* 1. **Câu 242.** Có 3 máy tính cùng cung cấp dịch vụ nhân bản. Các nhà sản xuất khẳng định rằng mỗi máy tính có thời gian trung bình giữa các lần xảy ra lỗi là 5 ngày? 1 lỗi thường mất 4 giờ để sửa. Vậy tính khả dụng của dịch vụ nhân bản ở đây là gì?
  2. *Ans.*

The probability that an individual computer is down is 4/(5\*24 + 4) ~ 0.03. Assuming failure-independence of the machines, the availability is therefore 1 – 0.033 = 0.999973.

**Câu 32**: Một client gửi một thông điệp yêu cầu kích thước 200 byte tới một service, service sẽ tạo ra một phản hồi kích thước 5000 byte. Hãy ước tính để hoàn thành một yêu cầu trong mỗi trường hợp sau, với giả định về hiệu suất được liệt kê bên dưới:

a. Sử dụng giao tiếp không kết nối (ví dụ: UDP).

b. Sử dụng giao tiếp hướng kết nối (ví dụ: TCP).

c. Quá trình server ở trong cùng một máy với client.

*Giả định:*

Độ trễ trên mỗi gói (cục bộ hoặc từ xa, phát sinh trên cả gửi và nhận): 5 ms

Thời gian thiết lập kết nối (chỉ TCP): 5 ms

Tốc độ truyền dữ liệu: 10 megabit mỗi giây

MTU: 1000 byte.

Thời gian xử lý yêu cầu máy chủ: 2 ms

Giả sử rằng mạng được tải nhẹ.

***Trả lời:***

A client sends a 200 byte request message to a service, which produces a response containing 5000 bytes. Estimate the total time to complete the request in each of the following cases, with the performance assumptions listed below:  
i) Using connectionless (datagram) communication (for example, UDP);  
ii) Using connection-oriented communication (for example, TCP);  
iii) The server process is in the same machine as the client.  
[Latency per packet (local or remote, incurred on both send and receive):5 milliseconds  
Connection setup time (TCP only):5 milliseconds  
Data transfer rate:10 megabits per second MTU:1000 bytes  
Server request processing time:2 milliseconds

Assume that the network is lightly loaded.]

The send and receive latencies include (operating system) software overheads as well as network delays. Assuming that the former dominate, then the estimates are as below. If network overheads dominate, then the times may be reduced because the multiple response packets can be transmitted and received right after each other.  
i) UDP: 5 + 2000/10000 + 2 + 5(5 + 10000/10000) = 37.2 milliseconds  
ii) TCP: 5 + 5 + 2000/10000 + 2 + 5(5 + 10000/10000) = 42.2 milliseconds  
iii)same machine: the messages can be sent by a single in memory copy; estimate interprocess data transfer rate at 40 megabits/second. Latency/message ~5 milliseconds. Time for server call:  
5 + 2000/40000 + 5 + 50000/40000 = 11.3 millisecond

1. **Câu 168:** Máy chủ NTP B nhận được thông báo của máy chủ A lúc 16: 34: 23.480 mang dấu thời gian 16: 34: 13.430 và trả lời nó. A nhận được tin nhắn lúc 16: 34: 15.725, mang dấu thời gian của B là 16: 34: 25.7. Ước tính phần bù giữa B và A và độ chính xác của ước tính.

Trả lời:

* 1. Q168. An NTP server B receives server A’s message at 16:34:23.480 bearing a timestamp 16:34:13.430 and replies to it. A receives the message at 16:34:15.725, bearing B’s timestamp 16:34:25.7. Estimate the offset between B and A and the accuracy of the estimate.
  2. *Ans.*

Let *a*  *Ti-2 – Ti-3* = 23.48 - 13.43 = 10.05; *b*  *Ti-1* – *Ti* = 25.7- 15.725 =9.975.

Then the estimated offset *oi* = (*a*+*b*)/2 = 10.013s, with estimated accuracy = *di*  2 = ± (*a*-*b*)/2 = 0.038s

(answers expressed to the nearest millisecond).

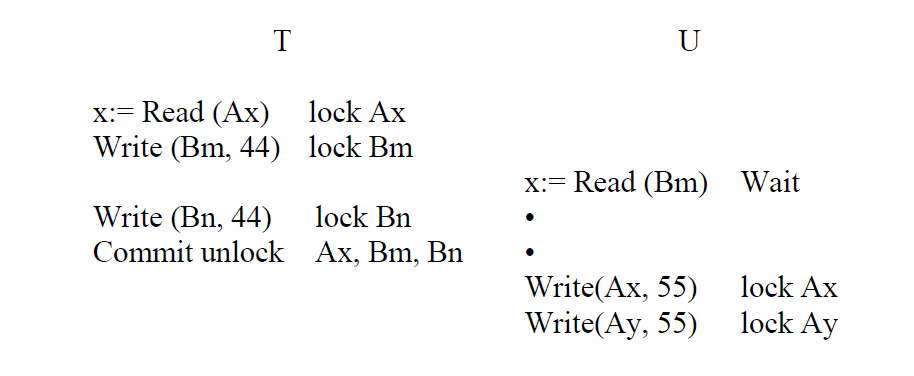
**Câu 258:**

Sao chép có sẵn được áp dụng cho các mục dữ liệu A và B với các bản sao Ax, Ay và Bm, Bn. Các giao dịch T và U được định nghĩa là:

*T*: *Read*(*A*); *Write*(*B,* 44). *U*: *Read*(*B*); *Write*(*A,* 55).

