

BÀI LÝ THUYẾT SỐ 3

CÁC HỆ THỐNG PHÂN TÁN VÀ ỨNG DỤNG

CHƯƠNG 3: ĐỊNH DANH TRONG HỆ PHÂN TÁN

Họ và tên: Nguyễn Duy Khánh

Mã lớp: 157542

MSSV: 20225019

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?

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

- Một thực thể có thể gắn với nhiều Access Point, các AP có thể triển khai các thực thể khác
- 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 di chuyển qua các Access Point, có thể di chuyển từ tiến trình này sang tiến trình khác, máy tính này sang máy tính khác, di chuyển qua lại giữa các mạng máy tính khác nhau

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

Các vấn đề cần xem xét khi sử dụng định danh là:

- Nếu thực thể không còn tồn tại thì có thể có định danh cho thực thể không
- Sự trùng lặp khi sử dụng lại các định danh, gây khó khăn trong quản lý, tương tranh
- Cạn kiệt không gian tên

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.

Ta sẽ kết hợp dịch vụ định vị phân cấp với cơ chế con trỏ chuyển tiếp. Khi thực thể bắt đầu di chuyển, nó sẽ để lại một con trỏ chuyển tiếp tại vị trí A trở đến vị trí tiếp theo (trung gian). Mỗi lần di chuyển tiếp theo, một con trỏ chuyển tiếp mới được để lại. Khi đến vị trí B, thực thể cập nhật địa chỉ mới của mình vào dịch vụ định vị phân cấp. Sau đó, chuỗi các con trỏ được dọn dẹp, và địa chỉ tại vị trí A bị xóa bỏ.

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 còn được gọi là ARP poisoning, là một cuộc tấn công Man in the Middle (MitM) cho phép những kẻ tấn công chặn giao tiếp giữa các thiết bị mạng.

Một số phương pháp ARP Spoofing khai thác sự thiếu chứng thực trong giao thức ARP bằng cách gửi thông tin ARP giả mạo vào mạng LAN:

- Tấn công xen giữa: Bí mật chuyển tiếp và có thể làm thay đổi giao tiếp giữa hai bên mà họ tin rằng họ đang trực tiếp giao tiếp với nhau
- Mac Proofing: Kỹ thuật điều chỉnh địa chỉ MAC để giả dạng người dùng chính thống hoặc tạo tấn công như từ chối dịch vụ. Kẻ tấn công nghe trộm địa chỉ MAC kích hoạt trên switch port của người dùng và sao lại. Sao lại địa chỉ MAC hướng giao thông mạng đến kẻ tấn công thay vì địa điểm ban đầu.
- Session Hijacking: chiếm quyền điều khiển của người dùng internet dựa vào đó kẻ tấn công có thể đánh cắp tiền từ tài khoản ngân hàng của người dùng, mua hàng, lấy dữ liệu cá nhân để thực hiện hành vi trộm cắp ID hoặc mã hóa dữ liệu quan trọng và yêu cầu tiền chuộc để trả lại.
- Nghe trộm, chặn bắt gói tin: kẻ tấn công kết nối độc lập với các nạn nhân và chuyển tiếp thông tin giữa họ để họ tin rằng họ đang nói chuyện trực tiếp với nhau qua kết nối riêng tư, trong khi thực ra toàn bộ cuộc trò chuyện được kiểm soát bởi kẻ tấn công
- DOS: Server nhận được quá nhiều yêu cầu, lấy định danh quảng bá khắp nơi dẫn đến lỗi cơ chế phân giải địa chỉ

Điểm yếu của cơ chế định danh sử dụng quảng bá: Giao thức ARP hoạt động dựa trên cơ chế quảng bá để ánh xạ địa chỉ IP sang địa chỉ MAC mà không có bất kì cơ chế xác thực nào. Vì vậy, kẻ tấn công có thể gửi các gói ARP giả mạo và dễ dàng thao túng bảng ARP của các thiết bị trong mạng. Sự thiếu hụt xác thực và cơ chế kiểm tra tính hợp lệ của thông tin nhận được khiến ARP trở thành mục tiêu dễ bị tấn công trong các mạng LAN.

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

Nếu hệ thống có nhiều thực thể, theo thời gian các thực thể di chuyển càng nhiều, thì chuỗi con trỏ mà hệ thống quản lý ngày càng lớn

Nếu chỉ cần một nút trung gian chết (hỏng) thì tất cả những con trỏ mà đi qua nút đó đều bị mất thông tin => dẫn đến không bao giờ tìm đến được các thực thể mà đi qua nút đó

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

Nhược điểm của giải pháp Home-based là vấn đề về quy mô:

- Thời gian hệ thống càng lâu thì các thực thể có thể truy cập vào rất xa home agent

- Vấn đề: client rất gần với thực thể đang tìm kiếm mà phải nhận và gửi yêu cầu rất xa đến phía home location. Mỗi khi muốn liên lạc với một thực thể, client phải liên lạc trước tiên với home agent, dù nó ở rất xa

Giải pháp: Nếu một thực thể đã dịch chuyển sang vị trí đủ xa so với Home Agent và dịch chuyển quanh vị trí đó đủ lâu thì người ta sẽ gán cho thực thể đó một Home Agent mới ở gần vị trí hiện tại của thực thể đó 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?

