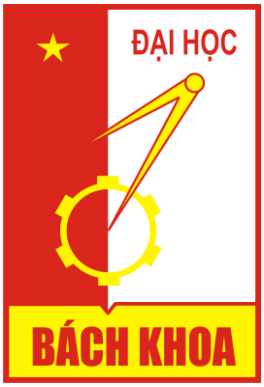
ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KỸ THUẬT MÁY TÍNH




# BÁO CÁO BÀI TẬP LỚN

**Đề tài: Lưu trữ và xử lý dữ liệu trò chơi điện tử trên điện thoại và máy tính bảng từ App store và Google play**

| | |
|---|---|
| Lớp | 144943 |
| Học phần | Lưu trữ và xử lý dữ liệu lớn |
| Mã học phần | IT4931 |
| Giảng viên hướng dẫn | TS. Trần Việt Trung |

Danh sách thành viên nhóm 1:

| Họ và tên | Mã số sinh viên |
|---|---|
| Trần Ngọc Bảo | 20215529 |
| Nguyễn Ngọc Bình Dương | 20204734 |
| Vũ Hồng Phước | 20204847 |
| Trần Bách Lưu Đức | 20200180 |
| Ha Duy Long | 20204841 |

Hà Nội, tháng 12 năm 2023

# Mục lục

# DANH MỤC HÌNH ẢNH

# CHƯƠNG 1: TỔNG QUAN

## 1.1 Giới thiệu

Ngày nay, bigdata trở thành một lĩnh vực vô cùng quan trọng trong cuộc sống với lượng dữ liệu khổng lồ được sinh ra trong quá trình vận động và phát triển của thế giới. Nhưng để khai thác hiệu quả nguồn tài nguyên dữ liệu này đòi hỏi cần có kiến thức và công cụ đi kèm.

Vậy nên, để tiếp cận với lĩnh vực này, nhóm chúng em quyết định chọn một loại dữ liệu đủ lớn trong khả năng để tiến hành tiến hành phân tích và lưu trữ đó là dữ liệu về "game trên các trang web google play và appstore". Các công đoạn khi thực hiện giải pháp này cơ bản sẽ bao gồm thu thập dữ liệu, lọc dữ liệu và biểu diễn, thống kê dữ liệu.

## 1.2 Mục tiêu đề tài

- Vận dụng được những kiến thức cơ bản về lưu trữ và xử lý dữ liệu lớn.
- Xây dựng được một hệ thống có khả năng thu thập được lượng dữ liệu lớn, xử lý và biểu diễn một cách trực quan.
- Thử nghiệm với những trường hợp, sự cố phát sinh trong hệ thống lưu trữ và xử lý như thêm node, mất node ...

## 1.3 Giới hạn đề tài

- Trong đề tài này nhóm thu thập, lưu trữ và xử lý dữ liệu về game trên web Google play và App store
-

## 1.4 Phương pháp

- Dựa trên những kiến thức đã học về cách lưu trữ và xử lý dữ liệu với hadoop và spark
- Thu thập tài liệu và tham khảo những dự án có liên quan

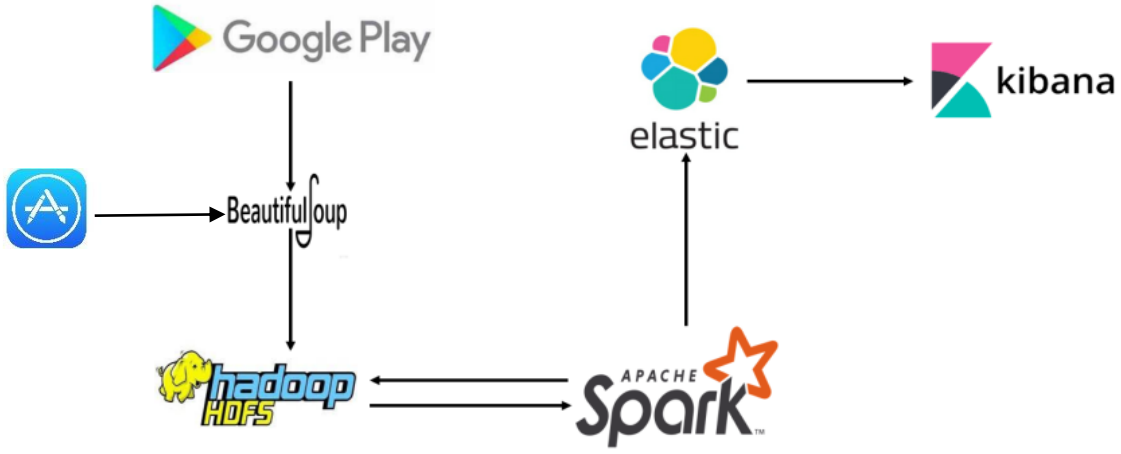
## 1.5 Bố cục báo cáo

Đề tài có tổng cộng 3 chương:

- **Chương 1 – Tổng quan**
  Trong chương này tìm hiểu các vấn đề hình thành nên đề tài. Kèm theo đó là một số nội dung và giới hạn của đề tài.
- **Chương 2 – Kiến trúc hệ thống**
  Giới thiệu các kiến thức nền tảng cũng như công nghệ và phần mềm được sử dụng trong đề tài bao gồm: kiến thức về tổ chức và lưu trữ dữ liệu với hadoop, xử lý dữ liệu với spark ...
- **Chương 3 – Các trải nghiệm khi xây dựng chương trình và hệ thống**
  Kết quả của của quá trình thu thập, lưu trữ và xử lý. Cùng với những trải nghiệm của nhóm trong suốt quá trình thực hiện.

# CHƯƠNG 2: TỔNG QUAN XÂY DỰNG HỆ THỐNG

## 2.1 Tổng quan hệ thống



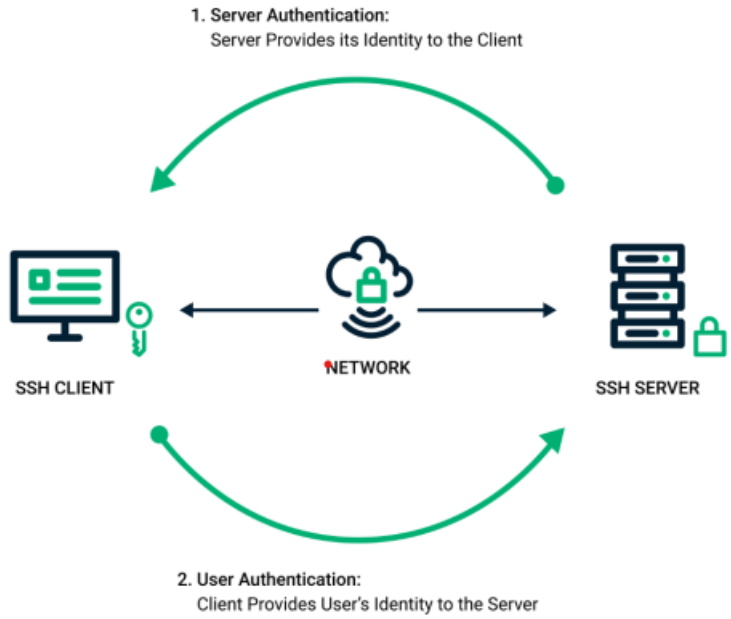

*Hình 2. 1 Kiến trúc hệ thống*

Hệ thống được xây dựng gồm 4 phần với các chức năng nhằm thu thập, xử lý, lưu trữ và trực quan hoá dữ liệu về game trong trang web. Các thành phần của hệ thống bao gồm:

1. Bộ phần thu thập dữ liệu: sử dụng BeautifulSoup4, là một thư viện để phân tích cú pháp các văn bảng dạng HTML và XML, chuyên dụng trong việc thu thập dữ liệu từ các trang web.
2. Bộ phận lưu trữ: hệ thống lưu trữ dữ liệu vào Hadoop dưới dạng HDFS File System (HDFS) để có thể lưu dữ liệu phân tán và có chức năng mở rộng, sao lưu, đảm bảo truy cập được khi một số máy mất kết nối.
3. Bộ phận xử lý dữ liệu: từ dữ liệu đã được lưu trong Hadoop, Spark được sử dụng để xử lý, làm sạch dữ liệu và thực hiện các truy vấn, giúp cho việc biểu diễn dữ liệu đơn giản hơn. Dữ liệu sau khi được làm sạch được lại được lưu về Hadoop và Elasticsearch.
4. Bộ phận biểu diễn dữ liệu: dữ liệu sau khi được xử lý bởi Spark được đưa vào Elasticsearch thông qua một thư viện mã nguồn mở là Elasticsearch for Apache Hadoop.

