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

Có cần thiết phải giới hạn số lượng các luồng trong một tiến trình. Vì số lượng các luồng ảnh hưởng đến hiệu năng và chi phí lập trình.

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

Không nên chỉ gắn một luồng đơn duy nhất với một tiến trình nhẹ.

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

Không nên chỉ có một tiến trình nhẹ đơn gắn với một tiến trình.

**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.**

Server đơn luồng có thể xử lý 1/(0.015+0.075/3) = 25 yêu cầu /s

Server đa luồng có thể xử lý 1/0.015 = 67 yêu cầu /s

**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.**

Không. Vì máy của user có thể làm server để các máy khác truy cập, trong khi các ứng dụng của máy này có thể coi là client, truy cập đến các máy khác.

**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?**

- Không thêm một chức năng mới nào trừ khi người thực hiện không thể hoàn thành một ứng dụng thật sự nếu thiếu chức năng đó.

- Việc quyết định hệ thống không phải là gì quan trọng không kém việc quyết định hệ thống là gì.

- Không phục vụ nhu cầu của cả thế giới, thay vào đó, tạo cho hệ thống có khả năng mở rộng được, sao cho các tính năng bổ sung về sau có thể được đáp ứng theo kiểu tương thích xuôi (tương thích với các phiên bản sau).

- Điều duy nhất tồi tệ hơn việc tổng quát hóa từ một ví dụ là tổng quát hóa không từ một ví dụ nào cả.

Nếu không thể hiểu một vấn đề một cách trọn vẹn, thì có lẽ tốt nhất là đừng nên đưa ra một giải pháp nào hết."

Nếu ta có một giải pháp có thể đạt được 90% hiệu ứng mong muốn cho 10% công việc, thì ta nên chọn giải pháp đơn giản hơn.

Cô lập các phức tạp một cách hết mức có thể."

- Cung cấp cơ chế chứ không cung cấp qui chế. Cụ thể, đặt các qui chế về giao diện người dùng vào tay của trình khách.

Trong thời gian thiết kế X, nguyên lý đầu tiên đã được điều chỉnh thành: Không bổ sung tính năng mới, trừ phi bạn biết một ứng dụng cụ thể nào đó sẽ đòi hỏi tính năng đó.

**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.**

Server đơn luồng:

+ Chỉ xử lý được một yêu cầu tại một thời điểm

+ Các yêu cầu có thể được xử lý tuần tự

+ Các yêu cầu có thể được xử lý bởi các tiến trình khác nhau

+ Không đảm bảo tính trong suốt

Server đa luồng:

+ Các yêu cầu mới cần chờ quá trình xử lý yêu cầu hiện tại kết thúc.

+ Lãng phí thời gian chờ đợi.

**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)?**

Đó là server không trạng thái.

**Câu hỏi 9: So sánh Docker và Virtual Machine:**

Giống nhau : Đều là công nghê ảo hóa

Khác nhau :

+ VMware mô phỏng phần cứng ảo và phải tính đến tất cả các yêu cầu hệ thống cơ bản-sau đó, hình ảnh máy ảo lớn hơn đáng kể so với các container. Điều dó nói rằng cũng có thể chạy nhiều phiên bản hệ điều hành kín đáo song song trên một máy chủ duy nhất với VMWare cho phép các tổ chức xây dựng các giải pháp IaaS thật sự tại nhà.Khả năng di chuyển máy và cách ly lớn hơn, hãy đi với VMware

+ Docker không tạo ra toàn bộ hệ điều hành ảo - thay vào đó, tất cả các thành phần yêu cầu không chạy trên máy chủ được đóng gói bên trong container với ứng dụng. Kể từ khi hạt nhân máy chủ được chia sẻ giữa các container Docker, các ứng dụng chỉ vận chuyển với những gì họ cần để chạy-không nhiều hơn, không kém. Điều này làm cho ứng dụng Docker dễ dàng và nhẹ hơn để triển khai và khởi động nhanh hơn so với các máy ảo. Các container của Docker thường nhanh hơn và ít tốn nhiều tài nguyên hơn so với các máy ảo.