

# Distributed File System

---

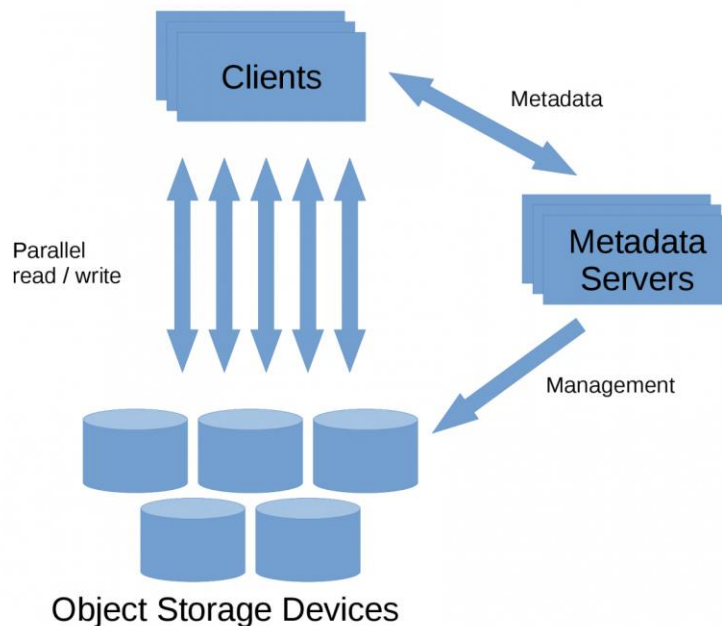
## Thành viên

- Nguyễn Nhật Khang – 21522193
- Nguyễn Hoàng Minh - 21522343

## Giới thiệu đề tài

Trong thời đại công nghệ số và dữ liệu lớn đang bùng nổ như hiện nay, việc lưu trữ phân tán để xử lý dữ liệu lớn một cách hiệu quả, bảo mật và tin cậy là một giải pháp rất quan trọng và mang lại nhiều lợi ích. Các hệ thống này cho phép dữ liệu được lưu trữ trên nhiều máy chủ khác nhau, giúp tối ưu hóa việc phân phối dữ liệu, tăng cường khả năng chống lỗi và đảm bảo khả năng mở rộng khi cần thiết.

Do đó, với yêu cầu xây dựng một ứng dụng sử dụng Golang, đồng thời nêu bật được những ưu điểm nổi bật của nó so với các ngôn ngữ lập trình khác (đặc biệt về mặt hiệu suất). Nhóm em quyết định lựa chọn đề tài **Distributed File System**, một hệ thống lưu trữ phân tán tệp tin sử dụng Go. Với đặc điểm hỗ trợ mạnh mẽ về xử lý đồng thời (concurrency) thông qua Goroutines cùng với hiệu suất cao và khả năng quản lý bộ nhớ tốt, Go rất phù hợp để xây dựng hệ thống này.



## Mục tiêu

Mục tiêu chính của đề tài là phát triển một hệ thống lưu trữ tệp phân tán với các tính năng chính như:

- Lưu trữ và phân tán tệp tin: Cho phép người dùng tải lên, tải xuống và xóa tệp tin trên hệ thống phân tán, với dữ liệu được chia nhỏ và lưu trữ trên nhiều nút khác nhau.
- Quản lý kết nối TCP giữa các nút: Sử dụng các giao thức mạng và Goroutines để xử lý các kết nối đồng thời giữa các nút trong hệ thống.
- Tối ưu hóa hiệu suất: Cải thiện tốc độ truy xuất và quản lý dữ liệu, có thể sử dụng Content Addressable Storage (CAS) để lưu trữ dữ liệu.
- Khả năng mở rộng và chống lỗi.
- Bảo mật dữ liệu

## Công nghệ sử dụng (dự kiến)

- Ngôn ngữ lập trình Golang: Go sẽ được sử dụng để phát triển các chức năng cốt lõi của hệ thống.
- Goroutines: Tận dụng Goroutines của Go để thực hiện các tác vụ đồng thời như tải tệp, kiểm tra tình trạng nút và quản lý kết nối TCP.
- TCP: Giao thức TCP sẽ được sử dụng cho việc kết nối giữa các nút trong hệ thống, với khả năng truyền tải dữ liệu lớn và bảo mật tốt.
- Content Addressable Storage (CAS): Dự kiến sử dụng CAS để quản lý tệp tin dựa trên nội dung của chúng, cho phép lưu trữ và truy xuất hiệu quả.
- ...

## Kế hoạch thực hiện

### 1. Giai đoạn 1: Nghiên cứu và thiết kế (2 – 3 tuần)

- Nghiên cứu các công nghệ và thuật toán liên quan như Go, Goroutines, TCP Handshake, CAS,...
- Thiết kế kiến trúc tổng thể của hệ thống, bao gồm mô hình lưu trữ phân tán, cơ chế chống lỗi và bảo mật.
- Xác định những chức năng cụ thể của hệ thống.

### 2. Giai đoạn 2: Phát triển (1.5 – 2 tháng)

- Triển khai mã nguồn cho các chức năng cơ bản như lưu trữ, truy xuất, và quản lý các kết nối mạng.
- Xây dựng cơ chế phân tán dữ liệu.
- Điều chỉnh các chức năng (nếu cần)

3. Giai đoạn 3: Nghiên cứu, xây dựng cơ chế chống lỗi và bảo mật (2 – 3 tuần)

- Nghiên cứu phát triển cơ chế phục hồi lỗi bao gồm tạo bản sao dữ liệu và tự động khôi phục dữ liệu khi mất kết nối.

4. Giai đoạn 4: Kiểm thử, tối ưu hoá và hoàn thiện (1 – 2 tuần)

- Giai đoạn này sẽ kiểm tra lại các chức năng của hệ thống, đảm bảo tính ổn định và hiệu suất của hệ thống.
- Hoàn thiện hệ thống và hoàn thành báo cáo.

## Kết luận

Dự án **hệ thống lưu trữ tệp phân tán** này không chỉ giúp giải quyết các vấn đề về hiệu suất và quản lý dữ liệu trong các hệ thống lớn, mà còn là một cơ hội để nhóm em nghiên cứu và ứng dụng các công nghệ như lập trình đồng thời, hệ thống mạng phân tán và bảo mật dữ liệu. Sự lựa chọn ngôn ngữ Go và việc tận dụng các ưu điểm của Goroutines sẽ giúp hệ thống trở nên hiệu quả, dễ mở rộng và bảo mật.