Hệ thống thực hiện A/B Test trong Quảng Cáo

Đồ án tốt nghiệp

Lê Tiến Chiến

Ngành: Khoa học máy tính

Ngày 12 tháng 8 năm 2022

Nội dung

Vấn đề

- Trong nghiệp vụ, khi ta muốn thử nghiệm nhiều giải pháp mới cho một bài toán
- Cách truyền thống đó là triển khai các giải pháp liên tiếp nhau và so sánh kết quả của từng giai đoạn
- Những giới hạn:
  - Thời gian không đủ tin cậy
  - Số liệu không trực quan
  - Không thể so sánh quá nhiều giải pháp

Mục tiêu

Chúng ta cần xây dựng hệ thống có thể giải quyết các vấn đề trên

- Giải quyết được như cầu phức tạp của nghiệp vụ
- Thử nghiệm nhiều giải pháp đồng thời
- Hệ thống trực quan, đáng tin cậy
- Quản lý thời gian rõ ràng
- Thiết lập thử nghiệm dễ dàng, hiệu quả

Từ đó, em xin giới thiệu hệ thống A/B Test được xây dựng dựa trên

- Sử dụng Golang cho Backend
- Lưu trữ dữ liệu ở Redis
- Xây dựng Website với ReactJS

A/B Test là gì

A/B Testing (hay còn được gọi là split testing hay bucket testing) là một phương pháp để so sánh giữa các biến thể của một thuật toán hoặc ứng dụng nào đó, từ đó tìm ra được biến thể nào có hiệu quả tốt hơn.

Ứng dụng của A/B Testing

- Xây dựng Website
- Email Marketing
- Quảng Cáo và Bán Hàng
- Thiết kế Ứng dụng di động

Cách hoạt động của A/B Test

A/B Testing về cơ bản là một cuộc thử nghiệm mà trong đó, hai hoặc nhiều biến thể của trang được hiển thị cho người dùng một cách ngẫu nhiên. Và kết quả của hành động đó sẽ được thể hiên qua conversion rate (tỉ lệ chuyển đổi), từ đó phân tích thống kê được sử dụng để xác định biến thể nào hoạt động tốt hơn cho mục tiêu chuyển đổi nhất định.

Cấu trúc A/B Test

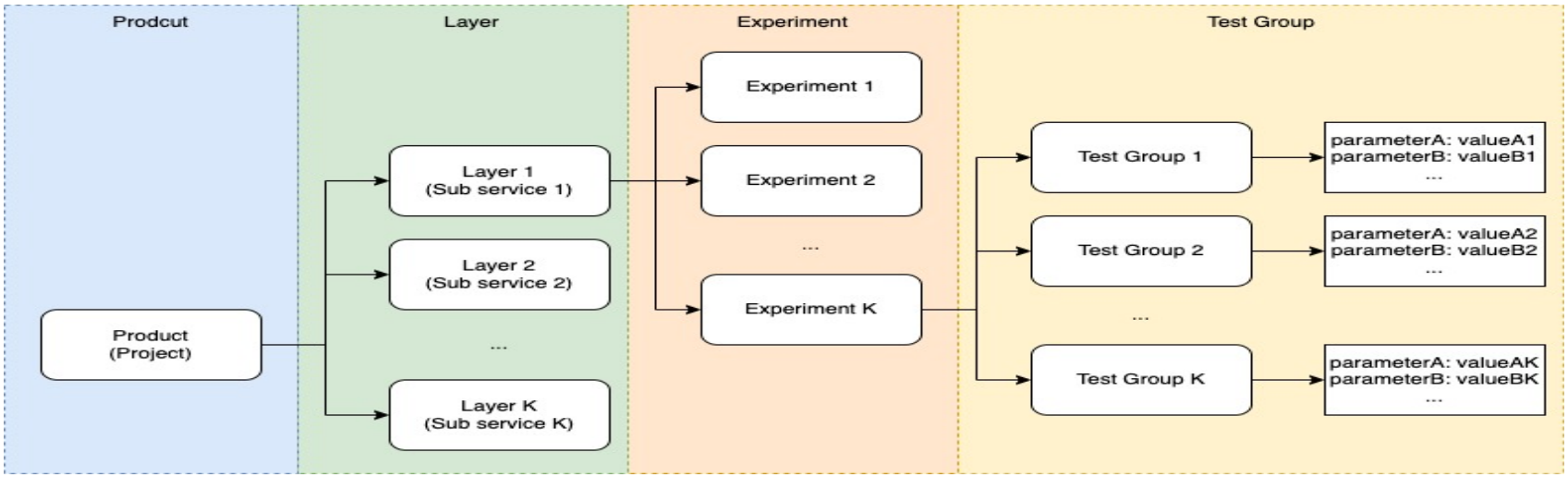
Product Phân tách nghiệp vụ lớn

Layer Phân tách nghiệp vụ nhỏ

Experiment Đại diện cho một thí nghiệm

Test Group Đại diện cho một biến thể của một thí nghiệm

Parameter Đại diện cho các đặc tính của một biến thể



Product

- Product dùng để phân tách giữa nghiệp vụ lớn
- Chỉ có thể sử dụng một Product cho một đối tượng

