

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Computer Networks - CO3094

Báo cáo

SIMPLE FILE SHARING APPLICATION

Giảng viên hướng dẫn: Lê Bảo Khánh

Sinh viên thực hiện: 2114417 - Nguyễn Ngọc Phú



Mục lục

1. Phân tích yêu cầu ứng dụng	3
2. Kiến trúc ứng dụng	4
3. Phân tích các application protocol được sử dụng	5
3.1. HTTP	5
3.2. WebSocket	5

1. Phân tích yêu cầu ứng dụng

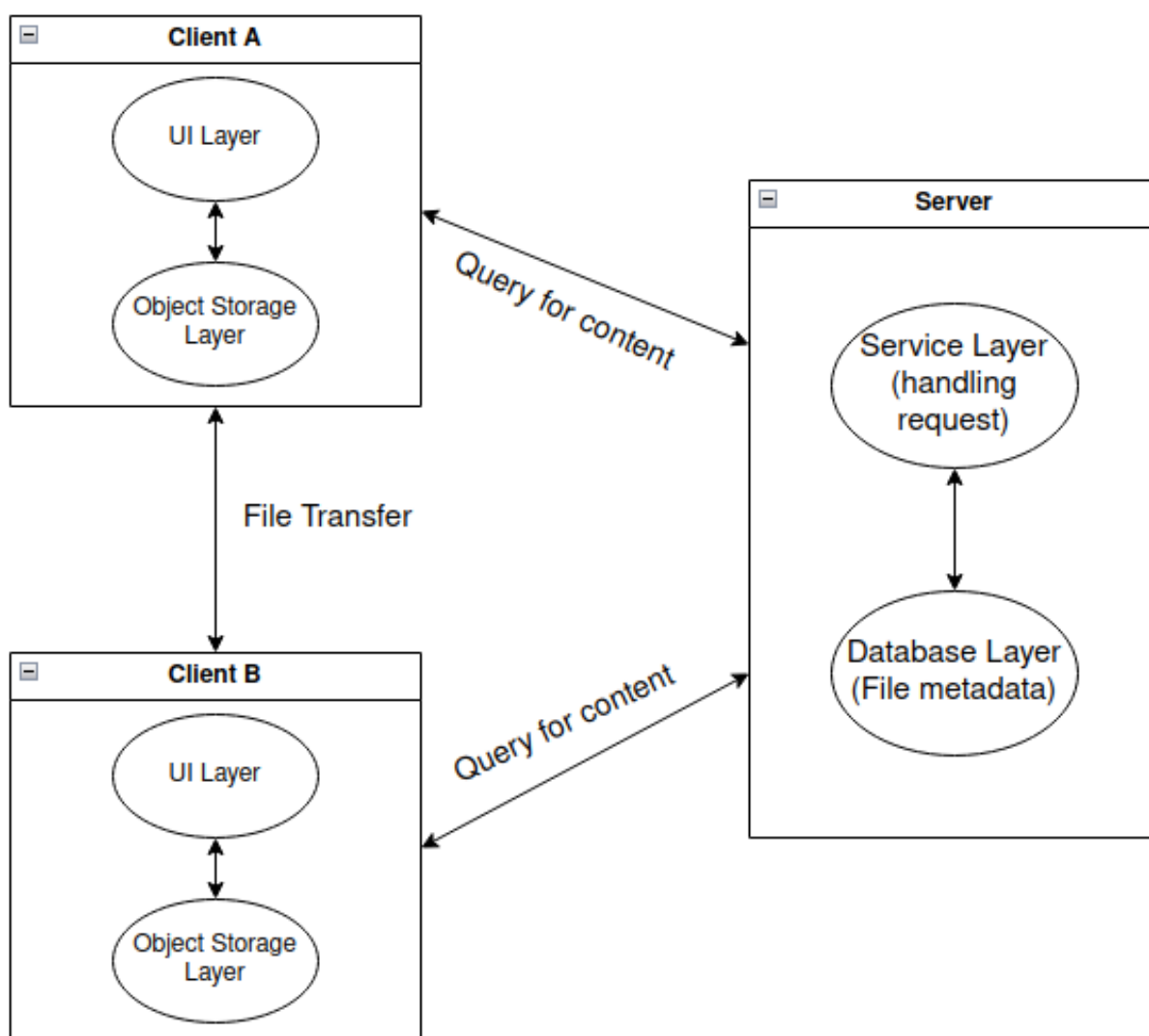
1. Server phải có thông tin của tất cả client tham gia vào hệ thống file-sharing, bao gồm: địa chỉ IP, thông tin tất cả file được lưu trong local repository của mỗi client (tên, URL, size, timestamp).
2. Server có một trình thông dịch chấp nhận các lệnh sau:
 - **discover hostname**: liệt kê danh sách các file (thông tin mỗi file bao gồm: tên, URL, size, timestamp) trên local repository của 1 client tên là *hostname*. Nếu không tồn tại hoặc client *hostname* tạm thời không chấp nhận bất cứ request nào, thông báo lỗi cho server.
 - **ping hostname**: kiểm tra client *hostname* có đang hoạt động trong hệ thống file-sharing hay không.
3. Client có một trình thông dịch chấp nhận các lệnh sau:
 - **publish lname fname**: thêm 1 file có đường dẫn *lname* từ hệ thống file trên thiết bị cá nhân vào local repository của client và được đặt tên là *fname*. Local repository cũng lưu thêm timestamp cho mỗi file được thêm vào để phân biệt các file cùng tên. Metadata của file (tên, URL, size, timestamp) sẽ được gửi đến server, server chỉ lưu version mới nhất của một file ứng với một client.
 - **fetch fname**: tìm và tải về một file tên *fname* vào local repository từ các client khác trong hệ thống file-sharing. Nếu có nhiều file cùng tên, lặp qua tất cả và chọn tải về file có version mới nhất (timestamp gần đây nhất). Nếu không có tồn tại file *fname*, thông báo lỗi cho client.
 - **delete fname**: Xóa một file tên *fname* ra khỏi local repository của client. Đồng thời, xóa metadata của file này ra khỏi server. Nếu có bất cứ request nào về thông tin file này đến server, thông báo lỗi *Không tồn tại file* cho client.
 - **ls fname**: Liệt kê địa chỉ IP của tất cả client chứa file *fname* trong hệ thống file-sharing.
 - **store**: Liệt kê metadata của tất cả file hiện đang được lưu trên server.
4. Nhiều client có thể tải xuống các file khác nhau từ một client “*mục tiêu*” tại cùng một thời điểm. Điều này yêu cầu ứng dụng client phải hỗ trợ đa luồng để xử lý việc tải nhiều file cùng một lúc mà không gây xung đột hoặc trì hoãn.