Số lượng bản ghi cho 1 finger table là $\log_2 n$ với n là kích cỡ không gian tên.

Một cặp khóa và giá trị được lưu trong DHT, bất cứ nút nào tham gia vào hệ thống đều có thể lấy được giá trị ứng với một khóa xác định. Chord sử dụng phương pháp băm nhất quán (consistent hashing) gián tiếp thực hiện việc cân bằng tải giữa các nút. Việc tham gia hay rời khỏi mạng sẽ chỉ làm cho một số khóa chuyển từ nút này sang nút khác, do đó khả năng mở rộng qui mô tương đối đơn giản. Nó cung cấp khả năng tìm kiếm nhanh chóng và hiệu quả, tăng cường khả năng mở rộng và tính linh hoạt của hệ thống.

Khi cần tìm khóa k nằm giữa hai giá trị liên tiếp (hai bản ghi liên tiếp của table) thì yêu cầu sẽ gửi đến nút q là nút có giá trị nhỏ hơn (bản ghi đứng trước).

$$a = FT_{p[j]} < k < FT_{p[j+1]}$$

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.

Trong giải pháp phân cấp, sử dụng cơ chế bộ đệm có tác dụng cải thiện hiệu năng để tối ưu hóa quá trình tìm kiếm, hệ thống có thể sử dụng bộ đệm. Giả sử nút A cần tìm kiếm nút B nhưng do quá xa nhau nên yêu cầu tìm kiếm sẽ được gửi từ nút A cho directory node của Domain ngay trên nó, cứ như vậy cho đến khi tìm được Directory Node biết được thông tin của nút B rồi forward xuống cho nút đó. Quá trình tìm kiếm như vậy sẽ phức tạp, tốn hiệu năng. Giải pháp sử dụng bộ đệm như sau: Nếu thực thể cứ dịch chuyển ổn định quanh domain thì nút Requester sau khi tìm kiếm theo đường vòng tìm được thực thể này thì nút này sẽ lưu trữ địa chỉ của Directory Node của Domain đó để lần sau không cần vòng lên trên hỏi mà gửi thẳng yêu cầu tới Directory lúc này để xem thực thể cần tìm nằm ở chỗ nào

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

Liên kết vật lý	Liên kết biểu tượng
Một liên kết vật lý là một quan hệ giữa tên tệp trong thư mục với một inode	Không trở thẳng trực tiếp tới inode cũ mà tạo ra 1 inode riêng như bình thường, chứa thông tin đường dẫn tuyệt đối đến file gốc

Cho phép nhiều hơn một liên kết trở tới nó. Tạo liên kết trực tiếp đến nút gốc	Nút liên kết chỉ chứa thông tin đường liên kết tới nút mà nó muốn liên kết.
Khi tạo thêm liên kết vật lí sẽ làm tăng con đếm tham chiếu tới tệp đó	Không làm tăng con đếm tham chiếu
Sau khi tạo xong file liên kết thì không còn sự phân biệt giữa file gốc và file liên kết	Có sự phân biệt giữa file gốc và file liên kết. Có thể xóa file liên kết và mở file gốc bình thường nhưng nếu xóa file gốc thì không mở được file liên kết nữa.

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?

Khi chúng ta thêm 1 node mới vào hệ thống Chord, chúng ta 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. Sau khi tham gia mạng, nút mới sẽ thông báo cho các nút có thể bị ảnh hưởng bởi sự xuất hiện của nó để cập nhật bảng finger nếu cần thiết.

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

Phân giải tên đệ quy có ưu điểm hơn hẳn phân giải tên không đệ quy.

Trong trường hợp Client Name Resolver nằm cách xa máy chủ tên miền: Ở các dải tên không đệ quy thì Client Name Resolver phải liên tục tạo ra các cặp yêu cầu – trả lời đến các máy chủ từ cao xuống thấp cho tới khi phân giải được tên miền. Còn ở phân giải đệ quy, Client Name Resolver chỉ cần gửi 1 cặp yêu cầu – trả lời, điều này thích hợp nếu Client và các Server cách xa nhau về mặt địa lý.

Ngoài ra, nó còn khai thác bộ đệm tốt hơn: Ở cơ chế phân giải tên đệ quy, sau khi phân giải xuống -> gửi ngược trở lại địa chỉ lên thì nút trên cùng sẽ lưu lại địa chỉ IP của nút lá (là của thực thể cần phân giải), lưu lại để lần sau nếu có 1 yêu cầu cùng 1 Domain Name được gửi đến thì nó sẽ trả lời luôn mà không cần gửi xuống dưới nữa, còn phân giải tên không đệ quy thì không gửi ngược trở lại để lưu vào bộ đệm

