

BÀI TẬP TRÊN LỚP

MÔN HỌC: HỆ PHÂN TÁN

CHƯƠNG 3: ĐỊNH DANH TRONG HỆ PHÂN TÁN

HỌ TÊN SV: NGUYỄN THANH HÀ

MSSV: 20210298

MÃ LỚP: 154056

MÃ HỌC PHẦN: IT4611

Câu hỏi 1: Tại sao không thể lấy địa chỉ của Access Point để sử dụng như địa chỉ của thực thể?

Trả lời:

Không thể lấy địa chỉ của Access Point để sử dụng như địa chỉ của thực thể vì:

- Một thực thể có nhiều địa chỉ và có thể thay đổi địa chỉ nếu thay đổi vị trí địa lý.
- Một thực thể có thể thay đổi Access Point và ngược lại, Access Point có thể gắn vào 1 thực thể khác.
- Nếu thực thể có nhiều Access Point thì sẽ không rõ địa chỉ để tham chiếu.

Câu hỏi 2: Với việc sử dụng Định Danh, các vấn đề gì cần phải xem xét?

Trả lời:

Với việc sử dụng định danh, các vấn đề cần phải xem xét là:

- Về sự tồn tại của thực thể: nếu thực thể không tồn tại, liệu có sự định danh không?
- Sự trùng lặp khi sử dụng lại các định danh.
- Không gian định danh là có giới hạn → Tương lai sẽ không đủ.

Câu hỏi 3: Xét một thực thể di chuyển từ vị trí A sang vị trí B. Trong quá trình di chuyển thực thể đó có đi qua các nút trung gian nhưng chỉ dừng lại ở đó khoảng thời gian ngắn. Khi đến B, thực thể đó dừng lại. Chúng ta biết rằng việc thay đổi địa chỉ trong một dịch vụ tổ chức vị trí phân cấp (hierarchical location service) là rất mất thời gian để hoàn thành, vì vậy cần tránh làm việc này khi thực thể tạm dừng ở các nút trung gian. Hãy đề xuất một mô hình kết hợp cả dịch vụ tổ chức vị trí phân cấp và cơ chế chuyển tiếp con trỏ (forwarding pointers) để có thể xác định được vị trí của thực thể khi nó ở các nút trung gian.

Trả lời:

Đề xuất mô hình: Mô hình kết hợp cả dịch vụ tổ chức vị trí phân cấp và cơ chế chuyển tiếp con trỏ để có thể xác định được vị trí của thực thể khi nó ở các nút trung gian

- Khi thông tin dịch chuyển vị trí trung gian A và B, để lại con trỏ khi đến B ghi thêm địa chỉ mới đó.
➔ Mô hình phân cấp
- Chuỗi con trỏ trung gian và địa chỉ A đều bị xóa.

Câu hỏi 4: Trình bày một số phương pháp ARP Spoofing để thấy được điểm yếu của phương pháp định danh sử dụng cơ chế quảng bá.

Trả lời:

Một số phương pháp ARP Spoofing:

- Tấn công xen giữa
- Mac Proofing
- Sesion Hijacking
- Nghe trộm, chặn bắt gói tin

Câu hỏi 5: Vấn đề còn tồn tại đối với cơ chế *chuyển tiếp con trỏ* (Forwarding Pointer) là gì?

Trả lời:

Vấn đề còn tồn tại với cơ chế chuyển tiếp con trỏ:

- Chuỗi dài vô hạn:
 - Giải pháp: sử dụng các shortcut
 - Có khả năng có chuỗi không tham chiếu được
- Lưu trữ vô số các tham chiếu:
 - Giải pháp: loại bỏ các tham chiếu
 - Bài toán toàn cục
- Nếu một 1 con trỏ bị hỏng thì sẽ không bao giờ tìm được chuỗi trên đó nữa

Câu hỏi 6: Nhược điểm của giải pháp Home-based là gì? Giải pháp nào để giải quyết nhược điểm đó?

