**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**



**BÀI TẬP LỚN**

**MÔN**

**ĐỀ TÀI**

**LỚP L--- NHÓM --- HK 241**

**Giảng viên hướng dẫn:**

Mục lụ

***[Thành phố Hồ Chí Minh – 2024](#_Toc19904)* [3](#_Toc19904)**

**[FUNCTION 4](#_Toc29999)**

**[1. Quản lý Tracker 4](#_Toc110)**

**[2. Tải xuống file (Download) 4](#_Toc21306)**

**[3. Tải lên file (Upload) 4](#_Toc20904)**

**[4. Giao diện người dùng (UI) 4](#_Toc13468)**

**[5. Quản lý dữ liệu tệp (File Manager) 5](#_Toc26871)**

**[6. Các tính năng Extra credit 5](#_Toc3182)**

|  |  |  |
| --- | --- | --- |
| **Sinh viên** | **Mã số sinh viên** | **Điểm số** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

*Thành phố Hồ Chí Minh – 2024*

**FUNCTION**

**1. Quản lý Tracker**

* **Ghi nhận và lưu trữ thông tin các Peer**: Tracker đóng vai trò lưu trữ thông tin về các Peer (Node) đang chia sẻ hoặc yêu cầu tải file. Khi một Node gửi yêu cầu kết nối, Tracker cần cập nhật và lưu trữ thông tin của Node đó, bao gồm địa chỉ IP, số cổng, và ID của Peer.
* **Cung cấp danh sách các Peer**: Khi một Node muốn tải một tệp, Tracker sẽ cung cấp danh sách các Peer đang sở hữu tệp đó. Danh sách này sẽ chứa địa chỉ IP và số cổng của các Peer.
* **Theo dõi trạng thái Peer**: Tracker cần quản lý trạng thái của các Peer (bắt đầu, dừng lại, hoặc hoàn thành việc chia sẻ/tải file) và cập nhật thông tin tương ứng.
* **Tracker Scrape**: Tracker có thể thực hiện **scrape** để cung cấp thông tin chi tiết về số lượng seeder và leecher đang hoạt động cho mỗi torrent. Điều này giúp đánh giá tình trạng mạng và tối ưu kết nối.

**2. Tải xuống file (Download)**

* **Gửi yêu cầu đến Tracker**: Khi Node muốn tải một tệp, nó sẽ gửi yêu cầu đến Tracker để nhận danh sách các Peer đang sở hữu tệp cần tải.
* **Kết nối với các Peer**: Sau khi nhận được danh sách các Peer từ Tracker, Node sẽ thiết lập kết nối với nhiều Peer cùng một lúc để tải các phần của tệp.
* **Quản lý nhiều luồng tải xuống (MDDT)**: Hỗ trợ tải xuống từ nhiều nguồn cùng lúc, tối ưu băng thông và giảm thời gian tải file nhờ sử dụng đa luồng.
* **Chiến lược tải xuống nâng cao**: **End Game Mode**: Tăng tốc độ tải phần cuối cùng của tệp bằng cách yêu cầu cùng một dữ liệu từ nhiều Peer. **Rarest-First Strategy**: Tải xuống những phần hiếm trước để đảm bảo không bị kẹt do thiếu Peer cung cấp.

**3. Tải lên file (Upload)**

* **Chia sẻ dữ liệu với các Peer khác**: Ngoài ra Node cũng đóng vai trò là Seeder (người chia sẻ) và cho phép các Peer khác kết nối để tải tệp từ nó.
* **Đáp ứng các yêu cầu từ Peer khác**: Khi một Peer gửi yêu cầu tải dữ liệu, Node phải kiểm tra xem nó lưu trữ phần dữ liệu đó hay không và gửi dữ liệu tương ứng.
* **Áp dụng chiến lược tải lên**: Node có thể sử dụng các chiến lược nâng cao như "tit-for-tat", tức là chỉ chia sẻ nhiều hơn với các Peer chia sẻ dữ liệu ngược lại để ngăn chặn việc "free-riders" (những người chỉ tải mà không chia sẻ).
* **Super Seeding Mode**: Node chỉ gửi mỗi phần dữ liệu cho một Peer duy nhất để khuyến khích Peer đó chia sẻ lại với các Peer khác.