Ví dụ

- MobileApp (Product)
  - Giao diện (Layer)
  - Cài đặt (Layer)
- Advertisement (Product)
  - Recall (Layer)
  - Rerank (Layer)

Layer

- Layer dùng để phân tách các nghiệp vụ nhỏ trong một Product
- Có thể sử dụng nhiều Layer đồng thời cho một đối tượng
- Sở hữu 100% lượng truy cập
- Có 2 loại hash strategy để quyết định Experiment
  - Dùng UserId
  - Dùng SessionId

Experiment

- Experiment là đại diện cho một thí nghiệm, thuật toán hay giải pháp
- Có thể tạo nhiều Experiment trong một Layer
- Experiment quản lý trạng thái, thời gian và dung lượng truy cập
- Chỉ có thể sử dụng dung lượng truy cập còn lại của Layer

Test Group

- Test Group đại diện cho một biến thể trong một Experiment
- Có thể tạo nhiều Test Group trong một Experiment

Ví dụ

- New Theme (Experiment)
  - Red (Test Group)
  - White (Test Group)

Parameter

Định nghĩa

- Parameter đại diện cho các đặc tính của một Test Group
- Có thể tạo nhiều Parameter trong một Test Group
- Là một cặp key-value mà đối tượng có thể đọc và sử dụng
- Chỉ có thể sử dụng những key từ danh sách do đối tượng chuẩn bị

Ví dụ

color=back, color=red, enabled=true, enabled=false, etc..

Domain Driven Design là gì

Domain-Driven Design là một phương pháp trong việc phân tích và phát triển phần mềm trong khi giải quyết nghiệp vụ phức tạp.

Ý tưởng của cách thức này là xây dựng sự kết nối chặt chẽ giữa thiết kế phần mềm và mô hình nghiệp vụ.

Entity	Các đối tượng được định danh
Value Object	Mô tả các khía cạnh của domain, bất biến, không có định danh
Aggregate	Nhóm các Entity và Value Object, đảm bảo toàn vẹn và ràng buộc
Service	Interface của hành động, chỉ quan tâm đến đối tượng xử lý bởi Service
Repository	Nơi lưu trữ đối tượng, dùng để truy xuất toàn cục

Cấu trúc hệ thống Backend

Service Là thành phần đảm nghiệm hệ thống HTTP, xử lý các truy cập từ Frontend

Storage Là thành phần tổng hợp dữ liệu, thực hiện A/B Test cho tầng Service

Database Là thành phần quản lý dữ liệu cho tầng Storage

Types Tổng hợp các thực thể ở tầng Backend

Cấu trúc A/B Test

Thực thể	Khuôn mẫu	Giải thích
Product	Entity	Product có id định danh riêng
	Aggregate	Product là tổng hợp của các Layers
Layer	Entity	Layer có id định danh riêng
	Aggregate	Layer là tổng hợp của các Experiment
Experiment	Entity	Experiment có id định danh riêng
	Aggregate	Experiment có nhiều Test Group
Test Group	Entity	Test Group có id định danh riêng
	Aggregate	Test Group gồm nhiều Parameter
Parameter	Value object	Parameter không có định danh

Hệ thống Backend

Thực thể	Khuôn mẫu	Giải thích
Service	Services	Service chịu trách nghiệm điều phối hoạt động giữa các tầng thông qua phương thức HTTP
Storage		Storage chịu trách nghiệm tổng hợp các Entity từ tầng Database
Database	Repository	Database là tầng trung gian giữa các tầng khác và tầng cơ sở dữ liệu (Redis)
Types	Aggregate	Types là nơi tổng hợp các Entities và các Value Object

Sau khi khởi tạo A/B Test

Product **Layer** **Experiment**

MobileApp / Giao diện / New Theme

Update

Traffic	Status	Test Group
50%	Started	3

Start Time	End Time
03/08/2022	31/08/2022

Test Group list

ID	Test Group Name	Parameter Name	Parameter Value
1	Trắng	color	white
2	Đen	color	black
3	Đỏ	color	red

Create new Test Group

Kết quả thí nghiệm

Test an experiment

Product *

MobileApp

UserId *

1

SessionId *

0

Raw Result

Product	Layer	Experiment	Test Group	Parameter Name	Parameter Value
MobileApp	Giao diện	New Theme	Đen	color	black


```
[
  {
    "product_id": 1,
    "layer_id": 1,
    "experiment_id": 1,
    "group_id": 2,
    "parameters": [
      {
        "name": "color",
        "value": "black"
      }
    ]
  }
]
```

Kết luận

Những gì đã đạt được:

- Ứng dụng Domain Driven Design vào thiết kế hệ thống
- Hệ thống thiết lập thử nghiệm A/B Test
- Hệ thống thực hiện A/B Test

Định hướng phát triển sắp tới:

- Hệ thống thu thập hiệu suất của A/B Test
- Hệ thống hiển thị kết quả của A/B Test