Thể hiện sự xen kẽ của T và U, giả sử rằng khóa hai pha được áp dụng cho các bản sao. Giải thích lý do tại sao một mình các khóa không thể đảm bảo khả năng tuần tự hóa một bản sao nếu một trong các bản sao bị lỗi trong quá trình tiến hành của T và U. Giải thích bằng cách tham khảo ví dụ này, làm thế nào xác thực cục bộ đảm bảo khả năng tuần tự hóa một bản sao.

**Trả lời:**



Giả sử Bm không thành công trước khi T khóa nó, thì U sẽ không bị trì hoãn (nó sẽ nhận được một bản cập nhật bị mất). Vấn đề phát sinh bởi vì Read có thể sử dụng một trong các bản sao trước khi nó bị lỗi và sau đó Write có thể sử dụng bản sao khác. Xác thực cục bộ đảm bảo khả năng tuần tự hóa một bản sao bằng cách kiểm tra trước khi cam kết rằng bất kỳ bản sao nào bị lỗi vẫn chưa được phục hồi. Trong trường hợp của T, quan sát thấy sự thất bại của Bm, đáng ra Bm chưa được phục hồi, nhưng nó đã xảy ra, vì vậy T bị hủy bỏ.

* 1. **Câu 255.** Giải thích lý do tại sao khiến một số người quản lý bản sao ở chế độ chỉ đọc có thể cải thiện hiệu suất của hệ thống tin đồn.
  2. *Trả lời.*
* Đầu tiên, các hoạt động đọc có thể được người quản lý bản sao chỉ đọc cục bộ đáp ứng, trong khi các bản cập nhật chỉ được xử lý bởi một số người quản lý bản sao khác. Đây là một sự sắp xếp hiệu quả nếu trung bình có nhiều thao tác đọc cho mỗi thao tác ghi.
* Thứ hai, vì các bản sao chỉ đọc không chấp nhận cập nhật, chúng không cần mục nhập dấu thời gian vectơ. Do đó, kích thước dấu thời gian vectơ được giảm xuống.
* **Q162.** Tại sao đồng bộ hóa đồng hồ máy tính lại cần thiết? Mô tả các yêu cầu thiết kế cho một hệ thống để đồng bộ hóa các đồng hồ trong một hệ thống phân tán.
* ***Trả lời:***
* Trong các mạng máy tính hiện đại, đồng bộ hóa thời gian là rất quan trọng vì mọi khía cạnh của quản lý, bảo mật, lập kế hoạch và gỡ lỗi mạng đều liên quan đến việc xác định khi nào các sự kiện xảy ra. Thời gian cũng cung cấp hệ quy chiếu duy nhất giữa tất cả các thiết bị trên mạng. Nếu không có thời gian đồng bộ, việc tương quan chính xác các tệp nhật ký giữa các thiết bị này là rất khó, thậm chí là không thể. Sau đây chỉ là một số lý do cụ thể:
* + Việc theo dõi các vi phạm bảo mật, việc sử dụng mạng hoặc các vấn đề ảnh hưởng đến một số lượng lớn các thành phần nếu thời gian không chính xác. Thời gian thường là yếu tố quan trọng cho phép một sự kiện trên một nút mạng được ánh xạ tới một sự kiện tương ứng trên một nút mạng khác.
* + Để giảm sự nhầm lẫn trong các hệ thống tệp được chia sẻ, điều quan trọng là thời gian sửa đổi phải nhất quán, bất kể hệ thống tệp đang sử dụng máy gì.
* + Các dịch vụ thanh toán và các ứng dụng tương tự phải biết chính xác thời gian.
* Mỗi nút có một đồng hồ cục bộ và đồng hồ của tất cả các nút luôn đồng bộ với nhau. Điều này làm cho việc thiết lập thứ tự thời gian thực toàn cầu của các sự kiện không chỉ trên một nút cụ thể mà còn trên các nút trở nên đơn giản.
* Các yêu cầu thiết kế chính:
* i) Cần có giới hạn về độ lệch giữa các đồng hồ hoặc giữa bất kỳ đồng hồ nào và UTC;
* ii) Không chỉnh lùi đồng hồ;
* iii) Chỉ những người được ủy quyền mới có thể đặt lại đồng hồ
* Trong thực tế, không thể đạt được (i) trừ khi chỉ có các hư hỏng nhẹ được giả định là xảy ra và hệ thống là đồng bộ.

**Q271.** Có phải bộ nhớ làm cơ sở cho việc thực thi hai quy trình sau đây là nhất quán tuần tự (giả sử rằng, ban đầu, tất cả các biến đều được đặt bằng 0)?

*P*1: *R*(*x*)1; *R*(*x*)2; *W*(*y*)1

*P*2: *W*(*x*)1; *R*(*y*)1; *W*(*x*)2

Trả lời:

P1 đọc giá trị của x thành 2 trước khi đặt y là 1. Nhưng P2 chỉ đặt x thành 2 sau khi nó đã đọc y thành 1 (trước đó y là 0). Do đó hai lần thực thi này không tương thích và bộ nhớ không nhất quán theo tuần tự.