

BÀI TẬP TRÊN LỚP

MÔN HỌC: HỆ PHÂN TÁN

CHƯƠNG 1: TỔNG QUAN VÀ KIẾN TRÚC HỆ PHÂN TÁN

HỌ TÊN SV: TRẦN TRUNG PHONG

MSSV: 20210676

MÃ LỚP: 149501

MÃ HỌC PHẦN: IT4611

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.

Trả lời:

Ví dụ về dịch vụ được coi là hệ phân tán:

1. Dịch vụ học tập và kết nối trực tuyến (Microsoft Teams, Zoom...)

Giải thích:

- Mỗi người dùng sử dụng 1 máy tính, thiết bị độc lập nhau và không phụ thuộc nhau.
- Các thiết bị được kết nối với nhau qua mạng Internet, chia sẻ tài nguyên cho nhau như hình ảnh, âm thanh...
- Có giao diện chung thống nhất, người dùng chỉ cần đăng nhập, tham gia họp mặt mà không cần quan tâm tới hệ thống đằng sau.

2. Mạng xã hội (Facebook, Instagram, Twitter...)

Giải thích:

- Mỗi người dùng sử dụng 1 máy tính, thiết bị độc lập và không phụ thuộc lẫn nhau.
- Người dùng sử dụng các máy tính để kết nối với nhau thông qua mạng Internet, chia sẻ tài nguyên cho nhau bằng các phương pháp như nhắn tin chat, chia sẻ file cho nhau.
- Mạng xã hội tất cả người dùng sẽ có 1 giao diện chung thống nhất với nhau về cách thức truy nhập và các dịch vụ. Người dùng cũng không cần quan tâm tới chi tiết hệ thống, chỉ cần đăng nhập và sử dụng.

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.

Trả lời:

- Tính chia sẻ tài nguyên của hệ phân tán có khả năng giảm chi phí và tăng tính sẵn sàng bởi: Hai hay nhiều hệ thống khác nhau có thể khai thác chung 1 data center cùng được đặt tài nguyên ở đó. Vì vậy mỗi hệ thống không cần triển khai riêng mà có thể chia sẻ với nhau.
- Tính chia sẻ tài nguyên của hệ phân tán có khả năng hỗ trợ làm việc nhóm bởi: Các tài nguyên được kết nối với nhau, có chung 1 data center, nên khi một máy chủ gặp sự cố thì vẫn có thể sử dụng máy chủ khác để thực hiện tiếp nhiệm vụ mà không làm gián đoạn quá trình.
- Tuy nhiên lại tăng rủi ro về an toàn thông tin vì:
 - Khi kết nối với nhau, nếu kẻ tấn công giả mạo là 1 trong những máy của hệ phân tán thì có thể truy cập tất cả thông tin trên data center.
 - Khi máy chủ bị tấn công sẽ dẫn đến tê liệt toàn bộ hệ thống

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.

Trả lời:

- Nhà quản trị hệ thống phải xem xét việc cân bằng giữa hiệu năng và độ trong suốt vì nếu hệ thống có độ trong suốt quá cao mà hiệu năng thấp thì khi gặp lỗi ở 1 máy hoặc nhiều máy, hệ thống có hiệu năng thấp khó có thể thực hiện được các cơ chế khắc phục, dẫn đến dịch vụ bị gián đoạn gây ra sự khó chịu cho người sử dụng.
- Ví dụ giải thích về trong suốt thứ lỗi: Khi hệ thống đang stream 1 bộ phim từ 1 máy chủ. Thì máy chủ đấy bị lỗi, hệ thống cần có giải pháp tìm máy chủ thay thế, phải lưu lại thời điểm mà bộ phim đang được phát, để máy chủ mới có thể tiếp tục phát mà không làm ảnh hưởng tới người dùng, làm người dùng không nhận ra. Nếu hệ thống có hiệu năng thấp sẽ khó có thể thực hiện được cơ chế trên.

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.

