**BÀI TẬP LÝ THUYẾT CHƯƠNG 2 (PHẦN 2)**

**HỆ PHÂN TÁN**

**Họ và tên: Lê Thị Yên**

**MSSV: 20183861**

**Câu hỏi 1:** **Một kết nối socket cần 4 thông tin nào? Tại sao phải cần đủ 4 thông tin đó?**

* Bản chất của việc trao đổi thông tin giữa các tiến trình là việc trao đổi thông tin giữa các socket của các tiến trình. Vì vậy mỗi tiến trình (cả tiến trình client và tiến trình server) đều phải tạo socket. Mỗi socket cần 2 thông tin: địa chỉ IP của máy, thông tin port gắn với socket đó. Do đó, mỗi kết nối socket cần 4 thông tin: địa chỉ IP của máy Client, port của tiến trình Client, địa chỉ IP máy server và port mà tiến trình Server đang chạy trên máy chủ
* Cần đủ 4 thông tin vì địa chỉ IP để xác định máy gửi và máy nhận trên môi tường mạng internet, cổng port để định danh ứng dụng mà dữ liệu sẽ được gửi tới

**Câu hỏi 2: Tại sao giao thức *yêu cầu-trả lời (request-reply)* lại được coi là *đồng bộ* và *tin cậy*?**

Diagram

Description automatically generated

Giao thức request-reply được coi là đồng bộ và tin cậy vì:

Sau khi gửi request lên cho tiến trình Server thì tiến trình Client tự block mình để chờ reply từ server trả về (thể hiện tính đồng bộ). Tiến trình Server nhận được request, thực thi yêu cầu, trả về kết quả cho Client. Tuy không cần cơ chế báo nhận nhưng khi server gửi thông điệp Reply về cho Client thì gio thức này coi thông điệp Reply cũng đóng vai trò như là thông điệp báo nhận (tính tin cậy)

**Câu hỏi 3:** **Hai vấn đề chính đối với giao thức RPC là gì?**

Hai vấn đề chính đối với giao thức RPC:

- Hệ thống không đồng nhất (trong việc truyền tham số)

+ Không gian nhớ khác nhau

+ Cách biểu diễn thông tin khác nhau

- Có lỗi xảy ra trong quá trình gọi thủ tục từ xa

**Câu hỏi 4:** **Vấn đề đối với truyền *tham biến*  trong RPC là gì? Còn đồi với truyền *tham chiếu*? Giải pháp đưa ra là gì?**

- Vấn đề đối với truyền tham biến:

+ Vấn đề khi biểu diễn dữ liệu khác nhau gây hiểu sai về giá trị của các tham biến đó -> sai về logic hoạt động

+ Các dữ liệu không thuộc cùng một kiểu, các kiểu dữ liệu khác nhau được biểu diễn khác nhau

- Vấn đề đối với truyền tham chiếu: Bộ nhớ phân tán

phía gọi và phía bị gọi nằm ở hai máy tính khác nhau, nên chúng sẽ được thực thi trên những không gian địa chỉ khác nhau, và điều này gây ra sự phức tạp không nhỏ

* Giải pháp: mở rộng mô hình RPC ra làm hai loại chính: Synchronous RPC và Asynchronous RPC.

• Asynchronous RPC (RPC không đồng bộ)

Diagram

Description automatically generated

- Client gửi tới server lời gọi thủ tục và chờ bản tin chấp nhận từ server.

- Phía server sẽ gửi một tín hiệu ACK về cho client thông báo đã nhận được yêu cầu và bắt đầu thực hiện yêu cầu RPC đó.

- Lúc này client sẽ tếp tục thực hiện công việc của mình mà không chờ kết quả từ server như ở RPC truyền thống.

• Synchronous RPC (RPC đồng bộ)

Diagram

Description automatically generated

- Thực hiện hai lời gọi, một từ client và một từ server.

- Client gửi tới server lời gọi thủ tục và chờ bản tin chấp nhận từ server.

- Server gửi bản tin chấp nhận về cho client thông báo đã nhận được yêu cầu và bắt đầu thực hiện yêu cầu RPC đó.

- Lúc này client sẽ tếp tục thực hiện công việc của mình.

- Khi thực hiện thủ tục xong, server sẽ thực hiện lời gọi tới client báo nhận lấy kết quả.

- Client thực hiện ngắt, nhận kết quả và gửi lại cho server tín hiệu ACK đã nhận kết quả thành công.

**Câu hỏi 5:** **So sánh RMI và RPC. Nhược điểm của RMI so với RPC là gì?**

RPC và RMI là các cơ chế cho phép máy khách gọi thủ tục hoặc phương thức từ máy chủ thông qua thiết lập giao tiếp giữa máy khách và máy chủ.

