**Báo cáo bài tập lý thuyết môn hệ thống phân tán và ứng dụng**

**IT4611**

Họ và tên: Lưu Tuấn Hùng

MSSV: 20225131

Nội dung báo cáo

**Câu hỏi 1**: Em hãy nêu thêm 2 ví dụ về dịch vụ được coi là Hệ Phân Tán (ngoài 2 ví dụ WWW và Email đã trình bày trên lớp). Dựa vào định nghĩa, giải thích tại sao chúng được coi là Hệ Phân Tán.

1. Hệ thống lưu trữ đám mây: DropBox

Khi làm việc với hệ thống lưu trữ dữ liệu này giống như đang làm việc với 1 thiết bị nhưng đằng sau là một hệ thống gồm tác vụ:

Dữ liệu được lưu trữ trên hiều máy chủ khác nhau.

Khi sử dụng dữ liệu hệ thống truy xuất đến máy chủ gần nhất. Nếu khi đang truyền dữ liệu máy chủ bị lỗi thì sẽ có máy chủ khác thay thế. Quá trình thay thế cũng được diễn ra để người dùng không gặp bất kì vấn đề gì.

1. Dịch vụ Youtube

Tương tự đây cũng là hệ thống lưu trữ dữ liệu dạng video, cũng đảm bảo các chức năng tương tự như trên bằng cả một hệ thống các trung tâm dữ liệu, máy chủ nhưng cho trải nghiệm người dùng như một máy tính duy nhất.

**Câu hỏi 2:** Tại sao nói tính chia sẻ tài nguyên của Hệ Phân Tán có khả năng: Giảm chi phí, tăng tính sẵn sàng và hỗ trợ làm việc nhóm? Tuy nhiên lại tăng rủi ro về an toàn thông tin? Giải thích.

* GIảm chi phí:

+ Giảm chi phí xây dựng máy chỉ do đã có thể thuê, sử dụng hệ thống cloud

+Chia sẻ tài nguyên

* Tính sẵn sàng:

+ Khi máy chủ bị lỗi thì sẽ có máy chủ khác thay thế

+ Khả năng mở rộng người dùng, thêm tài nguyên vào các máy chủ khác nhau

+ Truy cập bất kì đâu với tốc độ tốt do sử dụng máy chủ gần nhất

* Hỗ trợ làm việc nhóm

+ Chia sẻ tài nguyên qua dịch vụ như google drive, chỉnh sửa tài liệu trực tuyến

Tuy nhiên lại tăng rủi ro về an toàn thông tin:

* Dữ liệu do được chia sẻ trên các máy chủ nên có thể dễ bị tấn công
* Nếu không kiểm soát tốt quyền truy cập tài nguyên có thể gây ra lỗ hỏng bảo mật
* Tấn công DDoS: Kẻ tấn công gửi số lượng yêu cầu lớn với mục đích làm gián đoạn dịch vụ

**Câu hỏi 3**: Liên quan đến tính trong suốt, giải thích tại sao nhà quản trị hệ thống phải xem xét việc cân bằng giữa hiệu năng và độ trong suốt? Đưa ra ví dụ cụ thể để giải thích.

Trong một hệ phân tán, tính trong suốt có nghĩa là người dùng hoặc lập trình viên không cần biết hệ thống hoạt động như thế nào bên trong. Tuy nhiên, độ trong suốt cao có thể làm giảm hiệu năng, vì hệ thống phải xử lý nhiều tác vụ bổ sung để che giấu sự phân tán. Vì vậy nhà quản trị cần cân nhắc việc cân bằng giữa hiệu năng và độ trong suốt.

Ví dụ: Khi truy cập vào các trang web

Nếu độ trong suốt cao: Người dùng truy cập website mà không cần biết yêu cầu của họ đang được xử lý bởi máy chủ nào.

Vấn đề hiệu năng: Nếu hệ thống cố gắng giấu hoàn toàn thông tin về máy chủ thực tế, nó có thể phải kiểm tra nhiều bước để đảm bảo yêu cầu được gửi đến máy chủ tốt nhất, gây độ trễ.

