

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Nguyễn Hải Đan

**ĐÁNH GIÁ CÁC CƠ CHẾ GIAO TIẾP TRONG KIẾN
TRÚC MICROSERVICE**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ thông tin

HÀ NỘI – 2025

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Hải Đan

ĐÁNH GIÁ CÁC CƠ CHẾ GIAO TIẾP TRONG KIẾN
TRÚC MICROSERVICE

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: Your Supervisor's Name

HÀ NỘI – 2025

**VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

Nguyễn Hải Đan

**STUDY OF COMMUNICATION MECHANISMS IN
MICROSERVICE ARCHITECTURE**

Major: Information Technology

Supervisor: Your Supervisor's Name in English

HÀ NỘI – 2025

TÓM TẮT

Tóm tắt: Kiến trúc microservice đã trở thành một xu hướng quan trọng trong phát triển phần mềm hiện đại, cho phép xây dựng các hệ thống phức tạp từ các dịch vụ nhỏ, độc lập. Một trong những thách thức chính trong kiến trúc này là việc quản lý giao tiếp giữa các microservice. Khóa luận này tập trung nghiên cứu các cơ chế giao tiếp trong kiến trúc microservice, bao gồm các mô hình đồng bộ và bất đồng bộ, các giao thức và công nghệ được sử dụng, cũng như các thách thức và giải pháp trong việc triển khai. Nghiên cứu cũng đánh giá hiệu quả của các phương pháp giao tiếp khác nhau thông qua các trường hợp sử dụng thực tế và đề xuất các hướng tiếp cận tối ưu cho các tình huống cụ thể.

Từ khóa: *Microservice, Kiến trúc phần mềm, Giao tiếp dịch vụ, API, Message Queue.*

Mục lục

Chương 1. Mở đầu	1
1.1. Bối cảnh và sự cần thiết của đề tài	1
1.2. Mục tiêu nghiên cứu	1
1.3. Phạm vi nghiên cứu	1
1.4. Phương pháp nghiên cứu	1
1.5. Ý nghĩa khoa học và thực tiễn	2
1.6. Đối tượng và phạm vi nghiên cứu	2
1.6.1. Đối tượng nghiên cứu	2
1.6.2. Phạm vi nghiên cứu	2
1.7. Cấu trúc khóa luận	2
Chương 2. Cơ sở lý thuyết	4
2.1. Tổng quan về Microservice Architecture	4
2.1.1. Khái niệm và đặc điểm	4
2.1.2. Lợi ích và thách thức	4
2.2. Communication trong Microservices	4
2.2.1. Tầm quan trọng của giao tiếp	4
2.2.2. Các yếu tố ảnh hưởng đến giao tiếp	5
Chương 3. Phân tích các Communication Patterns	6
3.1. Cách phân loại các pattern	6
3.1.1. Tiêu chí phân loại theo communication mode	6
3.1.2. Tiêu chí phân loại theo communication scope	6
3.1.3. Các yếu tố ảnh hưởng đến việc lựa chọn pattern	6
3.2. Synchronous Communication Patterns	7
3.2.1. REST API Pattern	7
3.3. Asynchronous Communication (one-to-one)	7
3.3.1. Message Queue Pattern	7
3.4. Asynchronous Communication (one-to-many)	8
3.4.1. Pub/Sub Pattern	8
3.5. So sánh và đánh giá các patterns	8
3.5.1. Performance comparison	8
3.5.2. Error handling capabilities	8
3.5.3. Scalability considerations	8
Chương 4. Triển khai thử nghiệm	9
4.1. Mô tả bài toán và yêu cầu	9
4.1.1. Hệ thống thử nghiệm	9
4.1.2. Yêu cầu hệ thống	9
4.2. Cài đặt và triển khai	9
4.2.1. Thiết kế kiến trúc	9
4.2.2. Lựa chọn công nghệ	9
4.2.3. Chi tiết triển khai	10
4.3. Kết quả triển khai	10
4.3.1. Hiệu suất hệ thống	10
4.3.2. Độ tin cậy	10
4.3.3. Khả năng mở rộng	10