Trả lời:

- So sánh 2 kiểu hệ điều hành DOS và NOS

| | DOS | NOS |
|----------------|--|--|
| Định nghĩa | Quản lý một số máy tính và thiết bị phần cứng tạo thành một hệ thống phân tán. | Là hệ điều hành máy tính được thiết kế chủ yếu để hỗ trợ máy trạm, máy tính cá nhân và trong một số trường hợp, thiết bị đầu cuối cũ hơn được kết nối trên mạng cục bộ |
| Mục tiêu | Mục tiêu chính của DOS là ẩn và quản lý tài nguyên phần cứng. | Mục tiêu chính của NOS là cung cấp các dịch vụ cục bộ cho các khách hàng từ xa. |
| Sử dụng | DOS là một hệ điều hành kết hợp chặt chẽ cho nhiều bộ xử lý và nhiều máy tính đồng nhất. | NOS là một hệ điều hành được kết hợp lỏng lẻo cho nhiều máy tính không đồng nhất. |
| Kiến trúc | DOS tuân theo kiến trúc Client/server n tầng. | NOS tuân theo kiến trúc Client/server 2 tầng |
| Loại | Hai kiểu của NOS - Peer-to-peer và kiến trúc client / server. | Hai kiểu của DOS - hệ điều hành đa máy tính và hệ điều hành đa xử lý |
| Giao tiếp | Sử dụng tin nhắn để liên lạc | Sử dụng tệp để giao tiếp |
| Tính minh bạch | Cao | Thấp |

- Việc sử dụng Middleware là sự kết hợp ưu điểm của cả 2 mô hình DOS và NOS vì Middleware mang tính chất và ưu điểm như sau:
 - Giao diện lập trình phân tán độc lập với hệ điều hành
 - Tăng tính trong suốt
 - Có các ngôn ngữ được thiết kế đặc biệt cho tính toán phân tán
 - Thường được chạy tại không gian người dùng
 - Tăng mức độ trừu tượng hóa trong lập trình => Ít gây lỗi

- Độc lập với hệ điều hành và các giao thức mạng => Linh hoạt hơn

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.

Trả lời:

- Mô hình kiến trúc phân tầng OSI:
 - Tầng vật lý: Định nghĩa các chức năng cơ, điện để thực hiện kết nối vật lý truyền dòng bit giữa các hệ thống mở. Các phương thức truyền có thể đồng bộ hoặc không đồng bộ; song công, bán song công hoặc đơn công.
 - Tầng liên kết dữ liệu: Thực hiện các chức năng đồng bộ, phát hiện lỗi và điều khiển lỗi, đánh địa chỉ và điều khiển truy cập.
 - Tầng mạng: Thực hiện đánh địa chỉ của lớp mạng, kết nối liên mạng.
 - Tầng giao vận: Đảm bảo chất lượng dịch vụ khi kết nối liên mạng.
 - Tầng phiên: Đồng bộ và quản lý các phiên truyền dẫn mà không làm thay đổi chất lượng dịch vụ của lần truyền tải.
 - Tầng trình diễn: Thực hiện chức năng biểu diễn dữ liệu.
 - Tầng ứng dụng: Cung cấp giao diện sử dụng ứng dụng cho người sử dụng.
- 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: Nếu cần chuyển đổi cú pháp dữ liệu, ta chỉ cần thực hiện thay đổi trong tầng trình diễn mà không cần thay đổi các tầng khác.

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

Trả lời:

- Ví dụ một mô hình kiến trúc thuê bao/xuất bản: JMS.
- Phân tích mô hình JMS:
 - Trong JMS có hỗ trợ mô hình này, theo đó Sender/publisher gửi tin đến 1 topic nào đó và Receiver/subscribers đã truy cập đến topic tương ứng sẽ nhận được tin.
 - Trong mô hình publish/subscribe, sender – publisher không lập trình tin nhắn để gửi trực tiếp đến 1 receiver/subscriber cụ thể, mà thay vào đó, gửi tin đến 1 lớp các subscriber mà không biết cụ thể từng subscriber một. Subscriber tham gia vào một trong các lớp và nhận các message từ đó mà không biết là publisher nào gửi. Mô hình sử dụng Middleware để giao tiếp, không yêu cầu các bên tham gia phải biết rõ các thông tin về đối tác, nên có ưu điểm là tốc độ cao, phù hợp với ứng dụng yêu cầu trao đổi dữ liệu thời gian thực. Mô hình này có thể cung cấp một mạng lưới kết nối rộng lớn do đơn giản trong kết nối, có thể làm việc với lượng thông tin nhiều và có tính động cao do hỗ trợ người dùng từ nhiều nên tảng khác nhau. Tuy nhiên, mô hình có nhược điểm làm giảm sự linh hoạt trong việc sửa đổi publisher và cấu trúc của dữ liệu được publish.

