Họ và tên: Nguyễn Đình Tuấn Anh mssv- 20164767

Bài tập lý thuyết chương 4: Tiền trình và luồng

Câu hỏi 1: Có cần thiết phải giới hạn số lượng các luồng trong một tiến trình server?

Có, vì tôi khi một tiến trình server có thể xảy ra lỗi, chiếm quá nhiều tài nguyên của hệ thống, nếu không giới hạn số lượng luồng có thể gây ra việc lãng phí tài nguyên.

Câu hỏi 2: Có nên chỉ gắn một luồng đơn duy nhất với một tiến trình nhẹ?

Không nên vì nếu tiền trình nhẹ này bị treo thì luồng đơn này không được xử dụng gây lãng phí.

Câu hỏi 3: Có nên chỉ có một tiến trình nhẹ đơn gắn với 1 tiến trình?

Không nên, như thế không tận dụng được ưu điểm của tiến trình nhẹ, đó chính là nếu một tiến trình nhẹ bị ngắt thì cả tiến trình gắn với nó cũng bị ngắt theo.

Câu hỏi 4: Bài toán này yêu cầu bạn so sánh thời gian đọc một tệp (file) của một máy chủ tập tin (file server) đơn luồng và một máy chủ đa luồng. Phải mất tổng cộng 15 ms để nhận 1 yêu cầu (request) và thực hiện quá trình xử lý, giả định rằng các dữ liệu cần thiết nằm ở bộ nhớ đệm trong bộ nhớ chính. Nếu cần thiết phải thực hiện một thao tác truy cập ổ đĩa thì cần thêm 75 ms, biết rằng việc phải thực hiện thao tác này có xắc suất là 1/3. Hỏi máy chủ có thể nhận bao nhiêu yêu cầu/giây trong 2 trường hợp: máy chủ là đơn luồng và máy chủ là đa luồng (ngoài luồng nhận và xử lý request, sẽ có thêm 1 luồng để truy cập ổ đĩa nếu cần thiết)? Giải thích.

Giả sử trung bình cứ trong 3 yêu cầu thì có 1 yêu cầu cần thực hiện thao tác truy cập với ổ đĩa. Tổng thời gian để nhận 3 yêu cầu là 45ms, nhỏ hơn 75ms. Vậy với máy chủ đa luồng, trong một giây ta có thể phục vụ khoảng 1000/75 \*3 = 1000/25 =40 yêu cầu. Còn với máy chủ đơn luồng, với 3 yêu cầu ta mất 120ms suy ra trong một giây ta chỉ phục vụ được khoảng 24 yêu cầu.

Câu hỏi 5: Hệ thống X chỉ định máy của user chưa server, trong khi các ứng dụng lại được coi như client. Điều đó có vô lý không? Giải thích.

Trong hệ thống X Window, trình chủ chạy trên máy tính của người dùng, trong khi các trình khách có thể lại chạy trên các máy tính khác. Quan điểm này ngược với những cấu hình khách-chủ thông dụng trong các hệ thống thường thấy. Thông thường, *trình khách* chạy trên máy tính của người dùng, trong khi *trình chủ* lại chạy trên một máy tính ở xa. Sự đảo lộn này thường gây ra sự lúng túng cho những người mới dùng X. Thuật ngữ mà "X Window" dùng được lấy ra từ quan điểm của chương trình ứng dụng, hơn là từ góc độ của người dùng, hay từ phần cứng: trình ứng dụng ở xa kết nối với bộ hiển thị của trình chủ X đang chạy trên máy tại địa phương, và do đó nó hoạt động như những trình khách. Bộ hiển thị X tại máy địa phương chấp nhận giao thông dữ liệu hướng về mình, cho nên nó hoạt động như một trình chủ.

Câu hỏi 6: Giao thức thiết kế cho hệ thống X gặp phải vấn đề về tính mở rộng. Chỉ ra các giải pháp để giải quyết vấn đề đó?

Một lý do có thể là nếu nhìn vào lưu lượng truyền qua lại giữa máy khách và máy chủ, thì nó khá dài dòng. Điều này không gây ra vấn đề khi lưu lượng truy cập chỉ phải đi cục bộ trên một máy duy nhất, tuy nhiên khi lưu lượng cần đi qua kết nối mạng, thì điều đó trở nên rõ ràng hơn là giao thức không hiệu quả. Giao thức có thể chấp nhận được trên mạng LAN, nhưng ngay khi thử và mở rộng nó qua kết nối WAN hoặc giới thiệu mã hóa dưới dạng VPN hoặc bằng cách sử dụng kết nối SSH làm liên kết giữa máy khách và máy chủ, Giao thức thực sự bắt đầu cho thấy nó thiếu khả năng mở rộng.

Câu hỏi 7: Với việc xây dựng một server đồng thời, hãy so sánh việc server này tạo một luồng mới và tạo một tiến trình mới khi nhận được yêu cầu từ phía client.

Với 2 cách làm này thì đều có ưu và nhược điểm, ví dụ với việc tạo ra một

Luồng mới khi có một request thì việc này đáp ứng được nhanh yêu cầu từ client, tuy nhiên dễ gặp phải vấn đề quá tải và mất quá nhiều thời gian cho việc tạo và huỷ luồng. Với việc tạo ra một tiến trình khi nhận được request thì đây là trường hợp cha của trường hợp trước, việc tạo bao nhiêu luồng làm việc là do tiến trình này lựa chọn, ví dụ với các tiến trình yêu cầu các thao tác nặng tốn thời gian thì có thể khởi tạo nhiều luồng hơn. Cách làm này linh động hơn tuy nhiên cũng dễ gây ra tắc nghẽn và tốn tài nguyên khi hệ thống nhận được nhiều request một lúc.

Câu hỏi 8: Nếu bây giờ một webserver tổ chức lưu lại thông tin về địa chỉ IP của client và trang web client đó vừa truy cập. Khi có 1 client kết nối với server đó, server sẽ tra xem trong bảng thông tin, nếu tìm thấy thì sẽ gửi nội dung trang web đó cho client. Server này là có trạng thái (stateful) hay không trạng thái (stateless)?

Server này là có trạng thái do có lưu lại các thông tin để phục vụ người dùng.

Câu hỏi 9: So sánh Docker và Virtual Machine.

Docker : sử dụng chung một nhân kernel

VM: Sử dụng các OS khác nhau.

Docker: sử dụng ít tài nguyên

VM: Ngốn tài nguyên

Docker: Nhanh và nhẹ

VM: Nặng và chậm

Docker: lưu file thực thi và các dependencies.

VM: Hoạt động như trên một môi trường hệ điều hành khác.