4.3.4. Thiết lập hạ tầng.....	10
4.4. Đánh giá hiệu năng	11
4.4.1. Phương pháp đánh giá	11
4.4.2. Phân tích so sánh.....	11
Chương 5. Đánh giá và thảo luận	12
5.1. Phương pháp và tiêu chí đánh giá	12
5.1.1. Phương pháp đánh giá	12
5.1.2. Tiêu chí đánh giá.....	12
5.2. Kết quả đánh giá.....	12
5.2.1. Đánh giá hiệu suất	12
5.2.2. Đánh giá độ tin cậy	12
5.2.3. Đánh giá khả năng mở rộng	13
5.3. Thảo luận.....	13
5.3.1. Ưu điểm và hạn chế	13
5.3.2. Kiến nghị và hướng phát triển	13

Danh sách hình vẽ

Danh sách bảng

Danh sách từ viết tắt

API: Application Programming Interface – Giao diện lập trình ứng dụng

RPC: Remote Procedure Call – Gọi thủ tục từ xa

MQ: Message Queue – Hàng đợi tin nhắn

REST: Representational State Transfer – Chuyển giao trạng thái biểu diễn

SOA: Service-Oriented Architecture – Kiến trúc hướng dịch vụ

Chương 1.

Mở đầu

1.1. Bối cảnh và sự cần thiết của đề tài

Trong những năm gần đây, kiến trúc microservice đã trở thành một xu hướng quan trọng trong phát triển phần mềm hiện đại. Sự phát triển này đặt ra nhiều thách thức trong việc quản lý giao tiếp giữa các dịch vụ. Việc nghiên cứu và hiểu rõ các cơ chế giao tiếp trong kiến trúc microservice là cần thiết để:

- Tối ưu hóa hiệu suất của hệ thống
- Đảm bảo độ tin cậy và khả năng mở rộng
- Giảm thiểu độ phức tạp trong phát triển và bảo trì
- Nâng cao khả năng chịu lỗi của hệ thống

1.2. Mục tiêu nghiên cứu

Khóa luận này nhằm đạt được các mục tiêu sau:

- Phân tích và đánh giá các cơ chế giao tiếp trong kiến trúc microservice
- So sánh ưu nhược điểm của các phương pháp giao tiếp khác nhau
- Đề xuất các giải pháp tối ưu cho các tình huống cụ thể
- Thực nghiệm và đánh giá hiệu quả của các cơ chế giao tiếp

1.3. Phạm vi nghiên cứu

- Tập trung vào các cơ chế giao tiếp phổ biến trong microservice
- Đánh giá trên các tiêu chí: hiệu suất, độ tin cậy, khả năng mở rộng
- Thực nghiệm trên các nền tảng và công nghệ phổ biến

1.4. Phương pháp nghiên cứu

- Phương pháp nghiên cứu lý thuyết:
 - + Tổng hợp và phân tích tài liệu
 - + So sánh các phương pháp tiếp cận

- + Đánh giá ưu nhược điểm
- **Phương pháp nghiên cứu thực nghiệm:**
 - + Xây dựng môi trường thử nghiệm
 - + Triển khai các cơ chế giao tiếp
 - + Đo lường và đánh giá kết quả

1.5. Ý nghĩa khoa học và thực tiễn

- **Ý nghĩa khoa học:**
 - + Đóng góp vào việc nghiên cứu và phát triển các phương pháp giao tiếp trong kiến trúc microservice
 - + Cung cấp cơ sở lý thuyết cho việc lựa chọn và triển khai các cơ chế giao tiếp
 - + Đề xuất các hướng nghiên cứu mới trong lĩnh vực này
- **Ý nghĩa thực tiễn:**
 - + Cung cấp hướng dẫn thực tế cho việc triển khai các hệ thống microservice
 - + Giúp các nhà phát triển đưa ra quyết định phù hợp về cơ chế giao tiếp
 - + Tối ưu hóa hiệu suất và độ tin cậy của hệ thống

1.6. Đối tượng và phạm vi nghiên cứu

1.6.1. Đối tượng nghiên cứu

- Các cơ chế giao tiếp trong kiến trúc microservice
- Các công nghệ và giao thức giao tiếp phổ biến
- Các mô hình triển khai và quản lý giao tiếp