* **Giống nhau:**

**+** Cùng hỗ trợ lập trình với các giao diện

**+** Dựa trên giao thức yêu cầu/trả lời

**+** Mức độ trong suốt

* **Khác nhau:**

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **RPC** | **RMI** |
| Hỗ trợ | Lập trình thủ tục | Lập trình hướng đối tượng |
| Thông số | Cấu trúc dữ liệu thông thường được chuyển đến các thủ tục từ xa | Các đối tượng được truyền cho các phương thức từ xa |
| Hiệu quả | Thấp hơn RMI | Hơn RPC và được hỗ trợ bởi phương pháp lập trình hiện đại (ví dụ: mô hình hướng đối tượng) |
| Tham số | Bắt buộc | không bắt buộc |
| Lập trình | thấp hơn RMI | dễ dàng hơn RPC |
| Chi phí chung | hơn | Ít so sánh |

* **Nhược điểm của RMI so với RPC:**

+ Việc gọi phương thức của đối tượng từ xa luôn phức tạp hơn gọi phương thức cục bộ. Các đối tượng trên hai máy khác nhau hoạt động trên hai tiến trình khác nhau có hai không gian địa chỉ khác nhau nên:

* + Việc tham chiếu đến biến, địa chỉ của đối tượng là khác nhau ở các máy khác nhau
  + Các tham số truyền cho phương thức của đối tượng ở xa phải được đóng gói và chuyển qua mạng đến phương thức thực sự.
  + Lời gọi phương thức từ xa phải thông qua mạng và có thể bị ngắt ngang do mạng gặp sự cố

+ Phụ thuộc vào kết nối mạng

**Câu hỏi 6:** **Giải thích cơ chế trao đổi thông tin hướng thông điệp bất đồng bộ và bền vững.**

* Sử dụng MOM (Message-Oriented Middleware)
* Hệ thống hàng đợi thông điệp hỗ trợ trao đổi thông tin không đồng bộ bền vững
* Nút gửi và nút nhận không cần đồng bộ trong trong quá trình trao đổi thông điệp, nút gửi gửi thông điệp vào hàng đợi lưu trữ trung gian, sau đó nút nhận sẽ lấy yêu cầu từ hàng đợi trung gian đó ra -> phù hợp với dịch vụ chấp nhận độ trễ thời gian cao

Diagram

Description automatically generated with medium confidence

+ Mô hình a: nút sender gửi thông điệp vào hàng đợi, đồng thời nút receiver lấy thông điệp từ hàng đợi ra (cả sender và receiver đều đang chạy)

+ Mô hình b: cho phép nút receiver không chạy, tuy nhiên nút sender vẫn tiếp tục gửi thông điệp vào hàng đợi

+ Mô hình c: nút sender không chạy, nút sender tiếp tục lấy thông điệp từ hàng đợi ra

+ Mô hình d: cả nút sender và receiver đều không chạy, các thông điệp được lưu trữ bền vũng trong hàng đợi

**Câu hỏi 7: Trong trao đổi thông tin hướng dòng, những cơ chế thực thi QoS được thực hiện ở tầng nào? Giải thích. Trình bày một số cơ chế thực thi QoS để chứng minh điều đó.**

Cơ chế thực thi QoS được thực thi ở 2 tầng là tầng liên kết dữ liệu (data-link layer) và tầng mạng (network layer)

- Ở tầng mạng:

• sử dụng 3 bit đầu tiên trong trường Service Type - ToS trong phần đầu của gói dữ liệu IP và 3 bits đầu tiên (P2 đến P0) dùng để quy định các giá trị đánh dấu độ ưu tiên của packet và các giá trị này được gọi là IP Precedence

• Giá trị IP precedence nằm trong khoảng từ 0 đến 7.

* 3 bits đầu tiên (P2 đến P0): IP Precedence. Do sử dụng 3 bits nên sẽ có 8 giá trị (000 đến 111) định ra độ ưu tiên của gói tin từ thấp đến cao. Giúp router xử lý các gói tin này theo chất lượng dịch vụ
* 3 bits tiếp theo (T2 đến T0): bit T2 (T2=1): Yêu cầu truyền gấp. bit T1 (T1=1): Yêu cầu truyền với đường truyền chất lượng cao. bit T0 (T0=1): Yêu cầu truyền đảm bảo.
* 2 bit cuối (CU1-CU2): Không dùng tới (Currently and Unused).

• Trong trường hợp cần thiết phải phân chia nhiều hơn 8 lớp lưu lượng, chúng ta có thể sử dụng 6 bit đầu tiên của trường ToS gọi là trường DSCP

- Ở tầng liên kết dữ liệu:

• Trong phần đầu của khung dữ liệu ở lớp liên kết dữ liệu không có trường nào phục vụ cho việc phân lớp lưu lượng. Tuy nhiên ta có thể phân lưu lượng dựa vào việc chèn thêm các thẻ định danh VLAN gọi là tag. Mỗi tag gồm 4 byte trong đó trường CoS gồm 3 bit được dùng để phân lớp lưu lượng. Như vậy tại mức liên kết dữ liệu chúng ta cũng có thể phân chia lưu lượng thành 8 lớp với các mức ưu tiên tăng dần tương tự như khi sử dụng IP Precedence tại lớp mạng của gói tin IP.

* 1 số cơ chế thực thi của QoS:

- Cơ chế Guaranteed Services: dữ liệu đi qua mạng được dành riêng 1 băng thông chắc chắn cho dữ liệu. Thực hiện thông qua cơ chế RSVP và CBWFQ của QoS.

- Cơ chế Differentiated Services: cơ chế này cho phép phân lớp, phân loại các dịch vụ khác nhau và đặt độ ưu tiên khác nhau cho các loại dịch vụ. Thực hiện thông qua các tool QoS là PQ, CQ, WFQ và WRED.

- Sử dụng bộ đệm để giảm jitter: dòng dữ liệu được chuyển đến ở nút nhận sẽ không được chạy ngay cho Client mà sẽ được lưu trước vào bộ đệm và sau đó mới chạy để đảm bảo khoảng thời gian giữa các đơn vị dữ liệu liên tiếp nhau là giống với bên nút gửi

- Cơ chế Forward error correction (FEC): phân tán sự mất mát gói tin ra toàn bộ dòng dữ liệu