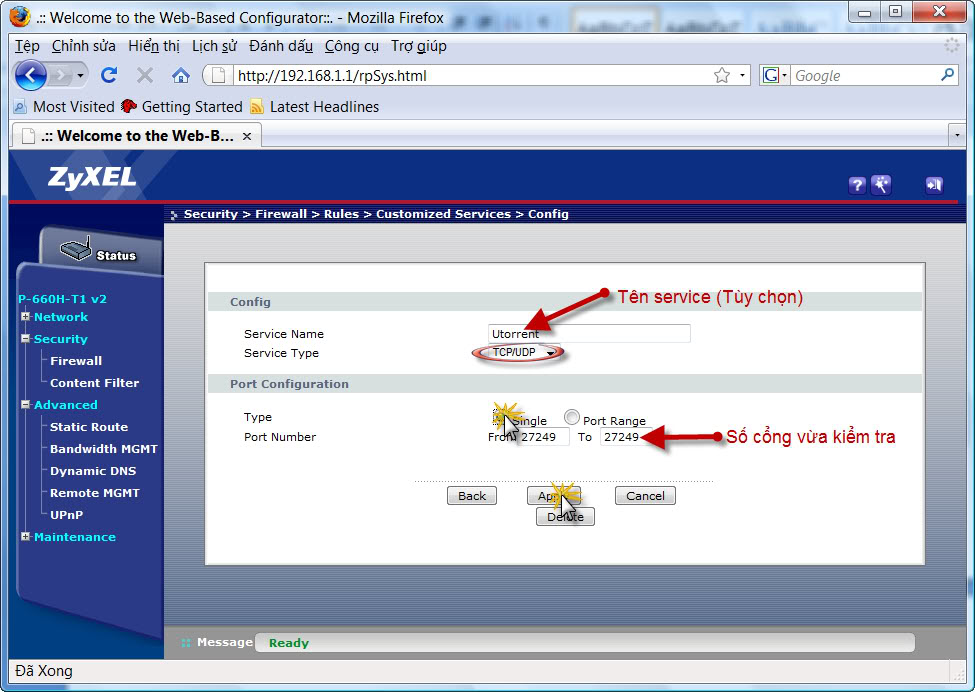




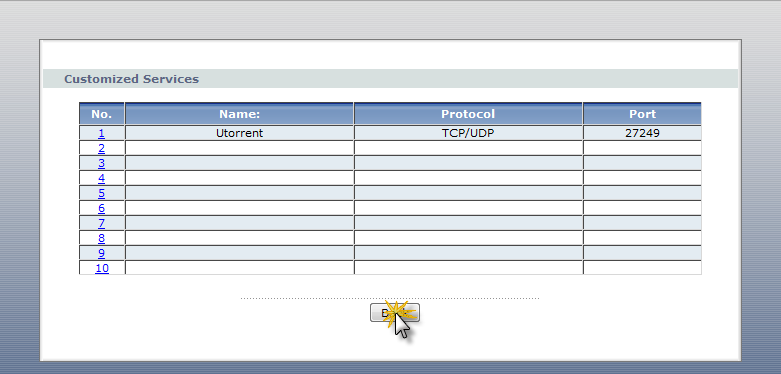
**Lưu ý**: Số port bạn có thể chọn một cách ngẫu nhiên. Ở đây chọn 27249

