|  |  |  |
| --- | --- | --- |
| ĐẠI HỌC BÁCH KHOA HÀ NỘI  TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG  ---🙢🙠---  A red and white logo  Description automatically generated  BÁO CÁO BÀI TẬP LỚN  Đề tài: Phân tích dữ liệu Liên minh huyền thoại     |  |  | | --- | --- | |  |  |   Giảng viên hướng dẫn: TS.  Nhóm 7: Nguyễn Trí Lâm 20200337  Nguyễn Đức Mạnh 20204830  Trần Anh Quốc 20200296    Lớp: CTTN - Khoa học máy tính  Học phần: Lưu trữ và xử lí dữ liệu lớn  Mã học phần: IT4490  Mã lớp: 143801  Khóa: K65  Hà Nội, 01/2024 |

Contents

[**1.** **Craw dữ liệu** 4](#_Toc155080739)

[2. Những dữ liệu cần thiết 4](#_Toc155080740)

[3. Craw riot\_id từ trang web OP.GG 4](#_Toc155080741)

[4. Craw puuid, match\_id, match\_detail 4](#_Toc155080742)

[5. Sử dụng Kafka trong quá trình craw dữ liệu 4](#_Toc155080743)

[6. Đẩy dữ liệu lên MongoDB 6](#_Toc155080744)

[**2.** **Xử lí dữ liệu với PySpark** 6](#_Toc155080745)

[**3.** **Visualize dữ liệu với Flask** 6](#_Toc155080746)

**Giới thiệu**

Liên minh huyền thoại, thường được gọi tắt là "LoL," là một trong những tựa game trực tuyến nhiều người chơi phổ biến nhất trên thế giới, được phát triển và phát hành bởi Riot Games. Từ khi ra mắt vào năm 2009, Liên minh huyền thoại đã thu hút hàng triệu người chơi trên toàn cầu và trở thành một biểu tượng trong thế giới game. Được xây dựng trên nền tảng của thể loại trò chơi MOBA (Multiplayer Online Battle Arena), LoL kết hợp những yếu tố chiến đấu đội hình và chiến thuật tinh vi, tạo nên một thế giới đầy sức hấp dẫn và cạnh tranh.

Báo cáo này sẽ tiến hành triển khai phân tích dữ liệu về trò chơi, bắt đầu từ việc craw dữ liệu cho đến phân tích và lưu trữ,sử dụng những công cụ cho dữ liệu lớn như Kafka, Spark, MongoDB.

Tổng quan mô hình của nhóm:

A diagram of a computer program

Description automatically generated

# **Craw dữ liệu**

* 1. Những dữ liệu cần thiết

Dữ liệu về trận đấu LMHT có thể gọi thông qua API trên trang web: [Riot Developer Portal (riotgames.com)](https://developer.riotgames.com/). Tuy vậy để lấy được dữ liệu chi tiết 1 trận đấu, chúng ta cần có thông tin về id của trận đấu đó. Trận đấu trong LMHT được gắn liền với mỗi người chơi và như vậy chúng ta sẽ cần id của mỗi người chơi, bao gồm puuid , riot\_id. Tổng quan lại, muốn lấy được thông tin chi tiết 1 trận đấu, chúng ta cần lấy theo trình tự sau:  
  
riot\_id (bao gồm riot\_name và riot\_tagLine) -> puuid -> match\_id -> match\_detail

* 1. Craw riot\_id từ trang web OP.GG

OP.GG là trang web phân tích chuyên về LMHT. Trang web cũng có xếp hạng người chơi. Vào trang web, trên thanh công cụ chọn Leaderboard:A screenshot of a computer

Description automatically generated

Danh sách người chơi sẽ hiện ra như sau:

A screenshot of a computer

Description automatically generated

Nhóm lấy dữ liệu trong 2000 trang. Để lấy được dữ liệu, nhóm sử dụng thư viện BeatifulShop4 – một thư viện chuyên dùng cho web scraping.

* 1. Craw puuid, match\_id, match\_detail

Để lấy được puuid, match\_id, match\_detail, nhóm sử dụng API được riot game cung cấp tại trang [Riot Developer Portal (riotgames.com)](https://developer.riotgames.com/).

* 1. Sử dụng Kafka trong quá trình craw dữ liệu

Để đảm báo tính chịu lỗi cho quá trình craw dữ liệu, nhóm sử dụng Kafka. Chi tiết như sau:  
1. Tạo topic tên ‘page’. Topic này sẽ bao gồm các số từ 1-2000, tượng trưng cho các trang cần craw từ OP.GG.

