**BÀI TẬP TRÊN LỚP MÔN HỌC: HỆ PHÂN TÁN**

**CHƯƠNG 5: Định danh trong hệ phân tán**

HỌ TÊN SV: Nguyễn Đức Thiên MÃ LỚP: 114175

MSSV: 20168806 MÃ HỌC PHẦN:IT4610Q

***Câu hỏi lý thuyết***

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

Vì thông tin có thể chuyển từ Access Point này sang Access Point khác, và một thực thể có thể có nhiều Access Point.

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

Các vấn đề cần xem xét:

* Không gian tên hữu hạn: dẫn tới cạn kiệt định danh cho các thực thể khi số lượng thực thể tăng lên.
* Tính bền vững, chịu được các thay đổi.
* Tính sẵn sàng, chịu lỗi.
* Khả năng rủi ro về bảo mật

*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.*

* Để tránh mất thời gian vào việc thay đổi địa chỉ trong một hệ thống phân cấp, ta có thể dùng con trỏ và cơ chế chuyển tiếp con trỏ kết hợp với mô hình phân cấp.
* Thay vì phải thay đổi địa chỉ của thực thể, ta chỉ cần thay đổi con trỏ ánh xạ đến địa chỉ của thực thể tại A hoặc tại B, dựa trên vị trí của thực thể trong quá trình di chuyển.

*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 Cache Poisoning: Tấn công vào cơ chế sử dụng bộ đệm trong kiến trúc phân cấp. Gửi các gói tin ARP để lưu địa chỉ MAC của hacker vào bộ nhớ đệm của hệ thống. Từ đó có thể kết hợp các phương pháp tấn công khác để đánh cắp thông tin hoặc chiếm quyền điều khiển.
* Tấn công DoS / DDoS: tấn công bằng phương pháp Ping of Death hoặc Ping Flood, khiến hệ thống bị quá tải trong việc xử lý các gói tin ARP.
* Man in the middle: Tấn công vào bộ nhớ đệm, thay thế các bản ghi thật bằng bản ghi giả mạo, từ đó dùng để giả mạo hệ thống.

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

* Chi phí lưu trữ các tham chiếu con trỏ quá nhiều. Khi loại bỏ một thực thể thì phải loại bỏ các tham chiếu tới.
* Không thể tham chiếu tới thực thể khi có một tiến trình trong chuỗi bị hỏng.

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

* Nhược điểm của phương pháp Home-based là luôn sử dụng một vị trí cố định. Điều này khiến hệ thống luôn phải đảm bảo vị trí này tồn tại. Vấn đề xảy ra khi một thực thể di chuyển vĩnh viễn vị trí của nó.
* Để giải quyết vấn đề này thì giải pháp đưa ra là trước khi đăng ký home-based, thực thể sẽ sử dụng một hệ thống định danh truyền thống khác. Nếu không tìm địa chỉ nhà thì sẽ sử dụng hệ thống này để tìm.
* Một giải pháp khác là nếu thực thể di chuyển ra đủ xa khỏi home-based thì sẽ di chuyển home-based của thực thể tới home-based gần nhất

*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?*

* Giúp chuyển tiếp gói tin nhanh hơn dựa vào bảng băm.
* Nếu không có bảng băm thì chuyển tiếp lần lượt tuyến tính lần lượt tới khi tìm thấy node cần tìm.
* Sử dụng bảng băm giúp tìm khoá k nhanh hơn.

*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.*

* Yêu cầu phân giải nếu đã tồn tại trong bộ đệm thì sẽ được xử lí luôn và phản hồi cho client thay vì phải forward lên node ở phía trên. Giúp giảm được thời gian phản hồi và giảm yêu cầu xử lí cho các nút ở phía trên trong mô hình phân cấp
* Ví dụ: máy A gửi yêu cầu X lên máy B. Máy B không xử lí được gửi tiếp lên cho máy C. Sau khi nhận được phản hồi từ C, máy B trả kết quả cho A và lưu vào bộ nhớ đệm. Các lần phân giải sau, B sử dụng kết quả trong bộ nhớ đệm trả về cho A luôn, tiết kiệm được thời gian chuyển tiếp lên C và giảm tải xử lí cho C.

*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 |
| * Truy cập trực tiếp đến vùng nhớ file, có khả năng truy cập kể cả khi file gốc đã bị xoá. * Chỉ áp dụng trên máy local * Tốn ít bộ nhớ * Không liên kết được thư mục * Có cùng inode và permission với file gốc | * Truy cập thông qua con trỏ, không truy cập được file khi file gốc bị xoá * Áp dụng được cả cho những máy khác trong cùng một mạng. * Tốn nhiều bộ nhớ hơn. * Liên kết được tới thư mục * Có inode và permission khác với file gốc |

*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?*

Không, chỉ cần cập nhập nút sau và nút trước.

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

* Phân giải đệ quy tập trung xử lí ở server, tối ưu hơn so với phân giải không đệ quy tập trung xử lí ở client, giảm số lượng yêu cầu và phản hồi gửi giữa client và server.
* Name server lưu trữ kết quả vào bộ nhớ đệm, giúp tăng tốc độc phản hồi.