**4. Giao diện người dùng (UI)**

* **Hiển thị thông tin tải lên và tải xuống**: Ứng dụng cần có giao diện hiển thị các thống kê về tốc độ tải lên, tải xuống, và các Peer đang kết nối.
* **Quản lý các tệp đang tải và đã tải**: Người dùng có thể xem danh sách các tệp đang được tải hoặc đã hoàn thành, cũng như xem chi tiết trạng thái tải của từng tệp (phần nào đã tải xong, phần nào còn thiếu).
* **Truy cập thống kê nhiều torrent cùng lúc**: Ứng dụng cung cấp thông tin chi tiết về từng torrent đang được tải hoặc seed, giúp người dùng theo dõi hiệu quả quá trình chia sẻ dữ liệu.

**5. Quản lý dữ liệu tệp (File Manager)**

* **Lưu trữ các tệp**: Các tệp sau khi được tải xuống cần được lưu trữ tạm thời và sau đó được ghép lại để tạo thành tệp hoàn chỉnh.
* **Kiểm tra tính toàn vẹn của tệp**: Đảm bảo rằng tất cả các mảnh tệp tải xuống không bị hỏng thông qua việc kiểm tra mã băm.
* Tích hợp với lưu trữ đám mây: Tự động tải lên các tệp hoàn tất vào **Google Drive** hoặc các nền tảng lưu trữ khác.

1. **Các tính năng Extra credit**

* **Bảng băm phân tán (DHT)**: Cho phép truyền tải tệp torrent của bạn mà không cần dùng đến Tracker tập trung. DHT thường là một nguồn tốt hơn để tìm nhiều Peer so với việc chỉ sử dụng Tracker tập trung của torrent.
* **Hỗ trợ nhiều torrent đồng thời**: Yêu cầu ứng dụng của bạn có khả năng tải xuống và tải lên nhiều torrent cùng lúc, đồng thời cung cấp khả năng truy cập vào các thống kê qua giao diện người dùng. Tracker scrape: ([https://en.wikipedia.org/wiki/Tracker\_scrape](https://en.wikipedia.org/wiki/Tracker_scrape" \t "_new)) là quá trình trao đổi để lấy thông tin meta đại diện cho khách hàng (client). Cần lưu ý rằng việc tham gia của một Peer vào quá trình truyền dữ liệu không bị ảnh hưởng bởi các giao dịch scrape.
* **Chiến lược tải xuống/seed**: Ứng dụng của bạn cần có ít nhất một chiến lược tải xuống cơ bản, nhưng nếu bạn chọn triển khai các chiến lược nâng cao (như Super Seeding, Rarest-First, hoặc End Game), bạn cần phải ghi lại và cung cấp các tài liệu bổ sung.
* **Tối ưu hóa việc lựa chọn Peer**: Ứng dụng của bạn cần có chiến lược chọn Peer cơ bản, ví dụ như phản hồi với tất cả các Peer. Tuy nhiên, nếu bạn muốn áp dụng tiêu chuẩn thực tế (như chỉ duy trì 4+1 Peer cùng lúc), bạn phải ghi chú lại chi tiết trong tài liệu.

PROTOCOL

**1. Client-Server (HTTP):**

* Kết nối: “ANNOUNCE <metainfo> <magnet>” các client sẽ thông báo cho server để kết nối.
* Files: “LIST\_FILES” các client được xem thông tin về các file mà tất cả các client đang chia sẻ
* “METADATA\_REQUEST <info\_hash>” Client yêu cầu server cung cấp metadata về 1 tệp cụ thể.
* “PING” Node sử dụng lệnh ping để kiểm tra tình trạng trực tuyến của tracker.

**2. Peer-to-peer (TCP):** mỗi peer có 1 TCP client để sử lý các “chunk” và 1 TCP server để giải quyết các “chunk”.

* Download: Client gửi “download <file\_id> <chunk\_id>”, nếu có, server sẽ gửi những “chunk” được yêu cầu, nếu không sẽ gửi “File not found”.
* Upload: Client gửi “upload <file\_id> <chunk\_id>”, Server sẽ nhận file theo các “chunk”
* “STARTED <bytes>”, “STOPPED” và “COMPLETED <file\_id>”được gửi khi bắt đầu, dừng và hoàn thành việc tải 1 file

1. **Giao thức cho Các Tính Năng Nâng Cao:**

* Distributed Hash Table (DHT): Mỗi node lưu trữ một phần nhỏ thông tin về các node khác trong mạng. Khi một node cần tìm kiếm tệp, nó gửi truy vấn qua hệ thống DHT thay vì tracker tập trung, từ đó nhận danh sách các peer có tệp cần tìm.
* Tracker Scrape (HTTP): Node gửi yêu cầu scrape đến tracker để nhận thông tin chi tiết về tình trạng của tệp và các peer có sẵn.