2. Tạo 1 tiến trình tên craw\_riot\_id. Tiến trình này sẽ bao gồm 1 consumer của topic ‘page’, craw dữ liệu của trang ứng với message nhận được. Tiến trình này cũng sẽ bao gồm 1 producer, giúp đẩy riot\_id craw được vào topic “riot\_id\_{x}\_{y}\_{z}” (trong đó x,y,z là năm, tháng, ngày hiện tại)

3. Tạo 1 tiến trình tên craw\_puuid. Tiến trình bao gồm 1 consumer nhận riot\_id từ topic “riot\_id\_{x}\_{y}\_{z}”, gọi API phù hợp để lấy được puuid ứng với mỗi riot\_id. Tiến trình cũng bao gồm 1 producer đẩy puuid thu thập được vào topic ‘puuid\_{x}\_{y}\_{z}” (trong đó x,y,z là năm, tháng, ngày hiện tại)

4. Tạo 1 tiến trình tên craw\_match\_id. Tiến trình bao gồm 1 consumer nhận puuid từ topic “puuid\_{x}\_{y}\_{z}”, gọi API phù hợp để lấy được các match\_id ứng với mỗi riot\_id. Tiến trình cũng bao gồm 1 producer đẩy match\_id thu thập được vào topic ‘match\_id\_{x}\_{y}\_{z}” (trong đó x,y,z là năm, tháng, ngày hiện tại)

5. Tạo 1 tiến trình tên craw\_match\_detail. Tiến trình bao gồm 1 consumer nhận puuid từ topic “puuid\_{x}\_{y}\_{z}”, gọi API phù hợp để lấy được các match\_detail ứng với mỗi riot\_id. Tiến trình cũng bao gồm 1 producer đẩy match\_detail thu thập được vào topic ‘match\_detail\_{x}\_{y}\_{z}” (trong đó x,y,z là năm, tháng, ngày hiện tại).

Chạy cả 5 tiến trình đồng thời.

Nhóm sử dụng Kafka cho việc craw này vì tính năng đánh dấu(offset) của Kafka. Điều này giúp cho nếu có lỗi xảy ra khiến cho tiến trình dừng lại, thì chỉ cần chạy lại tiến trình mà thôi, việc craw sẽ tiếp tục diễn ra tại offset đã đánh dấu, không cần phải craw lại từ đầu.

A computer screen with text and numbers

Description automatically generated

Cấu hình tự động cập nhật offset sau 300s cho consumer

A number of days and months

Description automatically generated with medium confidence

Các topic trong chương trình

* 1. Đẩy dữ liệu lên MongoDB

Tạo 1 consumer, nhận dữ liệu từ topic “match\_detail\_{x}\_{y}\_{z}”. Đọc từng message từ consumer và đẩy dữ liệu lên MongoDB  
A screen shot of a computer program

Description automatically generated

# **Xử lí dữ liệu với PySpark**

Dữ liệu mỗi trận đấu là 1 file json. Thông tin mỗi trận đấu rất nhiều, dưới đây là minh họa một phần dữ liệu 1 trận đấu. A screen shot of a computer code

Description automatically generated

Nhóm từ dữ liệu các trận đấu sẽ lấy ra thông tin tỉ lệ thắng mỗi tướng, các kèo đấu lợi thế và bất lợi cho mỗi tướng, phép bổ trợ, trang bị và ngọc bổ trợ phù hợp cho mỗi tướng. Nhóm thử chạy với 1 master và 2 worker cho Spark.

A screenshot of a computer screen

Description automatically generated

Trước tiên tạo 1 session cho Spark: A screenshot of a computer

Description automatically generated

Đọc thông tin từ dữ liệu và lấy ra các trận đấu chế độ summoner rift. Ở đây dữ liệu mỗi trận sẽ bao gồm participants là 1 mảng 10 phẩn tử tương ứng với thông tin 10 người chơi trong trận. Tách mỗi trận đấu thành các dòng trận đấu-người chơi.





Loại bỏ dữ liệu thừa trong dữ liệu:  


Lấy dữ liệu cần thiết trong mỗi dòng trận đấu-người chơi:  
A white background with red text

Description automatically generated

A screenshot of a computer

Description automatically generated

Tính tỉ lệ thắng:  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Ghi dữ liệu lên MongoDB



Các biển đổi khác, thầy có thể đọc chi tiết tại code của nhóm.

Chương trình chạy trên SparkUI:

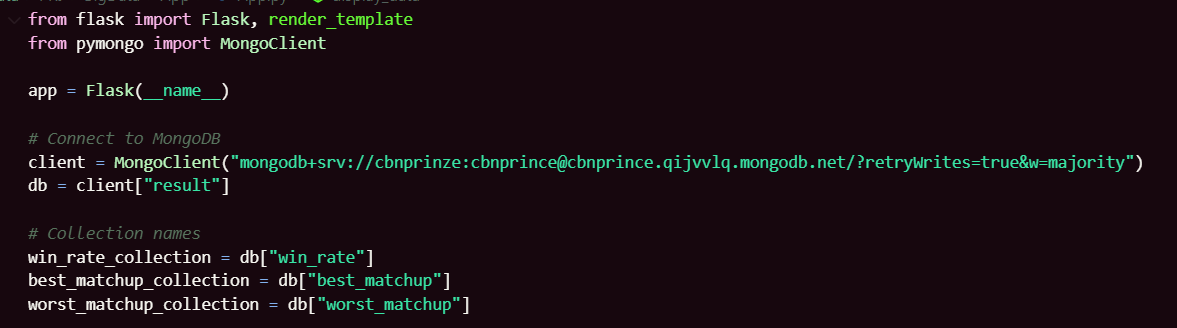
A group of blue text boxes

Description automatically generated

A screenshot of a computer

Description automatically generated

# **Visualize dữ liệu với Flask**



Đầu tiên chúng ta sẽ lấy dữ liệu đã xử lý từ các collection trên mongodb để hiển thị

Sau đó sử dụng Flask để hiển thị dữ liệu lên màn hình web

A screenshot of a video game

Description automatically generated