**Câu hỏi 5:** So sánh 2 kiểu HĐH DOS và NOS. Giải thích tại sao việc sử dụng Middleware là sự kết hợp ưu điểm của cả 2 mô hình trên.

|  |  |  |
| --- | --- | --- |
| Tiêu chí | DOS | NOS |
| Tính trong suốt | Cao | Thấp |
| Quản lý tài nguyên | Quản lý nhiều máy tính như một máy duy nhất | Các máy độc lập kết nối qua mạng |
| Truy cập tài nguyên | Người dùng truy cập như truy cập 1 máy | Phải chỉ định máy chứa tài nguyên |
| Tính tự trị | Yêu cầu các máy tính phải tương đồng nhau để hệ thống quản lý | Các máy tính độc lập với nhau. Không cần phải giống nhau |
| Hiệu suất | Cao do được quản lý tập trung | Tùy thuộc máy tính và tốc độ kết nối |
| Tính mở rộng | Khó khăn do nhiều yêu cầu về máy tính được thêm vào | Dễ dàng do chỉ cần kết nối qua mạng |

Middleware là sự kết hợp ưu điểm của DOS và NOS

* Có tính trong suốt và quản lý tài nguyên tốt của DOS
* Dễ dàng mở rộng và tự chủ của NOS

**Câu hỏi 6:** Trong mô hình kiến trúc phân tầng OSI của Mạng máy tính, hãy trình bày tóm tắt chức năng của từng tầng. Lấy ví dụ cụ thể khi chúng ta thay đổi/cập nhật một tầng bất kỳ thì không ảnh hưởng đến hoạt động của các tầng khác.

|  |  |
| --- | --- |
| Tầng | Chức năng |
| Tầng ứng dụng. | Cung cấp giao diện cho ứng dụng sử dụng mạng |
| Tầng trình bày. | Mã hóa, nén dữ liệu, đảm bảo dữ liệu có thể đọc được giữa nhiều hệ thống khác nhau. |
| Tầng phiên. | Quản lý phiên làm việc (session), duy trì kết nối giữa hai hệ thống. |
| Tầng giao vận. | Đảm bảo truyền dữ liệu toàn vẹn, kiểm soát lỗi, phân luồng dữ liệu (TCP, UDP). |
| Tầng mạng. | Định tuyến dữ liệu giữa các mạng khác nhau bằng địa chỉ IP. |
| Tầng liên kết dữ liệu. | Kiểm soát lỗi trong truyền dữ liệu giữa hai thiết bị với nhau |
| Tầng vật lý | Truyền dữ liệu dạng bit qua các phương tiện vật lý |

Các tầng hoạt động độc lập, trong suốt với nhau. Làm việc thông qua những lời gọi từ tầng trên xuống tầng dưới và trả lời từ tầng dưới lên.

Ví dụ: Nếu thay đổi đường dẫn tần vật lý từ không dây sang có dây thì các tầng trên không ảnh hưởng.

**Câu hỏi 7:** Cho ví dụ và phân tích một mô hình kiến trúc thuê bao/xuất bản (publish/subscribe).

Thông báo của youtber:

1.Publisher:

Kênh youtube đăng tải video mới

2.Broker:

Youtube gửi thông báo đến những người đăng ký kênh

3.Subcribers:

Người đăng ký kênh nhận được thông báo có video từ kênh đã đăng ký

**Câu hỏi 8:** Sự khác nhau giữa phân tán dọc và phân tán ngang là gì?

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Phân tán dọc | Phân tán ngang |
| Cách tiếp cận | Tăng tài nguyên phần cứng cho một máy tính | Thêm nhiều máy tính vào hệ thống |
| Khả năng mở rộng | Giới hạn bởi phần cứng. | Không giới hạn, có thể thêm node tùy nhu cầu. |
| Chi phí | Đắt đỏ do chi phí nâng cấp phần cứng cao | Linh hoạt, chi phí tăng dần theo số máy tính |
| Độ phức tạp | Dễ triển khai, không cần thay đổi kiến trúc ứng dụng | Phức tạp, yêu cầu thiết kế ứng dụng phân tán. |
| Hiệu suất | Phụ thuộc vào giới hạn phần cứng | Có thể xử lý đồng thời nhiều yêu cầu nhờ phân tải |
| Khả năng chịu lỗi | Thấp: Nếu một máy chính hỏng, hệ thống ngừng hoạt động. | Cao: Nếu một máy hỏng, các máy khác tiếp quản. |
| Ứng dụng điển hình | Cơ sở dữ liệu quan hệ (MySQL). | Hệ thống web |

