**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**

----🙠🕮🙢----



**HỆ CƠ SỞ DỮ LIỆU TN (CO2014)**

**BÀI TẬP LỚN SỐ 1**

**CHỦ ĐỀ: QUẢN LÝ QUÁN CÀ PHÊ**

**HK: 241 - LỚP: L02**

**GVHD: Trương Quỳnh Chi**

|  |  |  |
| --- | --- | --- |
| **Nhóm thực hiện: Nhóm Cà Phê** | | |
| Nguyễn Tấn Tài | : | 2212990 |
| Nguyễn Trung Hiếu | : | 2113357 |
| Đặng Trung Phát | : | 2212499 |
| Đồng Mạnh Cường | : | 2210420 |
| Trần Chấn Thịnh | : | 2213309 |

Thành phố Hồ Chí Minh, tháng 11 năm 2024

# **DANH SÁCH THÀNH VIÊN**

**Nhóm: Nhóm Cà Phê**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ tên** | **MSSV** | **Tỷ lệ đóng góp** | **Ký tên xác nhận** |
| 1 | Nguyễn Tấn Tài | 2212990 |  |  |
| 2 | Nguyễn Trung Hiếu | 2113357 |  |  |
| 3 | Đặng Trung Phát | 2212499 |  |  |
| 4 | Đồng Mạnh Cường | 2210420 |  |  |
| 5 | Trần Chấn Thịnh | 2213309 |  |  |
| 6 | Nguyễn Tấn Tài | 2212990 |  |  |

Contents

[DANH SÁCH THÀNH VIÊN 1](#_Toc183469666)

[1. Introduction 2](#_Toc183469667)

[**1.1. Objectives** 2](#_Toc183469668)

[**1.2 Overview of the Application (Tổng quan ứng dụng)** 2](#_Toc183469669)

[**1.4. Nền tảng lý thuyết** 4](#_Toc183469670)

# **1. Introduction**

## **1.1. Objectives**

Mục tiêu chính của bài tập lớn là xây dựng một ứng dụng chia sẻ file mạng ngang hàng đơn giản (Simple Torrent-like Application - STA) dựa trên lý thuyết và giao thức mạng. Ứng dụng phải đáp ứng các yêu cầu sau:

* Sử dụng giao thức TCP/IP để trao đổi dữ liệu giữa các node (máy ngang hàng) và tracker (máy chủ trung tâm).
* Triển khai mô hình Peer-to-Peer (P2P), cho phép tải xuống và tải lên dữ liệu giữa các node.
* Hỗ trợ truyền dữ liệu đa hướng (Multi-Directional Data Transfer - MDDT), nghĩa là:
* Một file có thể được tải đồng thời từ nhiều nguồn.
* Node có khả năng xử lý song song (đa luồng) các kết nối đến nhiều peers.
* Quản lý trạng thái các file thông qua hệ thống tracker, bao gồm thông tin:
* Các mảnh file (pieces) đã tải/xuất bản.
* Danh sách các peers chứa dữ liệu cần thiết.
* Tạo giao diện người dùng thân thiện qua dòng lệnh (CLI), cung cấp thông tin về:
* Tình trạng tải lên/tải xuống.
* Các file đang được quản lý.

Mục tiêu cuối cùng là áp dụng lý thuyết mạng máy tính vào thực tế, đồng thời rèn luyện kỹ năng xây dựng ứng dụng phân tán.

## **1.2 Overview of the Application (Tổng quan ứng dụng)**

Ứng dụng STA hoạt động dựa trên hai thành phần chính: Tracker và Peer Nodes.

***1.2.1 Tracker***

Tracker là máy chủ trung tâm thực hiện các nhiệm vụ sau:

Quản lý danh sách file và các node:

* Mỗi file chia sẻ được lưu trữ dưới dạng thông tin meta (Metainfo File), bao gồm:
* Danh sách các mảnh file (pieces) và hash của từng mảnh.
* Danh sách các node chứa các mảnh này (peers).
* Xử lý yêu cầu từ các node:
* Khi một node tải file, nó gửi yêu cầu đến tracker để nhận danh sách các peers đang giữ file đó.
* Tracker phản hồi danh sách IP và cổng của các peers.

***1.2.2 Peer Nodes***

Peer là các máy ngang hàng tham gia chia sẻ và tải dữ liệu. Mỗi node đảm nhận hai vai trò chính:

**Downloader (Client):**

* Yêu cầu danh sách peers từ tracker.
* Kết nối tới các peers để tải dữ liệu.
* Quản lý trạng thái các mảnh file đã tải/xuất bản.