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

Trả lời:

- Phân tán dọc: Tổ chức client-server như một kiến trúc đa tầng, kiểu phân phối xét một cách logic các thành phần khác nhau trên các máy khác nhau, cụ thể hơn đó là các chức năng được chia một cách logic và vật lý trên các máy khác nhau.
- Phân tán ngang: Các client hoặc server được chia vật lý các thành phần tương ứng, mỗi phần được chia sẻ một phần việc riêng được gọi là cơ chế cân bằng tải. Trong phân tán ngang thì các thành phần hoạt động song song với nhau để cùng thực hiện một công việc chức năng.

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

Trả lời:

- Kiến trúc tập trung:
 - Ưu điểm: Quản lý kết nối được tốt hơn, bảo mật cao hơn do có thành phần trung tâm điều phối, giao thức kết nối một cách đơn giản, dễ thực hiện.
 - Nhược điểm: Cân bằng tải kém, thường quá tải do thành phần trung tâm không đáp ứng được tất cả các kết nối cho toàn mạng. Kết nối thường chậm và khả năng tính toán giới hạn.
- Kiến trúc không tập trung:
 - Ưu điểm: Mạng có kích thước lớn, đáp ứng được nhiều người dùng và tốc độ thường nhanh do tính toán, tài nguyên được phân chia trên mạng.
 - Nhược điểm: Giao thức thường phức tạp và khó quản lý mạng, khả năng bảo mật không cao.

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?

Trả lời:

Nhược điểm quan trọng của hướng tiếp cận này là mạng hình dạng overlay phía trên thường không liên quan hay phụ thuộc vào mạng vật lý thực ở dưới nên có thể sẽ khác so với mạng bên dưới. Do đó, các node ở mạng lại xa nhau trên thực tế, vì vậy khiến cho kết nối có thể trở nên phức tạp và tốn kém ở mạng vật lý, dẫn đến chậm trong kết nối và định tuyến.

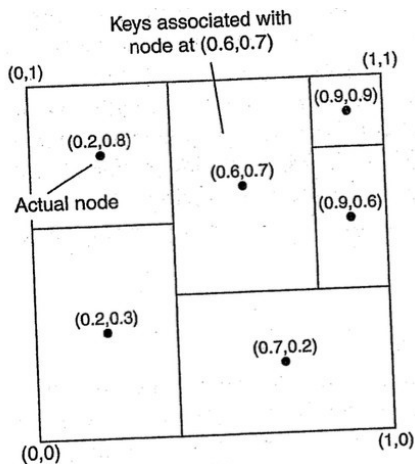
Câu hỏi 11: Xét một chuỗi các tiến trình P_1, P_2, \dots, P_n triển khai một kiến trúc client-server đa tầng. Cơ chế hoạt động của tổ chức đó như sau: tiến trình P_i là client của tiến trình P_{i+1} , và P_i sẽ trả lời P_{i-1} chỉ khi đã nhận được câu trả lời từ P_{i+1} .

Vậy những vấn đề nào sẽ nảy sinh với tổ chức này khi xem xét hiệu năng yêu cầu-trả lời tới P_1 ?

Trả lời:

Khả năng được trả lời của P_1 khi gửi yêu cầu có thể sẽ rất kém. Do để nhận được câu trả lời mong muốn, P_1 có thể sẽ phải đợi toàn bộ các tiến trình phía sau thực hiện xong. Giả sử chỉ một tiến trình bị lỗi hay bị chậm, cũng sẽ ảnh hưởng đến P_1 . Vấn đề thắt nút cổ chai cũng là một trong những nhược điểm lớn của mô hình này.

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.



Trả lời:

Giải thuật này có thể đảm bảo đến một lúc nào đó sẽ kết nối được đến node đích. Tuy nhiên giải thuật này không nhanh trong nhiều trường hợp do node gần nhất thường là các node quản lý các vùng kích thước nhỏ do đó trong đa số trường hợp sẽ cần routing qua nhiều node trước khi tới đích. Thay vào đó nên thực hiện ưu tiên routing đến các node gần đích nhất.

