# KIẾN TRÚC HỆ THỐNG

GFS (Google File System) là một hệ thống lưu trữ file được phát triển và sử dụng bởi Google từ những năm 90s với những yếu tố chính:

* Hầu hết các file dữ liệu có dung lượng lớn
* Hầu hết các hoạt động đọc là hoạt động đọc liền mạch một đoạn dài hoặc đọc ngẫu nhiên một đoạn ngắn
* Hầu hết các hoạt động viết là ghi thêm thay vì ghi đè, thay thế
* Băng tần rộng và liên tục quan trọng hơn độ trễ thấp
* Hỏng hóc vật lý được xem là thường lệ hơn là ngoại lệ, và được xem xét ngay từ bước đầu của việc xây dựng hệ thống

GFS sử dụng kiến trúc hệ thống tập trung, bao gồm một máy chủ (*master*)và các máy con lưu trữ khối dữ liệu (*chunkserver*).

Dữ liệu lưu trữ được chia thành các khối (*chunk*), được định danh bởi một mã 64 bit độc nhất bất biến gọi là tên khối (*chunk handle*)và được quản lý bởi máy chủ. Máy con lưu trữ các khối dữ liệu, đồng thời đáp ứng nhu cầu đọc/ghi dữ liệu được chỉ định bởi tên khối cùng vị trí byte.

Máy chủ sẽ quản lý mọi dữ liệu hoạt động, bao gồm định danh, thông tin kiểm soát truy cập, mapping từ file về các khối dữ liệu và vị trí các khối dữ liệu hiện tại. Máy chủ cũng kiểm soát hoạt động các khối, dọn dẹp các khối dư thừa cũng như di chuyển các khối dữ liệu.

Kiến trúc tập chung cho phép máy chủ thực hiện các tác vụ yêu cầu thông tin toàn cục của các khối, chả hạn như khóa, nhân bản và nhất quán. Để tránh thắt cổ chai máy chủ, mọi tác vụ đọc và ghi được thực hiện với các máy con.

Mỗi khối dữ liệu có độ lớn 64 MB. Việc sử dụng mức lớn cho khối dữ liệu giúp giảm áp lực cho máy chủ, giảm dung lượng metadata được lưu ở máy chủ. Ngoài ra, do dung lượng khối dữ liệu lớn, client sẽ thực hiện nhiều tác vụ hơn, điều này cho phép giữ một kết nối dài TCP giữa client và máy con thay vì nhiều kết nối nhỏ, giúp giảm overhead.

# CÁCH TRAO ĐỔI THÔNG TIN

Client sẽ liên lạc với máy chủ để lấy thông tin về vị trí các khối dữ liệu, sau đó liên lạc với các máy con để lấy dữ liệu tương ứng.

File được đặt trong các directories.

GFS hỗ trợ các thao tác với file thông thường: create, delete, open, close, read, write. Ngoài ra, hệ thống còn hỗ trợ 2 thao tác snapshot và record append.

Record append cho phép nhiều người dùng cùng append vào một file dữ liệu trong khi vẫn đảm bảo tính nhất quán.

Snapshot cho phép copy một directory với chi phí rất nhỏ.

Máy chủ và máy con liên lạc với nhau thông qua các gói tin đều đặn. Liên lạc giữa client và các máy con hoặc các máy con với nhau được thực hiện qua kết nối TCP.

# A picture containing diagram Description automatically generatedĐỊNH DANH TRONG HỆ PHÂN TÁN

Các khối dữ liệu được định danh bởi một định danh duy nhất có độ lớn tối đa 64 bit. Sử dụng prefix, các định danh thường có độ lớn nhỏ hơn 64 bit do cùng thuộc 1 file cho phép xử lý prefix có hiệu quả cao.

Định danh – tên khối – của các khối dữ liệu là bất biến và độc nhất trên toàn hệ thống. Mọi xử lý trên định danh được thực hiện duy nhất bởi máy chủ. Các directory được xác định duy nhất bởi full path name thay vì phân lớp. Mỗi directory được kèm theo một khóa đọc/ghi, dù là đường dẫn file hay đường dẫn địa chỉ. Các định danh này được lưu một cách hiệu quả dưới dạng một node trong bảng.

# XỬ LÝ ĐỒNG BỘ HOÁ

Định danh được đảm bảo tính bất biến và duy nhất, và được kiểm soát hoàn toàn bởi máy chủ.

Đồng bộ hóa trên toàn hệ thống được máy chủ kiểm soát thông qua một bản ghi hoạt động (*operation log*). Đồng hồ logic của hệ thống trong các tác vụ đồng thời là thứ tự trên bản ghi hoạt động này. Mỗi khi có các tác vụ thay đổi metadata, bản ghi sẽ ghi chú lại các thay đổi này. Mỗi file cũng như khối sẽ được đánh số bởi mã phiên bản một cách bất biến và duy nhất dựa theo thời gian logic nó được tạo ra trên bản ghi này.

Máy chủ thường xuyên tạo các checkpoint đề phòng trường hợp máy chủ bị hỏng. Ngoài ra, dữ liệu máy chủ liên tục được sao chép lên các máy chủ sao chép. Khi máy chủ bị hỏng, các phiên bản sao chép của máy chủ sẽ thực hiện vai trò của máy chủ bằng cách tái tạo lại trạng thái hiện tại của máy chủ thông qua checkpoint và operation log.