## 2.2 Chi tiết về thành phần hệ thống

### 2.2.1 SSH server




*Hình 2. 2 SSH server*

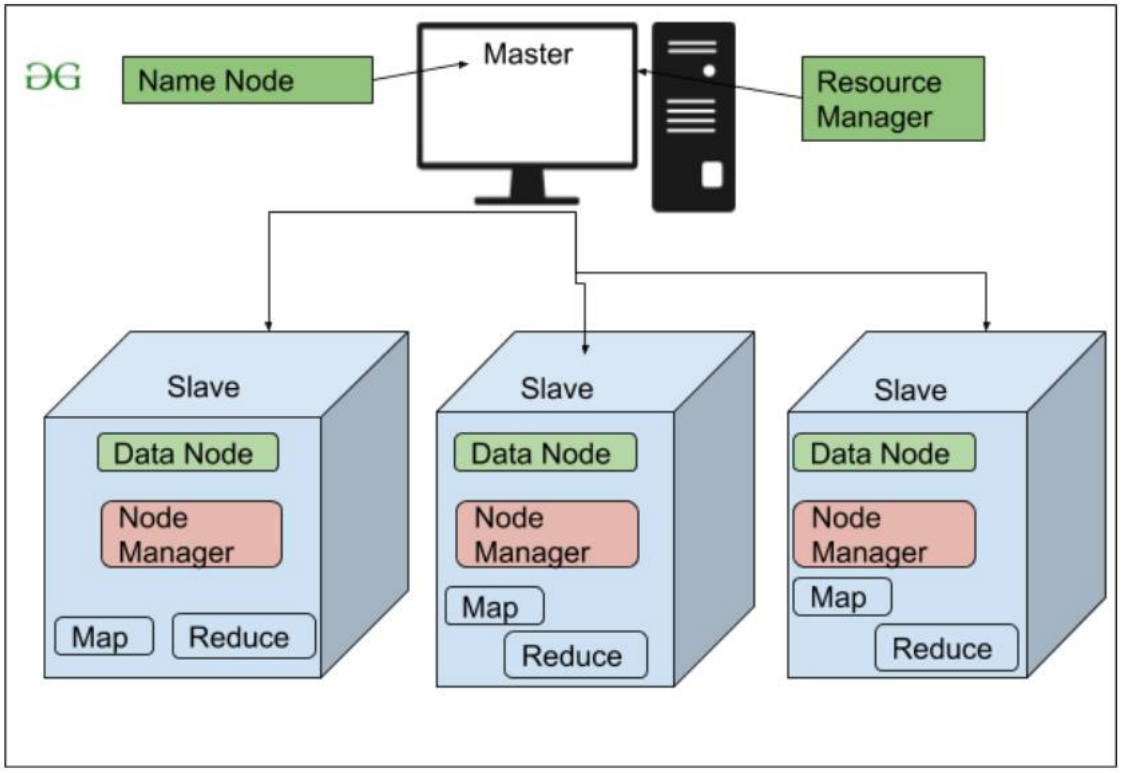
SSH, hay Secure (Socket) Shell, bao gồm cả giao thức mạng lẫn một bộ tiện ích để triển khai giao thức đó. SSH sử dụng mô hình client-server, kết nối một ứng dụng Secure Shell client (nơi session được hiển thị) với một SSH server (nơi session chạy). Triển khai SSH thường hỗ trợ cả các giao thức ứng dụng, dùng cho giả lập terminal hay truyền file.

Hadoop core sử dụng Shell (SSH) để giao tiếp với các slave node và để khởi chạy các quy trình máy chủ trên các slave node. Việc sử dụng cơ chế key-pair giúp việc giao tiếp giữa các máy không cần nhập nhiều lần mật khẩu mà vẫn đảm bảo độ bảo mật.

Khi Cluster đang hoạt động trong môi trường phân tán và việc giao tiếp cần thực hiện nhanh, SSH giúp cho NodeManager và các DataNode có thể giao tiếp với Namenode nhanh chóng.

### 2.2.2 Hadoop cluster

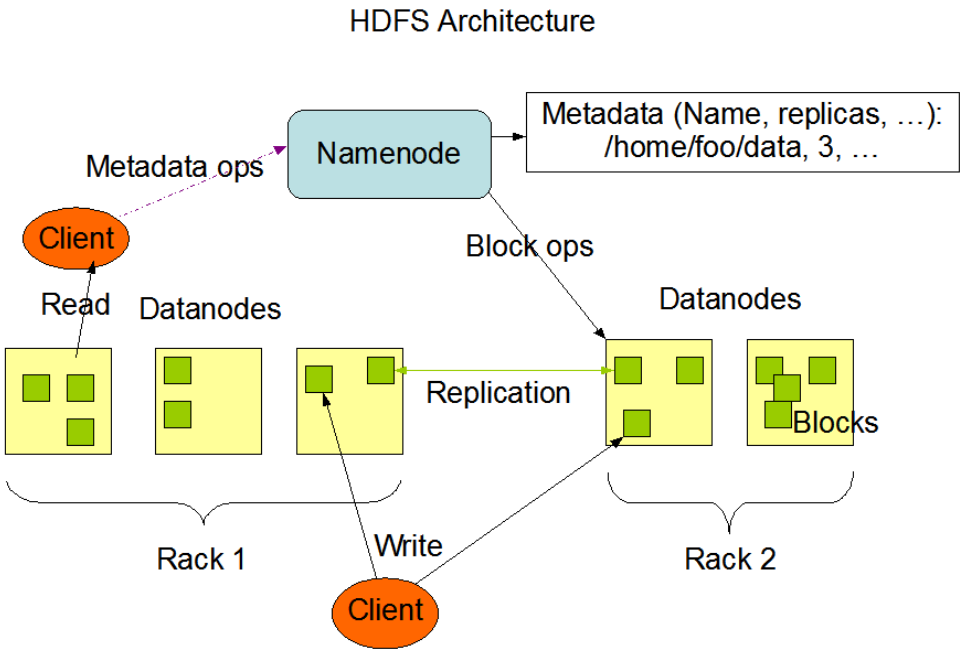
Hadoop Cluster là hệ thống file phân tán, cung cấp khả năng lưu trữ dữ liệu khổng lồ và tính năng tối ưu hoá việc sử dụng băng thông giữa các node.