Trả lời:

Nhược điểm của giải pháp Home-base

- Mỗi thực thể gắn với một home-agent
- Mỗi khi di chuyển sẽ tự động cập nhật home-agent
- Khi thực thể di chuyển đủ xa home-agent và đủ lâu, nó sẽ được gán bằng home-agent mới gần nó hơn

Câu hỏi 7: Khi áp dụng giải pháp sử dụng hàm băm phân tán vào hệ thống Chord thì nó đã tối ưu cơ chế định danh như thế nào?

Trả lời:

Áp dụng giải pháp sử dụng hàm băm phân tán vào hệ thống Chord

- Thiết lập vòng bằng các biến cục bộ $prev(n)$ và $succ(n)$

- Sử dụng bảng băm để xác định chỉ succ(k) của tên k
 - Xét theo vòng, chiều kim đồng hồ
 - Giá trị nằm giữa hai khóa liên tiếp nào thì chọn khó bé hơn
 - Sau khi không chọn được ở giữa nào nữa thì đi thẳng tới khóa k
 - Ở các vòng tiếp theo, “độ lớn” xếp theo thứ tự (ví dụ 1>28)

Câu hỏi 8: Trong *giải pháp phân cấp*, sử dụng cơ chế bộ đệm có tác dụng cải thiện hiệu năng như thế nào? Cho ví dụ.

Trả lời:

- Trong giải pháp phân cấp, sử dụng cơ chế bộ đệm có tác dụng cải thiện hiệu năng:
 - Các kết quả được lưu lại vào bộ đệm cache
 - Lần tới có thể truy xuất ngay mà không cần tính toán, tìm kiếm...
- Ví dụ bài toán tìm kiếm:

Sử dụng bộ đệm	Không sử dụng bộ đệm
<ul style="list-style-type: none"> - Đi từ lá -> node cha -> root -> Xuống tới khi tìm thấy thì dừng - Không lưu trữ kết quả - Lần tới lặp lại các bước như trên tới khi tìm thấy 	<ul style="list-style-type: none"> - Đi từ lá -> node cha -> root -> Xuống tới khi tìm thấy thì dừng - Lưu kết quả tìm được vào bộ đệm - Lần tới truy xuất ngay kết quả thay vì tìm lại từ đầu

➔ Nếu sử dụng bộ đệm cache thì từ lần tìm kiếm thứ hai sẽ không mất nhiều thời gian tìm kiếm

Câu hỏi 9: So sánh *liên kết vật lý* và *liên kết biểu tượng* trong hệ thống quản lý tệp của UNIX.

Trả lời:

Liên kết vật lý	Liên kết biểu tượng
<ul style="list-style-type: none">- Một liên kết vật lý là một quan hệ giữa tên tệp trong thư mục với một inode- Có thể có nhiều liên kết vật lý với 1 inode- Liên kết cuối cùng tới inode bị xóa thì các khối liên quan tới inode cũng bị xóa	<ul style="list-style-type: none">- Khi tạo liên kết biểu tượng, một inode mới được tạo ra- Inode này chứa tên của phần tử được trỏ tới- Khi xóa một file, các file khác không ảnh hưởng

Câu hỏi 10: Khi chúng ta thêm 1 node mới vào hệ thống Chord, chúng ta có cần phải cập nhật toàn bộ các bảng finger?

Trả lời:

Không cần phải cập nhật toàn bộ các bảng finger, chỉ cần cập nhật cho nút sau và nút trước vì trong hệ thống Chord, mỗi nút sẽ quản lý một khóa của nút trước đó

Câu hỏi 11: *Phân giải tên đệ qui* có ưu điểm gì so với *phân giải tên không đệ qui*?

Trả lời:

Ưu điểm của phân giải tên đệ qui so với phân giải tên không đệ qui

Có đệ quy	Không đệ quy
<ul style="list-style-type: none">- Xử lý nhiều ở bên server	<ul style="list-style-type: none">- Xử lý nhiều ở bên client

<ul style="list-style-type: none"> - Các name server tổ chức lưu trữ cache - Server lưu trữ kết quả - Có thể trả lời nhanh cho client 	<ul style="list-style-type: none"> - Nếu client ở xa độ trễ tăng do việc gửi đi gửi lại
--	--