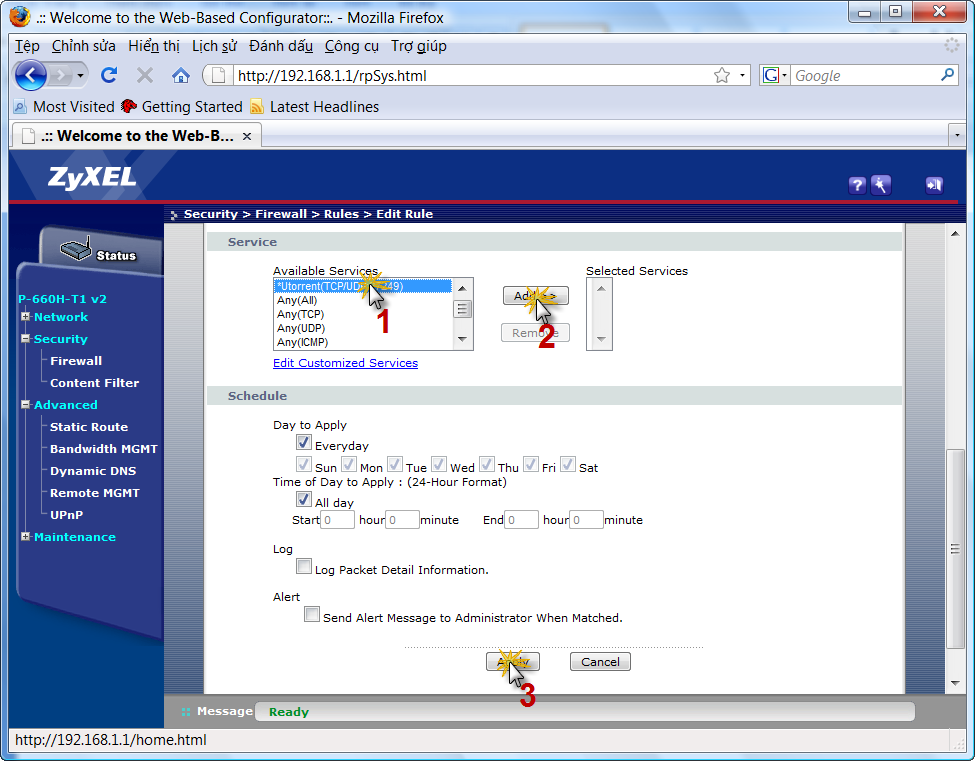


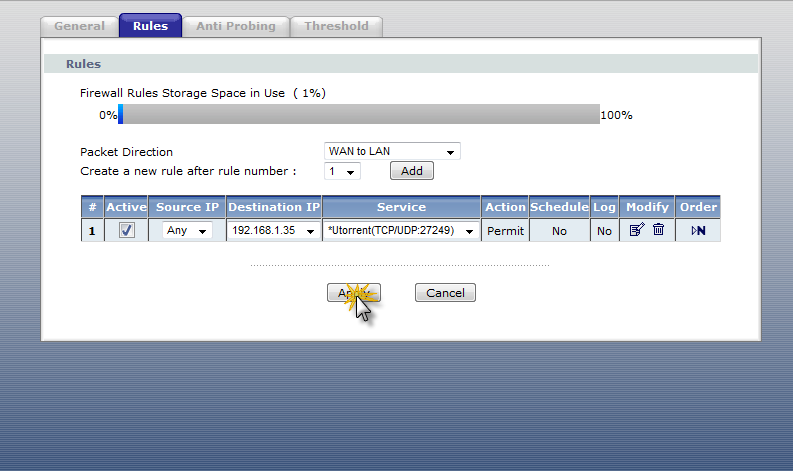




Sau đó ấn Back để trở lại phần Edit Rules

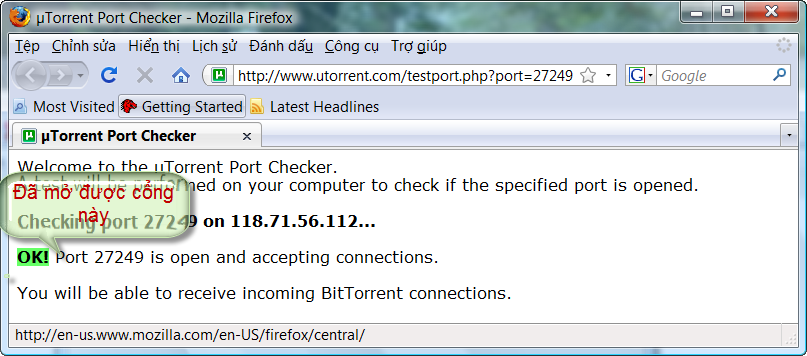


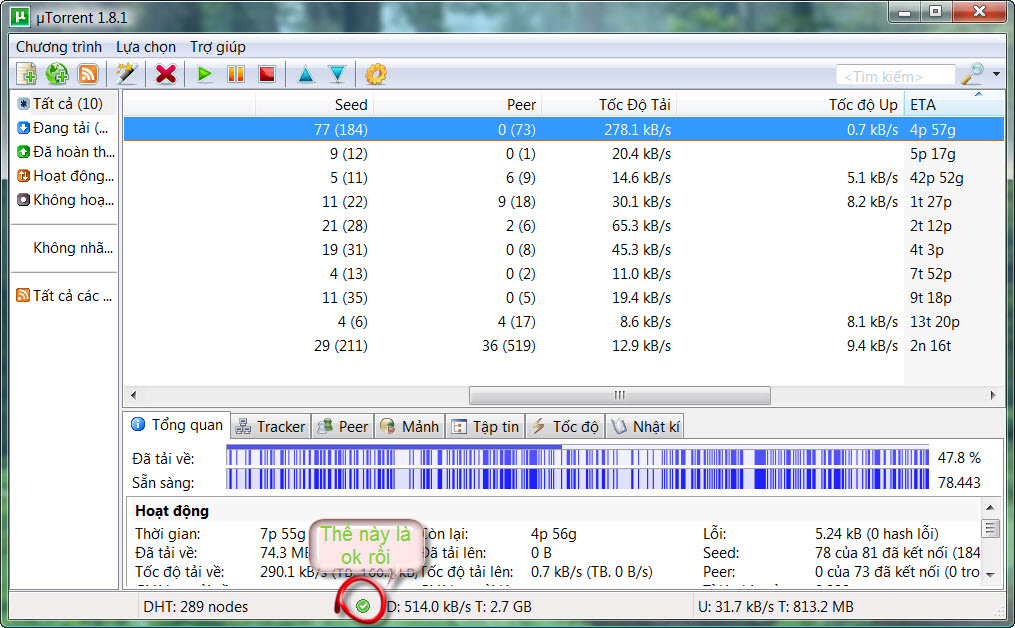




Kiểm tra lại kết quả







**Chú ý:** Nếu làm như trên mà vẫn không được thì kiểm tra lại tường lửa của windows có chặn không.