*Hình 2. 3 Hadoop cluster*

Hadoop được cài đặt trên các máy tính trong hệ thống phân tán theo kiến trúc master – slave. Hadoop có thể hoạt động trên một máy (giống như 1 team chỉ có 1 member) hoặc mở rộng tới hàng ngàn máy, với mỗi máy đều có thể sử dụng để lưu trữ hoặc tính toán dữ liệu. Khi lưu trữ trên Hadoop, file dữ liệu được chia thành các chunk và được lưu thành nhiều bản sao, giúp cho cụm Hadoop có khả năng chịu lỗi.

HDFS là nơi lưu dữ liệu của Hadoop, HDFS chia chia nhỏ dữ liệu thành các đơn vị dữ liệu nhỏ hơn gọi là các blocks và lưu trữ chúng phân tán trong các node của cụm Hadoop. HDFS sử dụng kiến trúc master/slave, trong đó master gồm một Name Node để quản lý hệ thống file metadata v và một hay nhiều slave Data Nodes để lưu trữ dữ liệu.




*Hình 2. 4 HDFS architecture*

Đối với hệ thống phân tích thông tin tuyển dụng dữ liệu thu thập được trên Google play và App store sẽ được lưu trên cụm Hadoop. Cụm Hadoop bao gồm một Namenode/SecondaryNamenode và 2 Datanode. Khi lượng dữ liệu tăng lên, kiến trúc này có thể mở rộng thêm bằng cách bổ sung các Datanode để tăng cường dung lượng lưu trữ của hệ thống.

### 2.2.3 Spark cluster

Apache Spark là một framework xử lý dữ liệu mã nguồn mở trên quy mô lớn. Spark cung cấp một giao diện để lập trình các cụm tính toán song song với khả năng chịu lỗi.

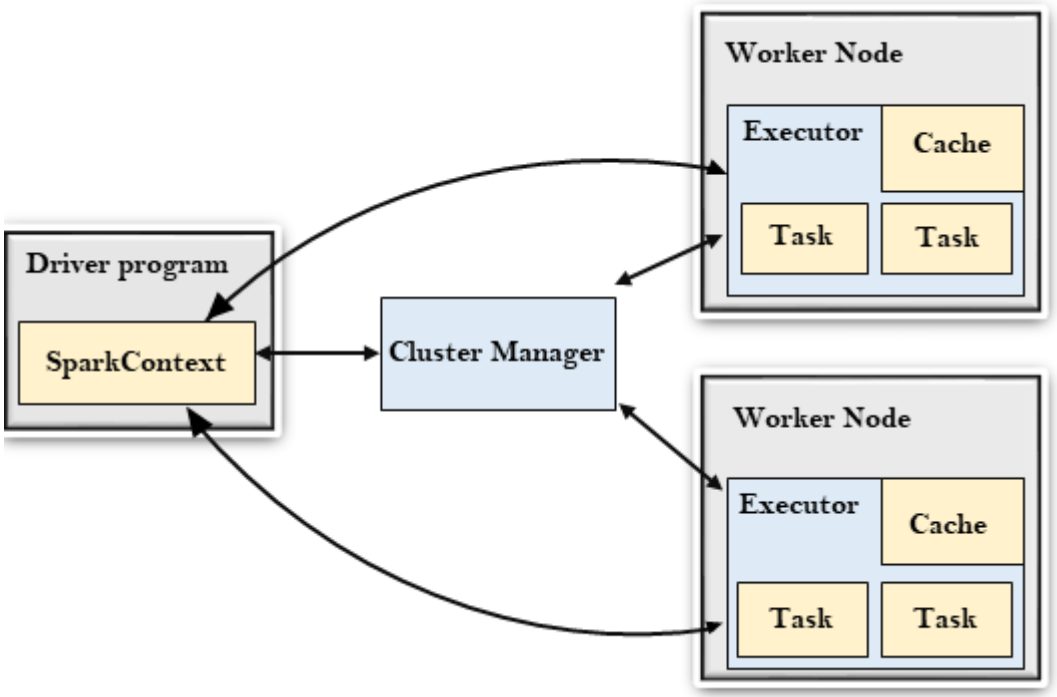
Tốc độ xử lý của Spark có được do việc tính toán được thực hiện cùng lúc trên nhiều máy khác nhau. Đồng thời việc tính toán được thực hiện hoàn toàn trên RAM.

Spark cho phép xử lý dữ liệu theo thời gian thực, vừa nhận dữ liệu từ các nguồn khác nhau đồng thời thực hiện ngay việc xử lý trên dữ liệu vừa nhận được. Những điểm nổi bật của Spark:

- Xử lý dữ liệu: Spark xử lý dữ liệu theo lô và theo thời gian thực.
- Tính tương thích: Có thể tích hợp với tất cả nguồn dữ liệu và định dạng tệp được hỗ trợ bởi cụm Hadoop.
- Hỗ trợ ngôn ngữ: Java, Python, Scala, R.
- Phân tích thời gian thực.

Kiến trúc của Spark bao gồm hai thành phần chính: trình điều khiển (driver) và trình thực thi (executors). Trình điều khiển dùng để chuyển đổi mã của người dùng thành nhiều tác vụ (tasks) có thể được phân phối trên các nút xử lý (worker nodes). Khi thực thi, trình điều khiển Driver tạo ra 1 SparkContext, sau đó giao tiếp với Cluster Manager để tính toán tài nguyên và phân chia các tác vụ đến cho các worker nodes.

Apache Spark xây dựng các lệnh xử lý dữ liệu của người dùng thành Đồ thị vòng có hướng hoặc DAG. DAG là lớp lập lịch của Apache Spark; nó xác định những tác vụ nào được thực thi trên những nút nào và theo trình tự nào.

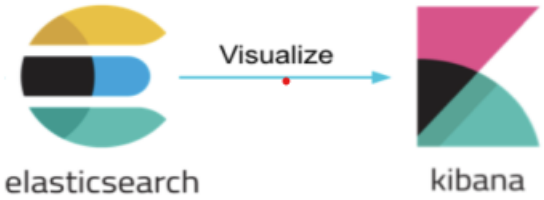



*Hình 2. 5 Spark cluster*

### 2.2.4 ElasticSearch và Kibana

Dữ liệu sau khi được làm sạch bởi Spark cần được biểu diễn dưới dạng bảng biểu, đồ thị để mang đến cho người dùng góc nhìn trực quan nhất. Elasticsearch và Kibana là những ứng dụng phù hợp để đảm nhận vai trò này. Là một công cụ tìm kiếm (với tốc độ gần thời gian thực) và phân tích dữ liệu phân tán, Elasticsearch có thể lưu trữ và phân tích nhiều loại dữ liệu khác nhau như: giữ liệu có cấu trúc, giữ liệu phi cấu trúc, giữ liệu số, dữ liệu về không gian địa lý, đánh chỉ mục dữ liệu một cách hiệu quả nhằm hỗ trợ quá trình tìm kiếm được thực hiện nhanh chóng. Các truy vấn trên Elasticsearch được thực hiện thông qua API, curl, python, hoặc qua Kibana. Kibana cung cấp giao diện đồ hoạ để người

dùng dễ dàng hơn trong việc khai phá, biểu diễn trực quan dữ liệu được lưu trên Elasticsearch.