1.6.2. Phạm vi nghiên cứu

- Tập trung vào các cơ chế giao tiếp phổ biến trong microservice
- Đánh giá trên các tiêu chí: hiệu suất, độ tin cậy, khả năng mở rộng
- Thực nghiệm trên các nền tảng và công nghệ phổ biến

1.7. Cấu trúc khóa luận

Khóa luận được tổ chức thành 5 chương:

- **Chương 1: Mở đầu** - Giới thiệu tổng quan về đề tài

- **Chương 2: Cơ sở lý thuyết** - Trình bày các khái niệm cơ bản
- **Chương 3: Phân tích các cơ chế giao tiếp** - Chi tiết về các phương pháp giao tiếp
- **Chương 4: Đánh giá và thực nghiệm** - Kết quả thực nghiệm và phân tích
- **Chương 5: Kết luận và hướng phát triển** - Tổng kết và đề xuất

Chương 2.

Cơ sở lý thuyết

2.1. Tổng quan về Microservice Architecture

2.1.1. Khái niệm và đặc điểm

Microservice Architecture là một kiến trúc phần mềm trong đó các ứng dụng được phát triển như một tập hợp các dịch vụ nhỏ, độc lập, mỗi dịch vụ chạy trong một quy trình riêng và giao tiếp với nhau thông qua các cơ chế nhẹ, thường là HTTP resource API.

2.1.2. Lợi ích và thách thức

- Lợi ích:
 - + Khả năng mở rộng độc lập
 - + Dễ dàng triển khai và bảo trì
 - + Sử dụng công nghệ đa dạng
 - + Khả năng chịu lỗi cao
- Thách thức:
 - + Quản lý giao tiếp giữa các dịch vụ
 - + Đảm bảo tính nhất quán dữ liệu
 - + Giám sát và debug phức tạp
 - + Quản lý phiên bản

2.2. Communication trong Microservices

2.2.1. Tầm quan trọng của giao tiếp

Giao tiếp giữa các microservice là yếu tố quan trọng quyết định hiệu suất và độ tin cậy của toàn bộ hệ thống. Việc lựa chọn cơ chế giao tiếp phù hợp ảnh hưởng trực tiếp đến:

- Hiệu suất của hệ thống
- Khả năng mở rộng
- Độ tin cậy
- Tính nhất quán dữ liệu

2.2.2. Các yếu tố ảnh hưởng đến giao tiếp

- Yêu cầu về độ trễ
- Tính nhất quán dữ liệu
- Khối lượng giao tiếp
- Mô hình giao tiếp (đồng bộ/bất đồng bộ)
- Phạm vi giao tiếp (one-to-one/one-to-many)

Chương 3.

Phân tích các Communication Patterns

3.1. Cách phân loại các pattern

3.1.1. Tiêu chí phân loại theo communication mode

- Synchronous Communication
 - + REST API
 - + gRPC
 - + GraphQL
- Asynchronous Communication
 - + Message Queue
 - + Event Bus
 - + Pub/Sub

3.1.2. Tiêu chí phân loại theo communication scope

- One-to-One Communication
 - + Direct API calls
 - + Point-to-point messaging
- One-to-Many Communication
 - + Event broadcasting
 - + Pub/Sub messaging

3.1.3. Các yếu tố ảnh hưởng đến việc lựa chọn pattern

- Performance requirements
- Data consistency needs
- System scalability
- Error handling requirements
- Development complexity

3.2. Synchronous Communication Patterns

3.2.1. REST API Pattern

- Request-Response model
- HTTP methods (GET, POST, PUT, DELETE)
- Stateless communication
- Order-Inventory check
- Payment processing
- Simple CRUD operations
- Ưu điểm:
 - + Simple implementation
 - + Immediate feedback
 - + Standard protocol
- Nhược điểm:
 - + High latency
 - + Resource blocking
 - + Tight coupling

3.3. Asynchronous Communication (one-to-one)

3.3.1. Message Queue Pattern

- Producer-Consumer model
- Message persistence
- Guaranteed delivery
- Long-running payment processing
- Background tasks
- Batch processing
- Ưu điểm:
 - + Better resource utilization
 - + Loose coupling
 - + Reliable delivery
- Nhược điểm:
 - + Eventual consistency
 - + Complex workflow
 - + Message ordering