2. Kiến trúc ứng dụng

Từ các yêu cầu được phân tích ở trên, ta sử dụng kiến trúc lớp cho hệ thống file-sharing, bao gồm 4 lớp:

1. Lớp lưu trữ cục bộ (Object Storage Layer): Lớp này chính là local repository của mỗi client, lưu metadata của file và dữ liệu của file.
2. Lớp giao diện (UI Layer): Chỉ xây dựng giao diện cho client, bao gồm các thành phần liên quan đến 5 command-line: **publish lname fname**, **fetch fname**, **delete fname**, **ls fname**, **store**.
3. Lớp xử lý nghiệp vụ (Service Layer): Xử lý tất cả các request từ client, tương tác với lớp cơ sở dữ liệu để cung cấp metadata của file cho client.
4. Lớp cơ sở dữ liệu (Database Layer): Lưu metadata của tất cả file trong hệ thống file-sharing.

Ngoài ra ứng dụng còn cung cấp command-shell interpreter để thực thi các command-line của server và thay cho lớp giao diện của client.



Hình 1: Kiến trúc hệ thống file-sharing

3. Phân tích các application protocol được sử dụng

3.1. HTTP

Protocol này được sử dụng cho các command line sau:

1. **discover hostname**: gửi một GET HTTP cho hostname yêu cầu trả về danh sách metadata của tất cả file hiện có trên local repository của client.
2. **publish lname fname**: gửi một POST HTTP tải dữ liệu của file *lname* lên local repository với tên mới là *fname*, sau đó gửi một POST HTTP tải metadata của file *fname* lên server.
3. **fetch fname**: gửi một GET HTTP có parameter là *fname* lên server để tìm kiếm thông tin các client đang lưu giữ file *fname*, sau đó gửi một GET HTTP cho client đó để tải dữ liệu file yêu cầu về local repository.
4. **delete fname**: gửi một DELETE HTTP có parameter là *fname* lên local repository để xóa dữ liệu tất cả các version của file, sau đó gửi một DELETE HTTP lên server để xóa metadata của file đó.
5. **ls fname**: gửi một GET HTTP có paramter là *fname* đến server yêu cầu trả về địa chỉ IP của tất cả client đang chứa file đó trong hệ thống file-sharing.
6. **store**: gửi một GET HTTP đến server yêu cầu trả về metadata của tất cả file hiện đang lưu giữ bởi server.

3.2. WebSocket

WebSocket là một giao thức mạng cho phép truyền dữ liệu hai chiều (full-duplex) giữa máy chủ và trình duyệt hoặc giữa các ứng dụng khác nhau qua mạng Internet. Giao thức này thường được sử dụng để xây dựng các ứng dụng web thời gian thực, như trò chuyện trực tuyến, trò chơi trực tuyến, cập nhật trực tiếp, và các ứng dụng có khả năng tương tác cao.

WebSocket khác với HTTP, một giao thức mà trình duyệt thường sử dụng để tải các trang web. Với HTTP, trình duyệt gửi yêu cầu đến máy chủ và sau đó máy chủ trả lời với dữ liệu. WebSocket cho phép thiết lập một kết nối duy trì giữa trình duyệt và máy chủ, cho phép dữ liệu được truyền đi và nhận về mà không cần thiết lập lại kết nối.

WebSocket sử dụng một handshake ban đầu dựa trên HTTP để thiết lập kết nối, sau đó nó cho phép gửi và nhận dữ liệu thông qua các gói tin tin nhắn.

Protocol này được sử dụng cho command line **ping**, cho phép target host liên tục trả về các gói dữ liệu nếu host còn đang hoạt động trong hệ thống file-sharing.