**Câu hỏi 9:** Phân tích ưu nhược điểm của kiến trúc tập trung và kiến trúc không tập trung

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Kiến trúc tập trung | Kiến trúc không tập trung |
| Quản lý | Tập trung, dễ dàng | Phân tán, phức tạp |
| Khả năng chịu lỗi | Thấp | Cao |
| Khả năng mở rộng | Hạn chế | Linh hoạt |
| Tính nhất quán | Dễ đảm bảo | Khó đảm bảo |
| Chi phí | Thấp so chỉ phù hợp hệ thống nhỏ | Cao vì cần nhiều máy và cấu trúc phức tạp |
| Bảo mật | Dễ kiểm soát nhưng rủi ro cao nếu bị tấn công | Rủi ro phân tán nhưng khó tấn công toàn hệ thống |

Nhược điểm:

+Kiến trúc tập trung dễ gây quá tải ( hiện tượng thắt nút cổ chai ) nếu có quá nhiều người dùng cùng truy cập.

+Kiến trúc không tập trung thì không gây ra hiện tượng cổ chai nhưng khó khan trong việc quản lý tài nguyên.

**Câu hỏi 10:** Trong một mạng overlay có cấu trúc, các thông điệp được định tuyến dựa theo hình trạng mạng (topology). Nhược điểm quan trọng của hướng tiếp cận này là gì?

Nhước điểm là mạng overlay có cấu trúc sẽ cần chi phí bao trì topology cao nhất là khi nhiều node thường xuyên tham gia hoặc rời mạng. Điều này gây giảm hiệu suất và tăng độ phức tạp quản lý hệ thống

**Câu hỏi 11:** Xét một chuỗi các tiến trình P1, P2, ..., Pn triển khai một kiến trúc client-server đa tầng. Cơ chế hoạt động của tổ chức đó như sau: tiến trình Pi là client của tiến trình Pi+1, và Pi sẽ trả lời Pi-1 chỉ khi đã nhận được câu trả lời từ Pi+1.

Vậy những vấn đề nào sẽ nảy sinh với tổ chức này khi xem xét hiệu năng yêu cầu-trả lời tới P1?

* Độ trễ cao do cần gọi các tiến tình lớn hơn để nhận được câu trả lời
* Dễ tắc nghẽn nếu trong chuổi chạy các chương trình có một chương trình chậm lại gây chậm toàn hệ thống
* Lãng phí tài nguyên do các tầng khác đều phải chờ câu trả lời
* Nếu có lỗi ở một tiến trình thì sẽ gây lỗi dây chuyền

**Câu hỏi 12:** Xét mạng CAN như trong hình. Giả sử tất cả các node đều biết node hàng xóm của mình. Một giải thuật định tuyến được đưa ra đó là gửi các gói tin cho node hàng xóm gần mình nhất và hướng đến đích. Giải thuật này có tốt không? Giải thích.

A diagram of a number of keys

AI-generated content may be incorrect.

Ưu điểm:

Dễ triển khai, chỉ cần thông tin về hàng xóm gần nhất.

Tốc độ xử lý nhanh, vì chỉ cần so sánh khoảng cách đến đích.

Hoạt động tốt trong các mạng nhỏ, nơi các node có thể kết nối trực tiếp.

Nhưng có một số vấn đề:

Bẫy cục bộ: Nếu không gian không đồng đều có thể có node ở vị trí cô lập khiến truy vấn bị mắc kẹt

Không tối ưu: Chỉ định tuyến hàng xóm gần nhất không đảm bảo con đường ngắn nhất