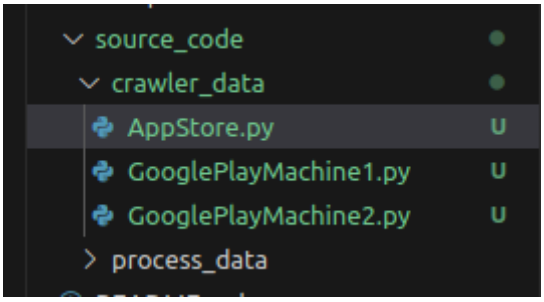

*Hình 2. 6 elasticsearch và kibana*

# CHƯƠNG 3: CÁC TRẢI NGHIỆM KHI XÂY DỰNG CHƯƠNG TRÌNH VÀ HỆ THỐNG

## 3.1 Thu thập dữ liệu

### Trải nghiệm 1: Tận dụng cấu trúc phân tán để thu thập dữ liệu

Có 3 file crawler dữ liệu  được chia đều cho 3 máy giúp quá trình thu thập diễn ra độc lập, nhanh chóng. Dữ liệu thu thập được ban đầu sẽ được lưu trong thư mục tmp của linux rồi sau đó sẽ được đẩy vào hdfs.




*Hình 3. 1 Tận dụng cấu trúc phân tán để thu thập dữ liệu*

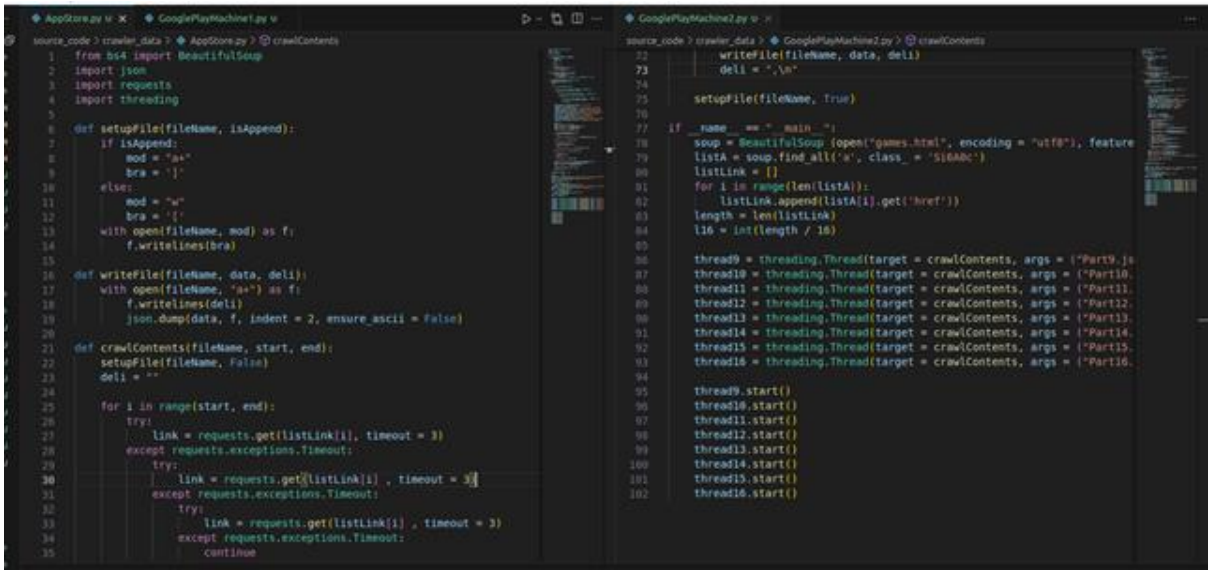
### Trải nghiệm 2: Lập trình đa luồng khi thu thập dữ liệu

Đối với App store: Hàng tuần thu thập được gần 400 game, dữ liệu thô dạng html trên website của mỗi game khoảng hơn 1 MB. Khi tăng số luồng lên 16 hay 32 thì tốc độ thu thập dữ liệu không thay đổi đáng kể. Link App store:

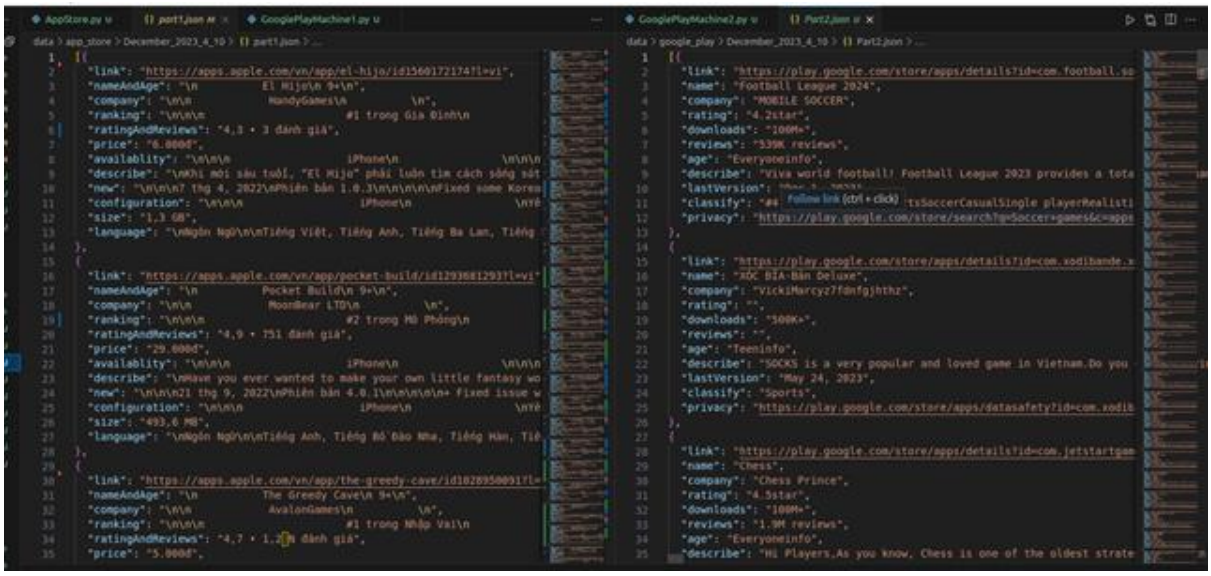
https://apps.apple.com/vn/genre/ios-tr%C3%B2-ch%C6%A1i/id6014?l=vi

Còn đối với Google play: Hàng tuần thu thập được gần 800 game, dữ liệu thô dạng html trên website của mỗi game khoảng hơn 3 MB. Tuy chia việc ra cho 2 máy nhưng số luồng tối đa máy có thể chạy ổn định là 8, khi tăng số luồng lên phần cứng của máy, tốc độ truy cập Internet không đáp ứng kịp khiến tốc độ thu thập dữ liệu còn giảm đi. Muốn tối ưu phải nâng cấp phần cứng của máy hoặc tăng số máy trong cụm. Link Google play:

https://play.google.com/store/games?device=phone&hl=vi-VN



*Hình 3. 2 Thu thập dữ liệu sử dụng đa luồng*

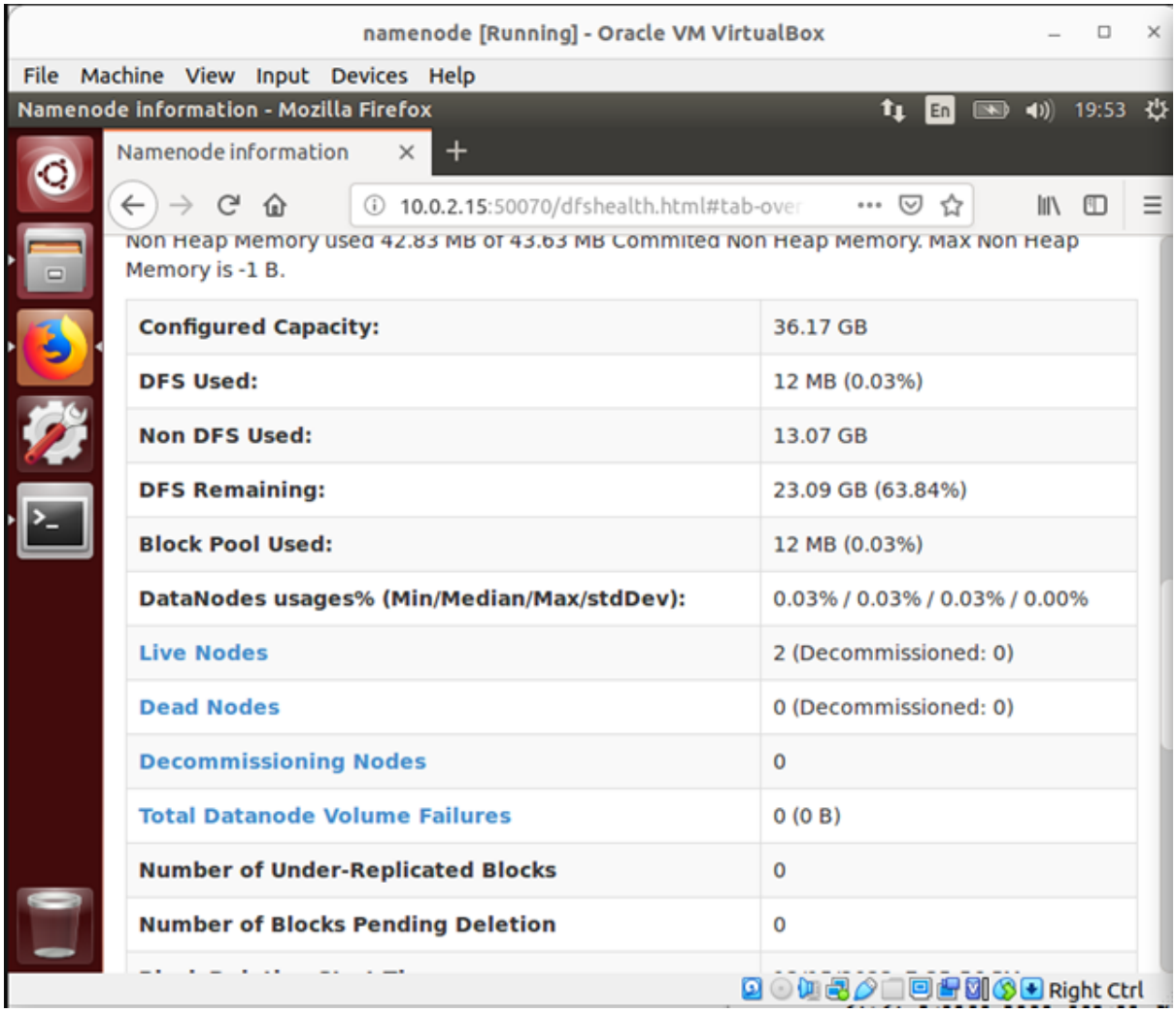


*Hình 3. 3 Kết quả thu thập dữ liệu từ các trang web*

## 3.2 Lưu dữ liệu vào hadoop

### Trải nghiệm 3: Lưu trữ dữ liệu phân tán trên cụm máy tính

Có 2 Live Nodes và 0 Dead Nodes chứng tỏ dữ liệu đã được lưu phân tán trên 2 máy datanode1 và datanode2




*Hình 3. 4 Lưu dữ liệu phân tán trên cụm máy tính*

**Trải nghiệm 4: Chống chịu lỗi trong hadoop bằng cách nhân bản dữ liệu**

Nhân bản 2 lần dữ liệu để chống chịu lỗi cho nên mỗi máy sẽ lưu trọn vẹn toàn bộ data trên nó.

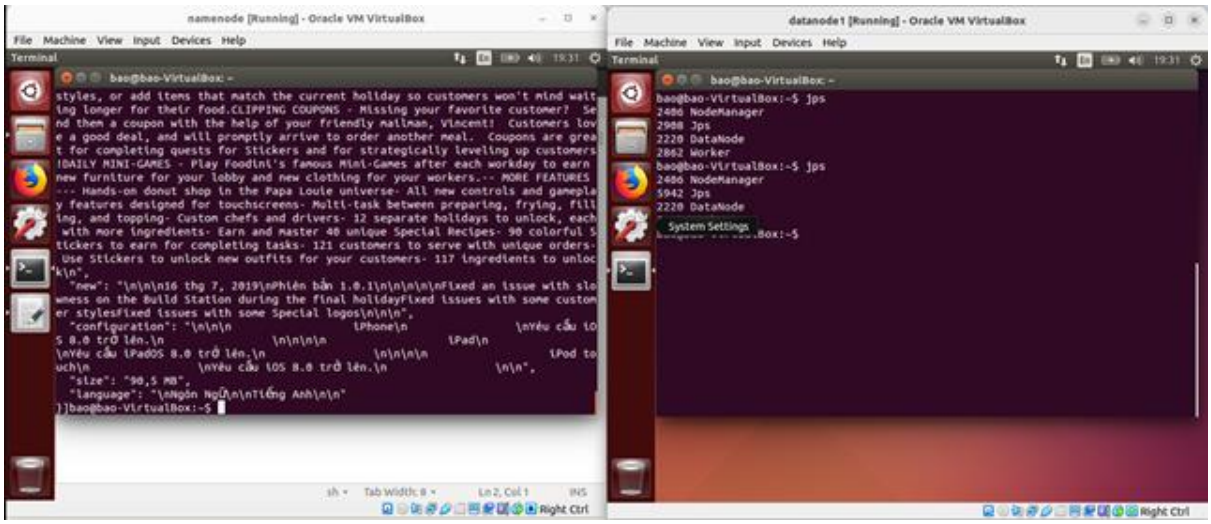
Dưới đây là hình ảnh các file được lưu trong hdfs của hadoop theo đường dẫn thực trong 2 máy datanode1 và datanode2.

*Hình 3. 5 Chống chịu lỗi trong Hadoop bằng cách nhân bản dữ liệu*

## Trải nghiệm 5 *: Thử loại bỏ các datanode để chứng minh khả năng chống chịu lỗi của Hadoop

Sau khi shut down máy ảo datanode2, cụm hadoop chỉ còn lại namenode và datanode1. Khi muốn xem dữ liệu trong hdfs của hadoop, chúng ta vẫn có thể xem được do đã config trong hdfs số bản sao là 2 cho nên datanode1 vẫn lưu trọn vẹn toàn bộ dữ liệu.

