

Tên: Nguyễn Quốc Tú

MSSV: 1250080219

Amazon DynamoDB - Hệ quản trị CSDL NoSQL

1. Nền tảng CSDL: Amazon DynamoDB

2. Lịch sử hình thành:

- DynamoDB được Amazon giới thiệu năm 2012, dựa trên ý tưởng từ Dynamo Paper (2007) - nghiên cứu nội bộ của Amazon về hệ thống lưu trữ phân tán có khả năng mở rộng cao lớn
- Mục tiêu, cung cấp CSDL NoSQL fully managed, độ trễ thấp (millisecond), mở rộng tự động mà không cần quản trị máy chủ

3. Bộ phận quản lý:

- Phát triển và vận hành bởi Amazon Web Service
- Người dùng không cần cài đặt hay quản lý hạ tầng

4. Mô hình lưu trữ

- key-value và Document Store
- Dữ liệu được lưu trong Table
- Nội item gồm:
  - + Partition key (bất biến)
  - + Sort key (tùy chọn)
  - + Các attribute (JSON-like)

HONGHA

Ví dụ item trong bảng - DynamoDB

```
{ "UserId": "Tu",  
  "PostTime": "2025-01-10T09:00",  
  "Content": "DynamoDB rất nhanh",  
  "Likes": 15,  
  "Tags": ["cloud", "aws"]  
}
```

## 5. Ngôn ngữ thao tác với dữ liệu

- AWS SDK / API (Java, Python, Node.js, ...)
- Câu lệnh phổ biến:
  - + PutItem
  - + GetItem
  - + Query
  - + Scan

## 6. Có chế độ phân tán dữ liệu

### 6.1 Phân vùng (Partitioning)

- DynamoDB dùng partition key để:
  - + Băm (hash)
  - + Phân phối dữ liệu tự động lên nhiều physical partitions
- Người dùng không cần cấu hình shard

### 6.2 Sao chép và đồng bộ

- Mỗi item được replicate tự động trên nhiều

Availability Zones (AZs)

- Không có single point of failure

### 6.3 Nhận quan

- Eventually consistent reads (mặc định)

- Strongly consistent reads (tùy chọn, trong cùng region)

### 7. Mô phỏng lũy thừa và truy vấn phân tán

#### 7.1 Bàn tay mô phỏng

- Ứng dụng mạng xã hội cần bao gồm các chức năng

- Thiết kế bằng

- Table: Posts

Thuộc tính | vai trò

UserId | Partition key

PostTime | Sort key

Content | Nội dung

Likes | Số like thích

Page | Nhóm

### 7.2 Mô phỏng lũy thừa phân tán

bhi ghi dữ liệu

1. Client gửi PutItem lên DynamoDB

2. DynamoDB:

- Hash UserId
  - Xác định partition đó
  - Lấy item vào trong partition đó
3. Item được replicate sang nhiều AZ phải như sau

Ví dụ:

User Id = U1 → Hash → Partition A

User Id = U2 → Hash → Partition C

- Ngoài dung (không thay shard, tất cả đến ra tại cùng)

### 7.3 Nguồn phỏng truy vấn phân tán (lucky)

- Truy vấn: lấy tất cả bài viết của UserId = U1  
 response = dynamodb.query(  
 TableName = 'Posts',  
 KeyConditionExpression = 'UserId = :uid',  
 ExpressionAttributeValues = {  
 ':uid': { 'S': 'U1' } }

) {

- Cách DynamoDB xử lý:

T Do truy vấn có Partition key, DynamoDB:
 

- Chỉ truy cập 1 partition duy nhất
- Không fan-out toàn bộ thông

Note:

Date: / /

+ → Rẽ nhau ( $O(1)$ )

7.4 Mô phỏng truy vấn phân tán toàn bộ (Scan)

- truy vấn: Tìm cái bài viết có tag = "cloud"

response = dynamodb.scan (

Table Name = 'Posts'

FilterExpression = 'contains (Tags, :t)',

ExpressionAttributeValues = {

:t: \$'s': 'cloud' }

}

)

- DynamoDB xử lý

1. Gửi request đến tất cả các dữ liệu

1. Gửi request đến tất cả partition

2. Mỗi partition lọc dữ liệu sau đó

3. Gộp kết quả trả về client

- Scan = truy vấn phân tán toàn hệ thống

→ chậm và tốn tài nguyên

XXXX

Sơ đồ kiến trúc Phân tán