**Seeder (Server):**

* Chia sẻ các mảnh file mà node đang sở hữu cho các peers khác.
* Hỗ trợ nhiều kết nối đồng thời (đa luồng) để tăng hiệu suất.

***1.2.3 Mô hình hoạt động***

1. **Tải file (Downloading):**

* Node gửi yêu cầu tới tracker để nhận danh sách peers chứa file cần tải.
* Node kết nối tới các peers, yêu cầu từng mảnh dữ liệu (pieces).
* Dữ liệu tải về được xác minh tính toàn vẹn qua hash.

1. **Chia sẻ file (Seeding):**

* Sau khi tải xong, node tự động chuyển sang chế độ seeding.
* Chấp nhận kết nối từ các peers khác và cung cấp dữ liệu.

1. **Đa hướng dữ liệu (MDDT):**

* Hỗ trợ tải file từ nhiều nguồn đồng thời.
* Quản lý danh sách các mảnh đã tải, tránh yêu cầu trùng lặp.

***1.2.4 Đặc điểm nổi bật***

1. **Magnet Text:**

* Là chuỗi ký tự trỏ tới Metainfo File trên tracker.
* Chứa thông tin cơ bản như hash của file và địa chỉ tracker.

1. **Metainfo File (.torrent):**

* Tệp chứa thông tin chi tiết về file, bao gồm:
* Danh sách các mảnh file (pieces).
* Kích thước mỗi mảnh.
* Địa chỉ tracker.

1. **Pieces:**

* File được chia thành các mảnh nhỏ (thường là 512 KB).
* Mỗi mảnh có một hash để xác minh tính toàn vẹn.

1. **Giao thức Tracker:**

* Sử dụng giao thức HTTP để giao tiếp giữa tracker và nodes.

## **1.4. Nền tảng lý thuyết**

Việc phát triển ứng dụng Simple Torrent-like Application (STA) dựa trên các khái niệm lý thuyết quan trọng trong mạng máy tính và hệ thống phân tán:

***1.4.1 Ngăn xếp giao thức TCP/IP***

Ứng dụng sử dụng ngăn xếp giao thức TCP/IP để đảm bảo việc truyền dữ liệu giữa các thành phần (tracker và node) diễn ra một cách tin cậy và hiệu quả. Giao thức TCP chịu trách nhiệm:

* Đảm bảo dữ liệu được truyền đến đúng thứ tự.
* Phát hiện và xử lý lỗi trong quá trình truyền tải.

Việc dựa trên TCP/IP giúp hệ thống duy trì độ ổn định cao, phù hợp với các ứng dụng đòi hỏi truyền dữ liệu chính xác như chia sẻ file.

***1.4.2 Mạng ngang hàng (Peer-to-Peer Networking)***

Mô hình Peer-to-Peer (P2P) được ứng dụng để phân tán tải giữa các node trong hệ thống, giảm bớt phụ thuộc vào tracker. Trong mô hình này:

* Mỗi node vừa là client (tải dữ liệu) vừa là server (chia sẻ dữ liệu).
* Tính phi tập trung giúp hệ thống mở rộng linh hoạt khi số lượng node tăng lên.

***1.4.3 Xác minh hash (Hash Verification)***

Để đảm bảo tính toàn vẹn của dữ liệu, mỗi mảnh file (piece) được gắn một mã hash duy nhất, được lưu trữ trong tệp Metainfo File. Trong quá trình tải file:

* Node tải về từng mảnh từ các peers và xác minh mã hash.
* Nếu hash không khớp, mảnh sẽ bị loại bỏ và tải lại từ nguồn khác.

Cơ chế này đảm bảo rằng dữ liệu cuối cùng chính xác và không bị sửa đổi.

***1.4.4 Đa luồng (Multi-threading)***

Ứng dụng sử dụng kỹ thuật đa luồng để:

* Xử lý đồng thời nhiều kết nối tới các peers khác nhau, giúp tối ưu hóa tốc độ tải.
* Hỗ trợ tải file từ nhiều nguồn đồng thời (MDDT) mà không bị nghẽn dữ liệu.

Kỹ thuật đa luồng không chỉ tăng hiệu suất mà còn giảm thời gian chờ khi xử lý các kết nối lớn.

***1.4.5 Quản lý trạng thái file***

Hệ thống duy trì trạng thái của từng file thông qua các tệp JSON, bao gồm:

* Danh sách các mảnh đã tải hoặc chưa tải.
* Hash của từng mảnh để kiểm tra tính toàn vẹn.
* Trạng thái của file (đang tải, hoàn thành, hoặc đang seed).

Cơ chế quản lý trạng thái này giúp ứng dụng dễ dàng xử lý gián đoạn và tiếp tục tải khi có lỗi kết nối.