**I. Lý thuyết**

**1. So sánh các mô hình giao tiếp**

**So sánh Remote Procedure Call (RPC), Message Passing, và Stream-Oriented Communication**

| **Đặc điểm** | **RPC** | **Message Passing** | **Stream-Oriented Communication** |
| --- | --- | --- | --- |
| **Cách hoạt động** | Gọi hàm từ xa như gọi hàm cục bộ | Trao đổi dữ liệu dưới dạng thông điệp | Gửi và nhận dữ liệu liên tục dưới dạng luồng |
| **Tính đồng bộ** | Thường là đồng bộ (blocking) | Có thể đồng bộ hoặc bất đồng bộ | Thường là bất đồng bộ |
| **Độ trễ** | Thấp nhưng dễ bị ảnh hưởng bởi lỗi mạng | Cao hơn do cần gửi và nhận thông điệp | Thấp, phù hợp với dữ liệu thời gian thực |
| **Khả năng mở rộng** | Khó mở rộng do chặt chẽ về giao thức | Dễ mở rộng với nhiều điểm giao tiếp | Mở rộng tốt, phù hợp với truyền dữ liệu lớn |
| **Tính trạng thái (Stateful/Stateless)** | Thường là stateful | Thường là stateless | Có thể cả hai |
| **Ví dụ sử dụng** | Gọi API từ xa, microservices | Hệ thống message broker (RabbitMQ, Kafka) | Streaming video, VoIP, IoT |

**Khi nào nên sử dụng từng mô hình?**

* **RPC:** Khi cần giao tiếp nhanh chóng, đồng bộ giữa các dịch vụ như gRPC hoặc các API REST.
* **Message Passing:** Khi cần giao tiếp linh hoạt giữa nhiều dịch vụ như trong các hệ thống phân tán hoặc kiến trúc event-driven.
* **Stream-Oriented Communication:** Khi cần truyền tải dữ liệu liên tục như video streaming, VoIP, hoặc các ứng dụng IoT.

**2. Giải thích cơ chế giao tiếp không đồng bộ**

**So sánh giữa Synchronous và Asynchronous Communication**

| **Đặc điểm** | **Synchronous Communication** | **Asynchronous Communication** |
| --- | --- | --- |
| **Cách hoạt động** | Người gửi phải chờ phản hồi trước khi tiếp tục | Người gửi có thể tiếp tục công việc mà không cần chờ |
| **Hiệu suất** | Thấp hơn do thời gian chờ | Cao hơn vì không bị chặn |
| **Độ phức tạp** | Đơn giản, dễ hiểu | Phức tạp hơn do cần cơ chế callback hoặc queue |
| **Ứng dụng** | Giao dịch ngân hàng, xác thực người dùng | Xử lý background tasks, gửi email, message queue |

**Ví dụ về tình huống sử dụng giao tiếp không đồng bộ**

* **Gửi email xác nhận đơn hàng:** Hệ thống đặt hàng có thể xử lý đơn hàng ngay lập tức mà không cần chờ email gửi xong.
* **Tải ảnh lên server:** Người dùng có thể tiếp tục sử dụng ứng dụng trong khi ảnh vẫn đang tải lên.
* **Hệ thống chat:** Tin nhắn có thể được gửi mà không cần chờ người nhận đọc ngay lập tức.

**3. Giao tiếp dựa trên Publish-Subscribe**

**Mô hình Publish-Subscribe trong hệ phân tán**

* **Người gửi (Publisher)** phát thông điệp mà không cần biết ai sẽ nhận.
* **Người nhận (Subscriber)** đăng ký nhận thông tin từ một kênh chủ đề (topic).
* **Message Broker** (Kafka, Redis Pub/Sub, RabbitMQ) đóng vai trò trung gian chuyển tiếp thông điệp.

**Ưu/Nhược điểm so với Message Queue**

| **Yếu tố** | **Publish-Subscribe** | **Message Queue** |
| --- | --- | --- |
| **Cơ chế hoạt động** | Một tin nhắn có thể gửi cho nhiều subscriber | Mỗi tin nhắn chỉ được xử lý một lần bởi một consumer |
| **Khả năng mở rộng** | Tốt hơn vì nhiều subscriber có thể nhận thông điệp | Giới hạn hơn do mỗi tin chỉ đến một consumer |
| **Tính lưu trữ** | Không lưu trữ lâu, subscriber chỉ nhận được tin nhắn khi đang online | Có thể lưu tin nhắn trong hàng đợi chờ xử lý |
| **Ứng dụng** | Hệ thống thông báo, livestream, event-driven architecture | Xử lý công việc theo hàng đợi như gửi email, xử lý đơn hàng |

**4. Các vấn đề khi truyền tin trên mạng**

**Tại sao packet loss, delay, jitter, out-of-order delivery ảnh hưởng đến hệ thống phân tán?**

| **Vấn đề** | **Ảnh hưởng đến hệ thống phân tán** |
| --- | --- |
| **Packet loss (Mất gói)** | Mất dữ liệu quan trọng, gây lỗi hoặc gián đoạn dịch vụ |
| **Delay (Độ trễ)** | Ảnh hưởng đến hiệu suất, đặc biệt trong giao tiếp real-time |
| **Jitter (Dao động độ trễ)** | Gây ảnh hưởng đến âm thanh, video, hoặc ứng dụng thời gian thực |
| **Out-of-order delivery (Gói tin đến không đúng thứ tự)** | Gây lỗi khi tái tạo dữ liệu hoặc thực hiện giao dịch |

**Cách khắc phục**

* **Packet loss:** Sử dụng TCP để đảm bảo dữ liệu đến đúng đích hoặc thêm cơ chế xác nhận lại gói tin.
* **Delay:** Sử dụng CDN (Content Delivery Network) hoặc tối ưu hóa đường truyền mạng.
* **Jitter:** Sử dụng buffer để ổn định dữ liệu khi phát video hoặc âm thanh.
* **Out-of-order delivery:** Đánh số thứ tự gói tin và dùng thuật toán sắp xếp lại khi nhận.

**II.Thực hành**

Link: *https://*[*github*](https://github.com/Tuyvu/K24.git)*.com/Tuyvu/K24.git*