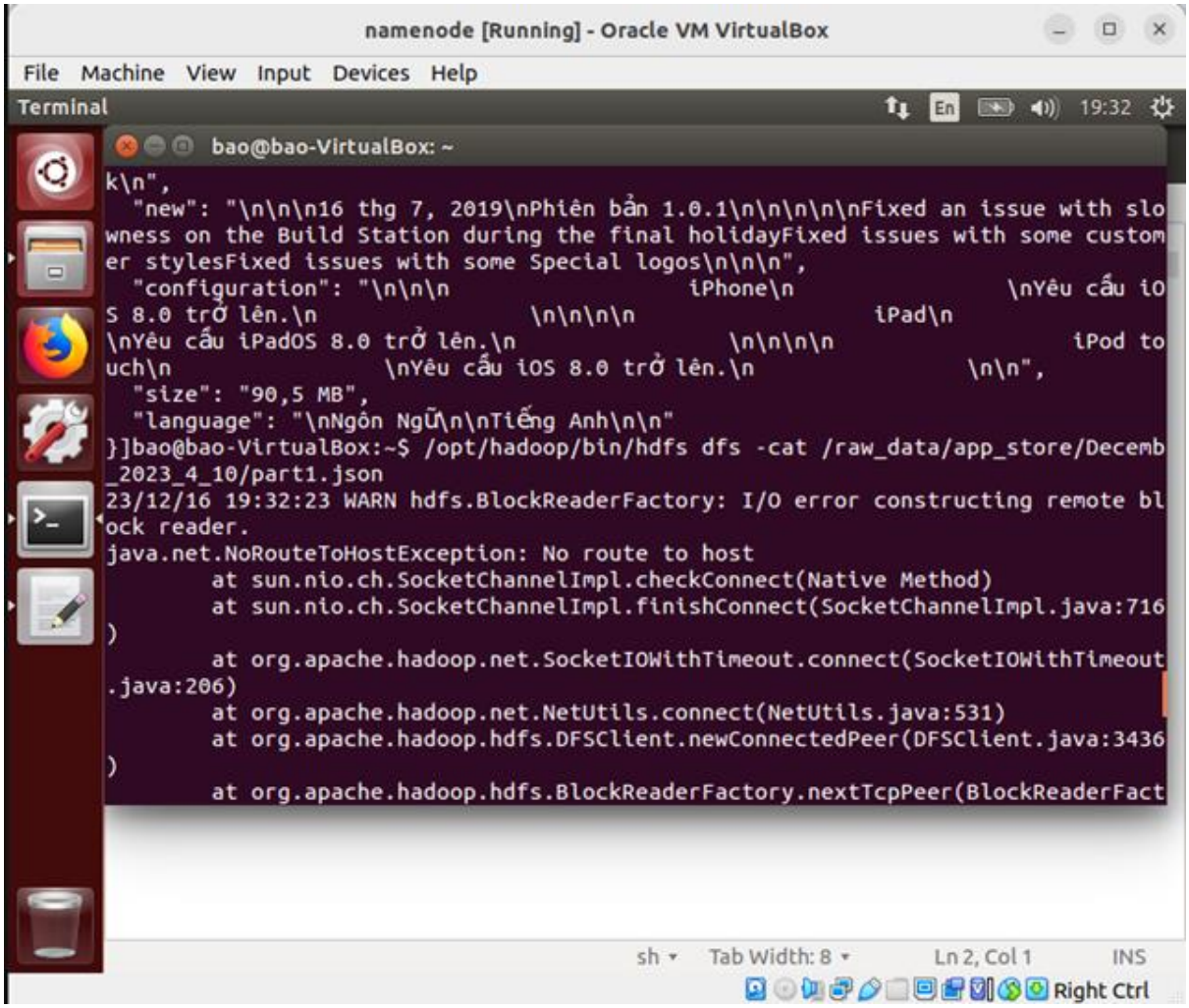
Dưới đây là hình ảnh khi muốn xem dữ liệu trong raw_data của app store



*Hình 3. 6 Kết quả khi xoá datanode 2*

Nếu shut down cả máy ảo datanode1, cụm hadoop chỉ còn lại namenode. Chúng ta sẽ không thể xem được dữ liệu nào cả.

Dưới đây cũng là hình ảnh khi muốn xem dữ liệu trong raw_data của app store tuy nhiên sẽ không thể cho ra kết quả.

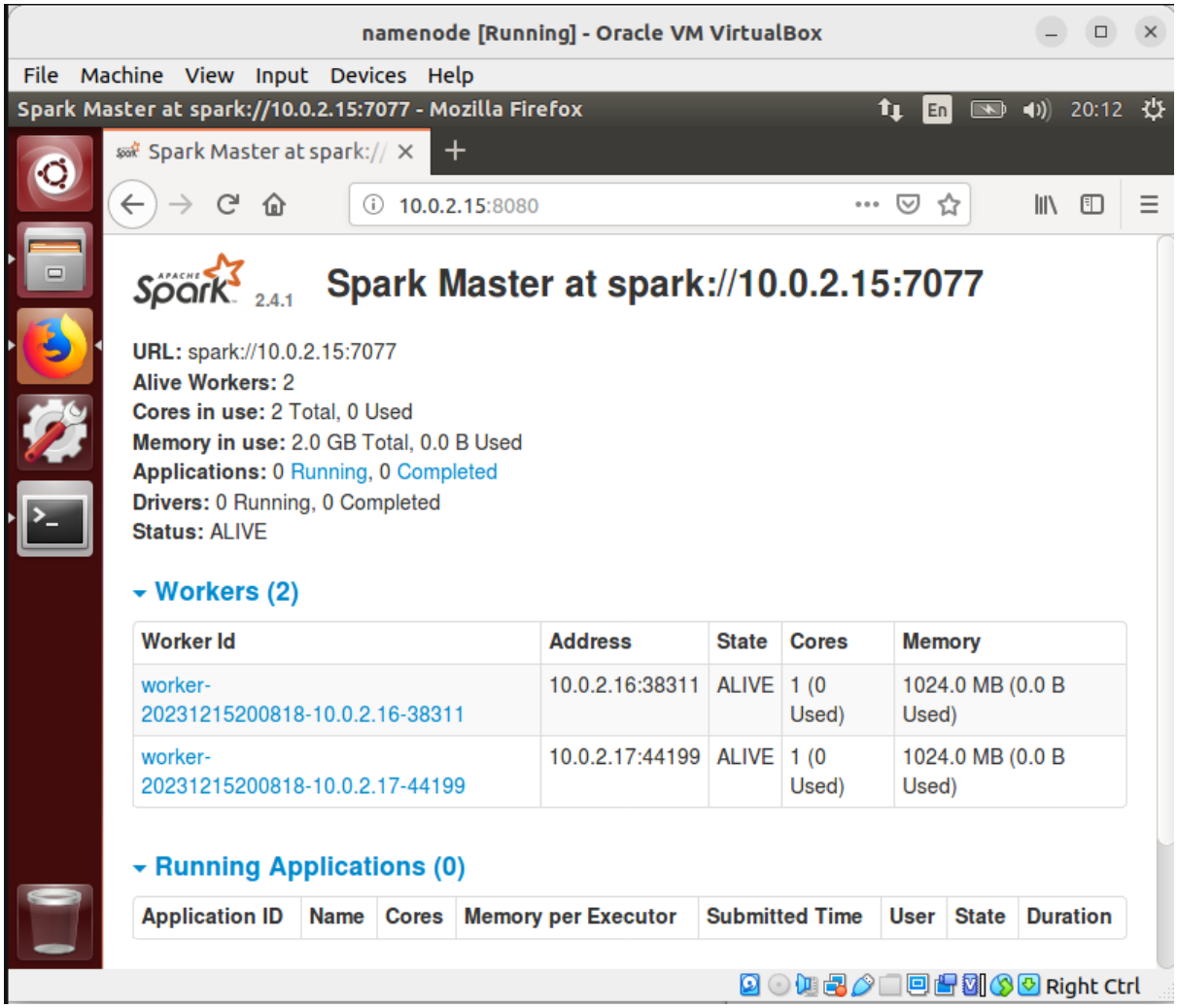



*Hình 3. 7 Kết quả sau khi shutdown cả hai datanode*

## 3.3 Xử lý dữ liệu bằng spark

### Trải nghiệm 6: Chạy spark cluster để xử lý dữ liệu

Cụm spark hoạt động với 1 master (máy namenode) và 2 worker (máy datanode1 và datanode2). Trong đó máy master chỉ điều khiển hoạt động còn việc xử lí dữ liệu sẽ do 2 máy worker đảm nhiệm.



