

Họ và tên: Đào Vũ Bảo Quỳnh

Lớp: 12_ĐH_CNTT5

Mô phỏng truy vấn phân tán

Hệ quản trị NoSQL: Apache Cassandra

1. Bài toán mô phỏng: Hệ thống theo dõi lịch sử mượn - trả sách điện tử (E-book) của một nền tảng thư viện số

- Đặc điểm dữ liệu:

- + Một người dùng có thể mượn nhiều sách
- + Một cuốn sách được & mượn bởi nhiều người
- + Truy vấn thường xuyên theo: Người dùng và khoảng thời gian
- + Dữ liệu phát sinh liên tục, yêu cầu mở rộng cao

2. Thiết kế bảng dữ liệu:

```
CREATE TABLE borrow_history_by_user (  
    user_id text,  
    ebook_id text,  
    borrow_time timestamp,  
    return_time timestamp,  
    device_type text,  
    PRIMARY KEY (user_id), borrow_time, ebook_id)  
);
```

user_id: Partition Key

borrow_time, ebook_id: Clustering Key

Mỗi partition lưu toàn bộ lịch sử mượn sách của một người dùng

Dữ liệu được sắp xếp theo thời gian mượn

3. Cấu hình cụm Cassandra mô phỏng

V0: Hệ thống triển khai 5 node

Node N1: Token 0 - 20%

Node N2: Token 20 - 40%

Node N3: Token 40 - 60%

Node N4: Token 60 - 80%

Node N5: Token 80 - 100%

Replication Factor = 3

Mỗi bản ghi tồn tại trên 3 node khác nhau

4. Mô phỏng lưu trữ dữ liệu phân tán

- Dữ liệu phải sinh: Người dùng U001 mượn E-book EB102
lúc 2025-10-01 20:30

- Quá trình:

+ Cassandra hash user_id = U001

+ Giá trị hash thuộc vùng token của Node N4

+ Dữ liệu được ghi đồng thời lên: Node N1, Node N4 (Primary replica), Node N5

U001 → N4 → N5 → N1

Nếu 1 node bất kỳ trong 3 node trên gặp sự cố, dữ liệu vẫn được đảm bảo an toàn.

5. Mô phỏng truy vấn phân tán

- Truy vấn: Lịch sử mượn sách của người dùng U001 trong tháng 01/2026

```
SELECT ebook_id, borrow_time, return_time
```

```
FROM borrow_history_by_user
```

```
WHERE user_id = 'U001'
```

```
AND borrow_time >= '2026-01-01'
```

```
AND borrow_time <= '2026-01-31';
```

- Cách Cassandra xử lý truy vấn:

+ Client gửi truy vấn đến Node N2 (Node bất kỳ)

+ Node N2 trở thành Coordinator

+ Node N2 xác định Partition U001 nằm trên N4, N5, N1

+ Node N2 gửi truy vấn song song tới 3 node replica

+ So sánh timestamp (Read Repair nếu cần)

+ Tổng hợp kết quả → trả về client

Client

Node N2 (Coordinator)

N4 N5 N1

↓
Dữ liệu hợp nhất

6. Mô phỏng tình huống lỗi

- TH: Node N4 bị ngắt kết nối
- Hệ thống phản ứng:
 - + Coordinator (N2) vẫn nhận dữ liệu từ Node N5, Node N1
 - + Không gián đoạn truy vấn
 - + Khi N4 hoạt động lại \rightarrow Read Repair tự động đồng bộ dữ liệu

7. Đánh giá mô phỏng

- Ưu điểm:
 - + Truy vấn nhanh theo người dùng
 - + Phân tán dữ liệu đều
 - + Không cần JOIN
 - + Phù hợp dữ liệu tăng liên tục
- Nhược điểm:
 - + Không truy vấn tốt theo ebook_id
 - + Phải thiết kế bảng theo truy vấn trước

8. Kết luận

- Mô phỏng trên cho thấy Apache Cassandra xử lý truy vấn phân tán hiệu quả trong hệ thống thư viện số. Nhờ kiến trúc peer-to-peer, replication và coordinator node, hệ thống vẫn hoạt động ổn định ngay cả khi một node gặp sự cố.