**BÀI TẬP TRÊN LỚP**

**MÔN HỌC: HỆ PHÂN TÁN**

**CHƯƠNG 5: ĐỊNH DANH TRONG HỆ PHÂN TÁN**

HỌ TÊN SV: Mạc Quang Huy MSSV: 20173169

MÃ LỚP: 118636 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:**   * Nếu thực thể là 1 tiến trình -> Access Point là máy tính. Nếu thực thể là 1 máy tính -> Access Point là mạng * Thực thể có thể dịch chuyển qua lại giữa các AP. Một AP có thể gán với nhiều địa chỉ khác nhau * Cần tách biệt dịch vụ và vị trí * Không thể dùng địa chỉ của Access Point để định danh chính xác các thực thể |

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

|  |
| --- |
| **Trả lời:**   * Với không gian tên phắng: Không gian tên hữu hạn (n bit -> 2^n địa chỉ) * Vấn đề: Cạn kiệt không gian tên * Giải pháp: Mở rộng không gian tên, chấp nhận vi phạm điều 3 (Một định danh mãi mãi chỉ trỏ đến 1 thực thể) |

**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:**   * Khi thông tin dịch chuyển chuyển tới các vị trí trung gian giữa A và B, để lại con trỏ cho đến khi đến B ghi thêm địa chỉ mới đó * Khi đã xong, chuỗi con trỏ trung gian bị xóa đi, địa chỉ ở A cũng 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: giao thức trao đổi thông tin trong mạng cục bộ, qua địa chỉ MAC

|  |
| --- |
| **Trả lời:**  Ví dụ: ARP Spoofing   * Hub có dữ liệu gì cũng chuyển tiếp ngay đến cổng, không như switch là chọn cổng      * Attacker sẽ modifier ARP Cache lại: * Bên Bob: Ipaddr Alice là MacAddr của attacker * Bên Alice: Ipaddr Bob là MacAddr của attacker * Điểm yếu: Không có cơ chế xác thực cho ARP reply |

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

|  |
| --- |
| **Trả lời:**  **Vấn đề:**   * Càng hoạt động nhiều, chuỗi hoạt động càng lớn -> Hệ thống duy trì, quản lý khó * Sử dụng tái định hướng con trỏ * Chuỗi dài vô hạn * Sử dụng các short cut. Có khả năng có chuỗi không tham chiếu được * Lưu trữ vô số các tham chiếu * Loại bỏ các tham chiếu |

**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 giải quyêt các thực thể chuyển động

More infor: http://elib.ictu.edu.vn/Opacdigital/wpViewFileConvertedTrailer.aspx?Id=696

|  |
| --- |
| **Trả lời:**   * Nhược điểm của home-based: * Sử dụng địa chỉ tĩnh cho địa điểm quản lý (phải chắc chắn rằng nơi quản lý các thực thể di động thì luôn luôn tồn tại) * Đăng ký nơi quản lý như là một dịch vụ tên truyền thống và để cho máy khách tìm kiếm địa điểm của nơi quản lý đầu tiên. Bởi vì có thể cho rằng địa điểm của nơi quản lý là tương đối ổn định, địa điểm đó có thể được lưu trữ hiệu quả sau khi nó được tìm kiếm |

**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:**   * Chord cung cấp chức năng của một DHT(distributed-hash-table) hỗ trợ tìm kiếm. Consistent hashing giữ cân bằng tải. Hơn nữa, cân bằng tải làm việc ngay cả trong một dải băm luôn thay đổi, như khi các nút lỗi hoặc rời khỏi hệ thống hay khi nút mới join vào. |

**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:**   * Giúp cho việc tìm kiếm, cập nhật một cách nhanh chóng * Ví dụ: Khi domain con di chuyển giữa 2 subdomains |

**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 cứng** (liên kết vật lý) | **Liên kết mềm** (liên kết biểu tượng) | | Liên kết cứng chỉ dành cho các tập tin; không thể liên kết đến một tệp trên phân vùng khác với số inode khác nhau.  Nếu bản sao *thực sự* bị xóa, **liên kết sẽ hoạt động** , bởi vì nó truy cập dữ liệu cơ bản mà bản sao thực sự đang truy cập. | Có thể tạo liên kết đến tệp và thư mục và bạn có thể tạo liên kết (phím tắt) trên phân vùng khác nhau và với số inode khác với bản gốc.  Nếu bản sao *thực sự* bị xóa, liên kết **sẽ không hoạt động** . | |

**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 các node sau và node trước vì trong hệ thống Chord, mỗi nút sẽ quản lý khóa của node 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: * Phân giải tên có đệ quy xử lý nhiều ở bên server, phân giải không có đệ quy xử lý nhiều bên client. Khi client ở xa, phân giải không đệ quy phải gửi đi gửi lại dẫn tới độ trễ cao * Phân giải tên có đệ quy các name server tổ chứ lưu trữ bộ nhớ đệm, server lưu kết quả, có thể trả lời nhanh cho client |