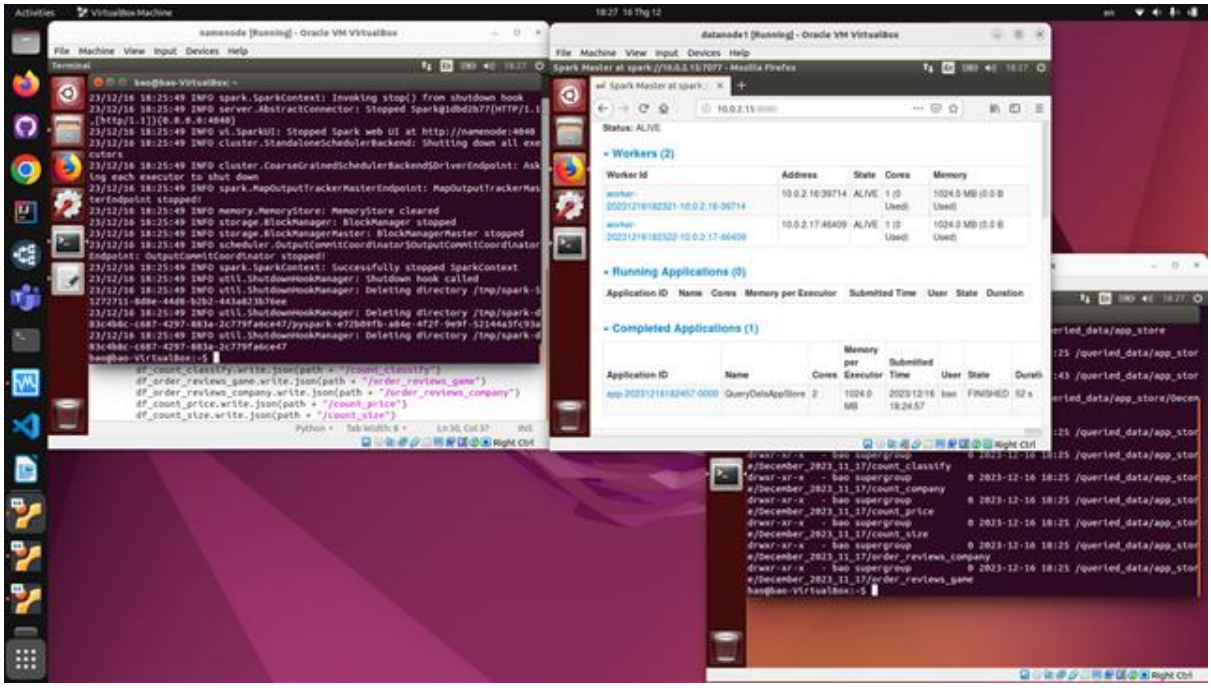

*Hình 3. 8 Khởi động spark cluster*

### Trải nghiệm 7: Lọc, truy vấn dữ liệu bằng pyspark

Quá trình xử lý dữ liệu bằng pyspark được chia thành 2 mục riêng biệt là app store và google play do cấu trúc dữ liệu thu thập được từ 2 nền tảng là khác biệt. Do đó các hàm lọc dữ liệu, các câu truy vấn dữ liệu của app store và google play là tương đối khác nhau.

Dữ liệu game của app store và google play được xử lý hàng tuần. Những đoạn code sau khi được đưa vào spark chạy được master chia thành các job phân chia cho 2 excutor (chính là 2 worker trong cụm spark) để thực hiện 1 cách nhanh chóng.

Dưới đây là hình ảnh sau khi cụm spark sau khi thực hiện trong xử lý dữ liệu app store trong 1 tuần, thời gian hoàn thành công việc và dữ liệu được tiếp tục lưu ngược trở lại hdfs của hadoop



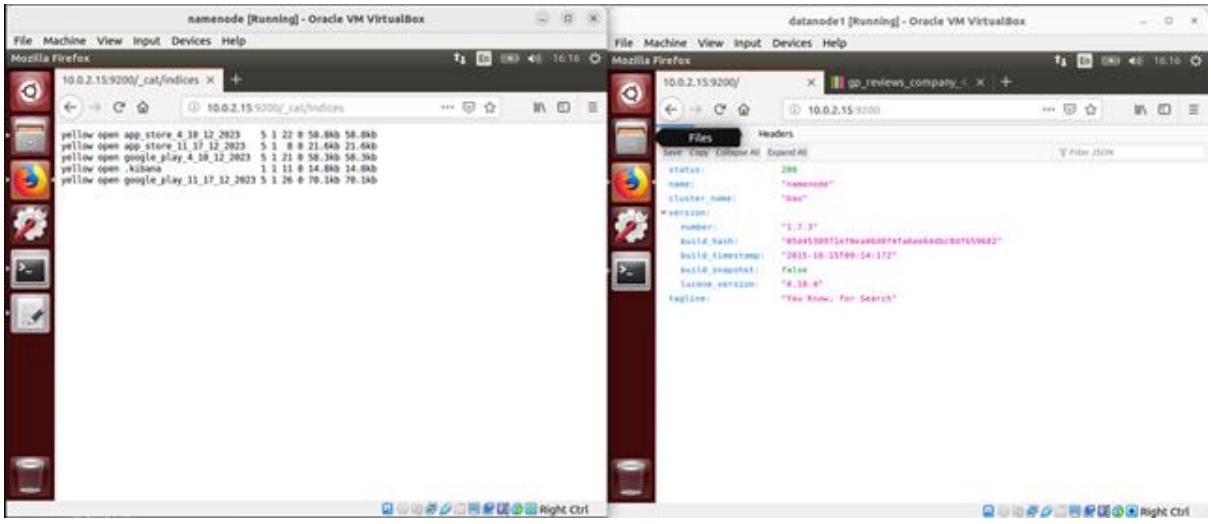
*Hình 3. 9 Xử lý dữ liệu app store trong một tuần*

## 3.4 Thống kê, biểu diễn dữ liệu bằng elasticsearch và kibana

### Trải nghiệm 8: Thiết lập cụm elasticsearch để lưu dữ liệu sau truy vấn

Thiết lập elasticsearch cluster tương tự với hadoop cluster. Những dữ liệu sau khi được truy vấn xong sẽ dùng các dòng lệnh trong terminal để post dữ liệu bằng giao thức http từ hdfs lên cụm elasticsearch.