Máy chủ không duy trì một bản ghi vị trí các khối dữ liệu liên tục. Thay vào đó, bản ghi này được cập nhật một lần lúc khởi động bằng cách hỏi các máy con, sau đó dựa vào operation log để duy trì bản ghi này. Bằng cách duy trì các gói tin trao đổi giữa máy chủ và máy con, bản ghi này được cập nhật đều đặn thay vì được đồng bộ hóa liên tục.

Thiết kế hướng dữ liệu này thực hiện ý tưởng quan trọng của hệ thống: hỏng hóc của các máy con được xem như là thường lệ thay vì là ngoại lệ, và được nghiên cứu ngay từ lúc bắt đầu đề xuất hệ thống.

# NHÂN BẢN VÀ NHẤT QUÁN

Mặc định, một khối được nhân bản với 3 bản ghi trên toàn hệ thống. Người dùng có thể điều chỉnh giá trị này.

Để đảm bảo tính nhất quán, khi có yêu cầu thay đổi dữ liệu (thường là append), máy chủ sẽ trao cho một máy con chứa khối đó *hợp đồng* (chunk lease). Khối dữ liệu ở máy con này sẽ được gọi là bản chính, các bản sao chép khác được gọi là bản phụ. Bản chính sẽ kiểm soát mã phiên bản cho các thay đổi được gửi từ client. Chunk lease mặc định kéo dài 60s, tuy nhiên có thể bị hủy bỏ bởi máy chủ, hoặc được kéo dài vô thời hạn trong trường hợp tác vụ vẫn đang được sử dụng. Nếu máy con chứa lease bị hỏng, máy chủ có thể đưa lease cho một bản sao chép ở máy con khác. Diagram

Description automatically generated

Client sẽ hỏi vị trí file, các khối, bao gồm cả các bản sao chép, sau đó cache các thông tin này để thuận tiện liên lạc. Client sau đó sẽ gửi thông tin cần thay đổi tới tất cả các máy con chứa bản lưu, nơi nó được lưu trong một buffer. Client không cần liên lạc với máy chủ trong suốt quá trình này, cho phép luồng dữ liệu đường truyền theo đường ống, tối ưu theo hình trạng mạng với bất cứ thứ tự nào, không yêu cầu gửi tới bản chính đầu tiên. Một khi dữ liệu được các bản ghi xác nhận, client gửi yêu cầu write tới bản chính. Bản chính sẽ chịu trách nhiệm đánh mã phiên bản, áp dụng các thay đổi này lên ổ đĩa, sau đó chuyển tiếp yêu cầu tới các bản sao chép. Bản con sau đó sẽ bắt đầu áp dụng các thay đổi này lên ổ đĩa và gửi xác nhận hoàn thành tới bản chính. Cuối cùng, bản chính gửi trả lời cho client cũng như cố gắng tự sửa các lỗi gặp phải khi viết lên các bản sao chép.

Nếu lệnh write khiến cho giới hạn khối vượt khả năng lưu trữ tối đa, chương trình client sẽ gỡ lệnh này thành nhiều lệnh khác nhau.

# TÍNH CHỊU LỖI CỦA HỆ PHÂN TÁN

Tính chịu lỗi của hệ thống là một trong những tiêu chí quan trọng nhất được xét đến trong quá trình xây dựng hệ thống khi lỗi phần cứng được xem là thường lệ hơn là ngoại lệ.

Như đã nói ở trên, máy chủ có thể được tái lập bằng các checkpoint và operation log. Bằng việc giữ operation log ngắn, thời gian tái lập máy chủ trở nên nhanh hơn, cũng như bộ nhớ dành cho operation log cũng có thể được giảm tải. Các checkpoint lỗi thời có thể được xóa đi để tiết kiệm dung lượng lưu trữ của máy chủ. Với việc mọi hoạt động máy chủ sẽ chỉ được xác thực khi bản ghi được hoàn thành ở ổ đĩa máy chủ lẫn các máy chủ sao chép, quá trình tái lập lại máy chủ mới có thể được thực hiện song song với các yêu cầu read của client.

Nhờ việc nhân bản các khối dữ liệu trên các máy con khác nhau, các máy con gặp trục trặc sẽ không ảnh hưởng tới khả năng đáp ứng của hệ thống. Khi máy chủ phát hiện một hoặc nhiều bản ghi không thể truy cập do máy con ngừng hoạt động hay thông qua kiểm tra checksum, máy chủ sẽ tìm cách sao chép từ bản ghi vẫn còn hoạt động sang các máy con khác. Một bản ghi có thể bị mất nếu máy chủ không kịp phát hiện đồng loạt các bản sao chép bị hỏng. Khi đó, bản lưu sẽ bị đánh dấu không đáng tin cậy. Client sẽ nhận được thông báo bản lưu bị lỗi thay vì nhận được bản lưu sai.

# KẾT THÚC

GFS là một hệ thống được phát triển bởi Google để đáp ứng các nhu cầu lưu trữ của tổ chức. Hệ thống này được Google giải tán vào năm 2010, thay thế bằng Colossus, với khả năng mở rộng và tối ưu cao hơn, bởi GFS đã đạt tới giới hạn lưu trữ ở thời điểm đó. Cho tới hiện nay, xương sống cho mọi hoạt động của Google vẫn là Colossus, bao gồm tất cả mọi dịch vụ của tập đoàn này: Search, Youtube, Cloud,…