**BÀI TẬP LÝ THUYẾT**

**MÔN HỌC: HỆ PHÂN TÁN**

**CHƯƠNG 3: ĐỊNH DANH TRONG HỆ PHÂN TÁN**

**Họ tên SV: Lê Thị Yên MSSV: 20183861**

**Mã lớp: 128745**

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

* Một thực thể có thể liên kết nhiều hơn một Access Point (ví dụ máy tính có thể cấu hình cho nó kết nối với 2 mạng khác nhau được)
* Một thực thể có thể dịch chuyển qua lại các Access Point (ví dụ máy tính chuyển từ mạng này qua mạng khác…)
* Một Access Point có thể kết nối với nhiều thực thể khác nhau
* Không thể dùng địa chỉ Access Point để định danh chính xác 1 thực thể

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

* Các vấn đề cần phải xem xét khi sử dụng Định danh: Với quan hệ 1:1 ( 1 Định danh mãi mãi chỉ trỏ đến 1 thực thể và 1 thực thể chỉ được xác định bởi 1 định danh) thì khi số lượng thực thể nhiều mà không gian định danh hữu hạn -> Cạn kiệt không gian tên
* Giải pháp:

+ Mở rộng không gian tên

+ Tái sử dụng định danh -> có khả năng bị trùng

**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.**

* Khi thông tin dịch chuyển vị trí trung gian giữa A & B, để lại con trỏ khi đến B ghi thêm địa chỉ mới đó -> mô hình phân cấp
* Chuỗi con trỏ trung gian bị xóa đi, đỉa chị ở A cũng 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.**

* ARP spoofing là một cuộc tấn công Man in the Middle cho phép những kẻ tấn công chặn giao tiếp giữa các thiết bị mạng. Kẻ tấn công ARP spoofing giả vờ là cả hai bên tham gia của một kết nối mạng. Khi kẻ tấn công giả mạo ARP, chúng có thể: Tiếp tục định tuyến thông tin liên lạc như hiện tại, kẻ tấn công có thể đánh hơi (sniffing) các gói tin và đánh cắp dữ liệu, bất kỳ lưu lượng truy cập gửi cho địa chỉ ip của một máy chủ sẽ bị chuyển sang cho kẻ tấn công.
* Một số phương pháp ARP Spoofing:

+ Man-in-the-middle.

+ MAC Spoofing.

+ Session Hijacking….

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

Vấn đề còn tồn tại đối với cơ chế chuyển tiếp con trỏ (Forwarding Pointer) là: không có phạm vi cụ thể để thực hiện, tất cả các vị trí trung gian trong một chuỗi sẽ phải duy trì một phần của nó trong chuỗi Forwarding Pointer, liên kết có thể bị phá vỡ

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

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

* Giải pháp Home-based: mỗi thực thể được gán cho 1 vị trí gốc, vị trí gốc này giữ dấu vị trí hiện thời của thực thể. Client giao tiếp với vị trí gốc để xác định xem thực thể thực sự nằm ở đâu.
* Nhược điểm:có thể gây ra trễ mạng, và nếu thực thể quyết định đổi vị trí gốc thì có thể dẫn đến việc mất dấu vĩnh viễn.

**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?**

Chord sử dụng một hệ thống không gian định danh m-bit để gán lựa chọn ngẫu nhiên cho nốt cũng như các khóa cho các thực thể cụ thể. Một thực thể với khóa k thuộc quyền hạn của các node được gọi là nốt kế nhiệm của k là succ(k). Phải sử dụng bảng băm để xác định địa chỉ succ(k) của tên k.

Việc duy trì bảng ánh xạ giữa khóa và các giá trị được lưu phân tán trên các nút. Khi đó việc thay đổi một số nút tham gia vào hệ thống chỉ ảnh hưởng đến số nhỏ các khóa liên quan

**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.**

Với giải pháp phân cấp, mạng được chia ra thành các domain không chồng lên nhau, chúng có thể được nhóm lại để tạo nên các domain mức cao hơn. Thông tin được tìm theo các nút từ lá => node cha => root, tới khi nào tìm thấy thì dừng lại.

Như vậy sử dụng cơ chế bộ đệm có tác dụng cải thiện hiệu năng: các kết quả được lưu lại vào cache.Và lần tới có thể truy suất ngay mà không cần tính toán, tìm kiếm, ...

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

* **Hard links** là các liên kết cấp thấp ( low-level links) mà hệ thống sử dụng để tạo các thành phần của chính hệ thống file, chẳng hạn như file và thư mục. Liên kết cứng sẽ tạo một liên kết trong cùng hệ thống tập tin với 2 inode entry tương ứng trỏ đến cùng một nội dung vật lý (cùng số inode vì chúng trỏ đến cùng dữ liệu).
* **Symbolic links** là một file đặc biệt trỏ đến một file hoặc thư mục khác - được gọi là **target**. Khi được tạo, một symbolic links có thể được sử dụng thay cho target file. Nó có thể có một tên độc nhất, và được đặt trong bất kỳ thư mục nào. Nhiều symbolic links thậm chí có thể được tạo cho cùng một target file, cho phép truy cập target bằng nhiều tên khác nhau.

|  |  |
| --- | --- |
| **Liên kết vật lý** | **Liên kết biểu tượng** |
| Chỉ liên kết được tới file, không liên kết được tới thư mục | Có thể liên kết được tới thư mục |
| Không tham chiếu được tới file trên ổ đĩa khác | Có thể tham chiếu tới file/thư mục khác ổ đĩa |
| Liên kết tới một file vẫn còn ngay cả khi file đó đã được di chuyển | Liên kết không còn tham chiếu được nữa nếu file được di chuyển |
| Được liên kết với inode tham chiếu vật lý trên ổ cứng nơi chứa file | Liên kết tham chiếu tên file/thư mục trừu tượng mà không phải địa chỉ vật lý. Chúng được cung cấp inode riêng của mình |
| Có thể làm việc với mọi ứng dụng | Một số ứng dụng không cho phép symbolic link |

**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?**

Diagram

Description automatically generated

* Không cần phải cập nhật toàn bộ các bảng finger
* Trong hệ thống Chord, các nút màu xám là các nút quản lý khóa, nút màu trắng là các khóa. Các khóa được quản lý bởi các nút có id nhỏ nhất mà lớn hơn nó. Khi thêm 1 node vào hệ thống thì chỉ cần cập nhập cho nút trước và nút sau nó, không cần cập nhập cả hệ thống.

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

Ưu điểm của phân giải có đệ qui so với phân giải không đệ qui:

* Phân giải tên có đệ qui xử lý nhiều ở bên server, phân giải không đệ qui xử lý nhiều bên client. Khi client ở xa, phân giải không đệ qui phải gửi đi gửi lại dẫn tới độ trễ cao.
* Phân giải tên có đệ qui các name server tổ chức lưu trữ bộ đệm, server lưu trữ kết quả, có thể trả lời luôn và nhanh cho các client