3.4. Asynchronous Communication (one-to-many)

3.4.1. Pub/Sub Pattern

- Publisher-Subscriber model
- Topic-based routing
- Event-driven architecture
- Order notifications
- User activity logging
- Real-time updates
- Ưu điểm:
 - + High scalability
 - + Decoupled services
 - + Efficient broadcasting
- Nhược điểm:
 - + Message ordering
 - + Delivery guarantees
 - + Complex setup

3.5. So sánh và đánh giá các patterns

3.5.1. Performance comparison

- Latency metrics
- Throughput capabilities
- Resource utilization

3.5.2. Error handling capabilities

- Retry mechanisms
- Error propagation
- Recovery strategies

3.5.3. Scalability considerations

- Horizontal scaling
- Load balancing
- Service discovery

Chương 4.

Triển khai thử nghiệm

4.1. Mô tả bài toán và yêu cầu

4.1.1. Hệ thống thử nghiệm

- E-commerce order processing system
- Microservices architecture
- Multiple communication patterns

4.1.2. Yêu cầu hệ thống

- Order-Inventory management
- Payment processing
- Order notifications
- User activity logging

4.2. Cài đặt và triển khai

4.2.1. Thiết kế kiến trúc

- Service boundaries
- Communication patterns
- Data flow
- Error handling

4.2.2. Lựa chọn công nghệ

- Spring Boot for services
- RabbitMQ for message queue
- Kafka for pub/sub
- Docker for containerization

4.2.3. Chi tiết triển khai

- REST API implementation
- Message Queue implementation
- Pub/Sub implementation
- Activity tracking system

4.3. Kết quả triển khai

4.3.1. Hiệu suất hệ thống

- Latency metrics
- Throughput results
- Resource utilization
- Error rates

4.3.2. Độ tin cậy

- Processing times
- Success rates
- Error handling
- Recovery times

4.3.3. Khả năng mở rộng

- Broadcast performance
- Service failure impact
- System stability
- Resource efficiency

4.3.4. Thiết lập hạ tầng

- Docker containers
- Service discovery
- Message brokers
- Monitoring tools

4.4. Đánh giá hiệu năng

4.4.1. Phương pháp đánh giá

- Test scenarios
- Performance metrics
- Testing tools
- Data collection

4.4.2. Phân tích so sánh

- Synchronous vs Asynchronous
- One-to-One vs One-to-Many
- Resource utilization
- Error handling

Chương 5.

Đánh giá và thảo luận

5.1. Phương pháp và tiêu chí đánh giá

5.1.1. Phương pháp đánh giá

- Phân tích định tính
- Phân tích định lượng
- So sánh với các nghiên cứu liên quan

5.1.2. Tiêu chí đánh giá

- Hiệu suất hệ thống
- Độ tin cậy
- Khả năng mở rộng
- Độ phức tạp triển khai
- Chi phí vận hành

5.2. Kết quả đánh giá

5.2.1. Đánh giá hiệu suất

- So sánh các phương thức giao tiếp
- Phân tích độ trễ
- Đánh giá thông lượng
- Hiệu quả sử dụng tài nguyên

5.2.2. Đánh giá độ tin cậy

- Khả năng chịu lỗi
- Cơ chế phục hồi
- Đảm bảo tính nhất quán
- Xử lý sự cố

5.2.3. Đánh giá khả năng mở rộng

- Khả năng mở rộng ngang
- Cân bằng tải
- Phát hiện dịch vụ
- Quản lý phiên bản

5.3. Thảo luận

5.3.1. Ưu điểm và hạn chế

- Điểm mạnh của các phương pháp
- Những hạn chế cần khắc phục
- Các vấn đề cần nghiên cứu thêm

5.3.2. Kiến nghị và hướng phát triển

- Cải tiến phương pháp
- Mở rộng phạm vi nghiên cứu
- Ứng dụng thực tế
- Hướng nghiên cứu tương lai

Tài liệu tham khảo

Tài liệu tham khảo

[1] Tài liệu tham khảo 1

[2] Tài liệu tham khảo 2