

BÀI TẬP TRÊN LỚP
MÔN HỌC: HỆ PHÂN TÁN CHƯƠNG
3: TIẾN TRÌNH VÀ LUỒNG

HỌ TÊN SV:
MÃ LỚP:

MSSV:
MÃ HỌC PHẦN:

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

Trả lời:

Cần giới hạn số lượng các luồng trong một tiến trình server vì nó gây tốn bộ nhớ. Cần cân bằng giữa chi phí lập trình và hiệu năng

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

Trả lời:

Không. Khi các tiến trình nhẹ ánh xạ 1-1 với các luồng tăng user. Khi luồng đó bị block, sẽ block tiến trình nhẹ tương ứng đó

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

Trả lời:

Không. Có thể tùy biến để phát triển để giải quyết bài toán về block system call

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 xác 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.

Trả lời:

- Đơn luồng:

Thời gian để nhận 1 yêu cầu: $15 \times 2/3 + (15+75) \times 1/3 = 40$ (ms)

⇒ Số yêu cầu/giây: $1000/40 = 250$ yêu cầu

- Đa luồng:

⇒ Số yêu cầu/giây: $1000/15 \approx 67$ 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.

Trả lời:

Điều đó không vô lý. Các ứng dụng sẽ sinh ra các tiến trình, gửi các yêu cầu cho máy user để xử lý và được coi như là Client

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?

Trả lời:

- Cải thiện giao thức X protocol
- Nén thông điệp

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.

Trả lời:

- Tạo tiến trình tốn tài nguyên hơn so với tạo luồng
- Tạo tiến trình mới => Xử lý tách biệt tài nguyên => Đơn giản hơn
- Nếu muốn các collection trao đổi với nhau thì nên tạo ra luồng

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)?

Trả lời:

Server có trạng thái:

- Server lưu dữ liệu của client, điều đó đồng nghĩa với việc ràng buộc giữa client và server vẫn được giữ sau mỗi request (yêu cầu) của client. Data được lưu lại phía server có thể làm input parameters cho lần kế tiếp.

Câu hỏi 9: So sánh Docker và Virtual Machine.

Trả lời:

Virtual Machine	Docker
Cô lập quy trình cấp phần cứng	Cô lập quy trình cấp OS
Mỗi máy ảo có 1 OS riêng	Mỗi container có thể chia sẻ OS
Khởi động trong vài phút	Khởi động trong vài giây
VMs có dung lượng vài GB	Containers có dung lượng nhỏ: KBs/MBs
Ready-made VMs khó tìm	Dễ dàng có sẵn Pre-built docker containers
Chuyển qua máy chủ mới dễ dàng	Containers bị hủy và tạo lại khi di chuyển
Tạo máy ảo mất nhiều thời gian	Tạo Containers trong vài giây
Sử dụng nhiều tài nguyên	Sử dụng ít tài nguyên