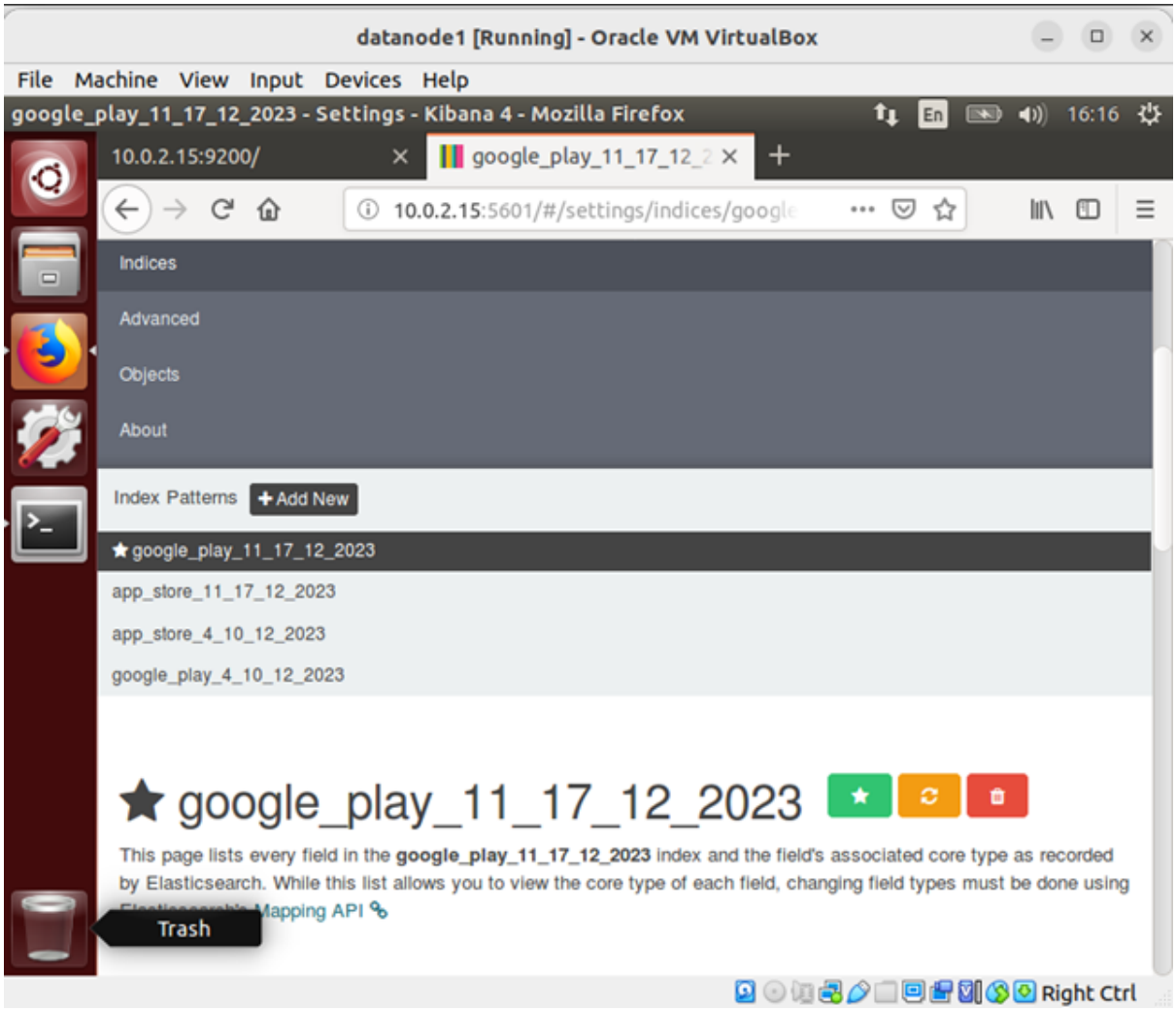
Elasticsearch, kibana được kết nối sẵn với nhau phục vụ cho việc biểu diễn dữ liệu.



*Hình 3. 10 Khởi động elasticsearch cluster*

**Trải nghiệm 9: Biểu diễn dữ liệu bằng kibana**

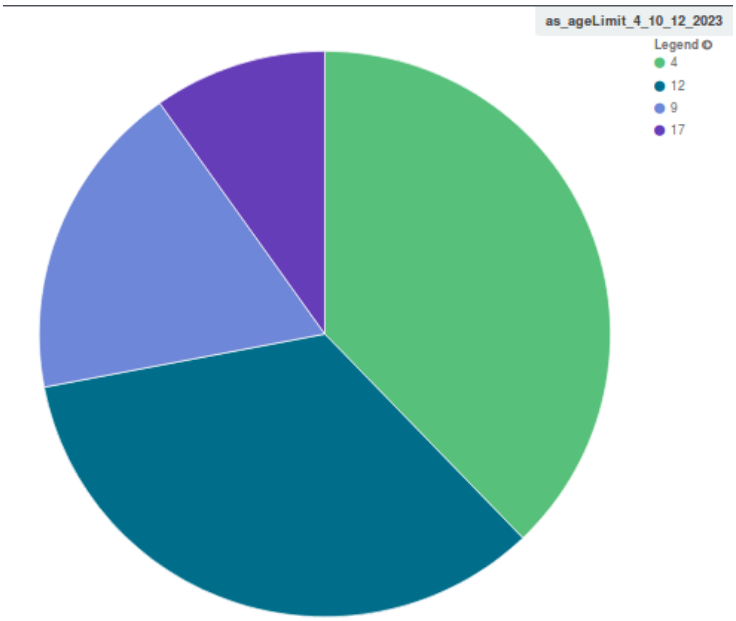
Thêm các index trong elasticsearch vào kibana để bắt đầu tiến hành biểu diễn dữ liệu bằng các biểu đồ trực quan.



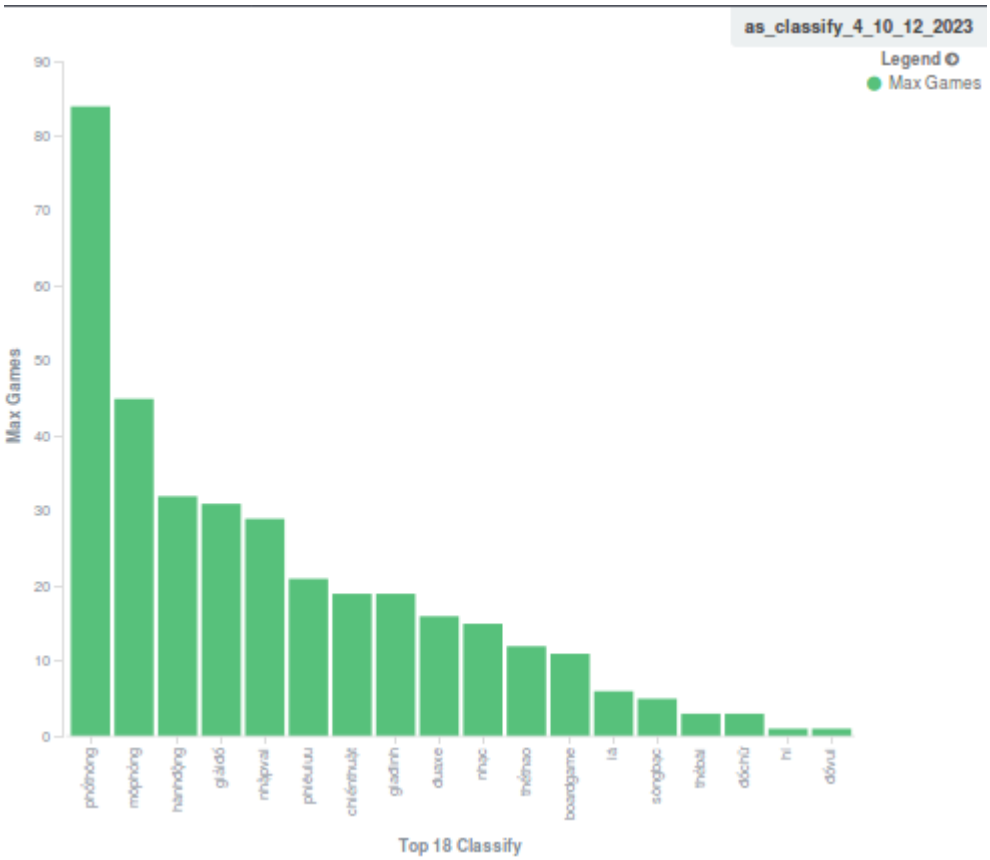

*Hình 3. 11 Khởi động kibana và vẽ biểu đồ*

## Các biểu đồ của dữ liệu 4-10/12/2023:

## App store

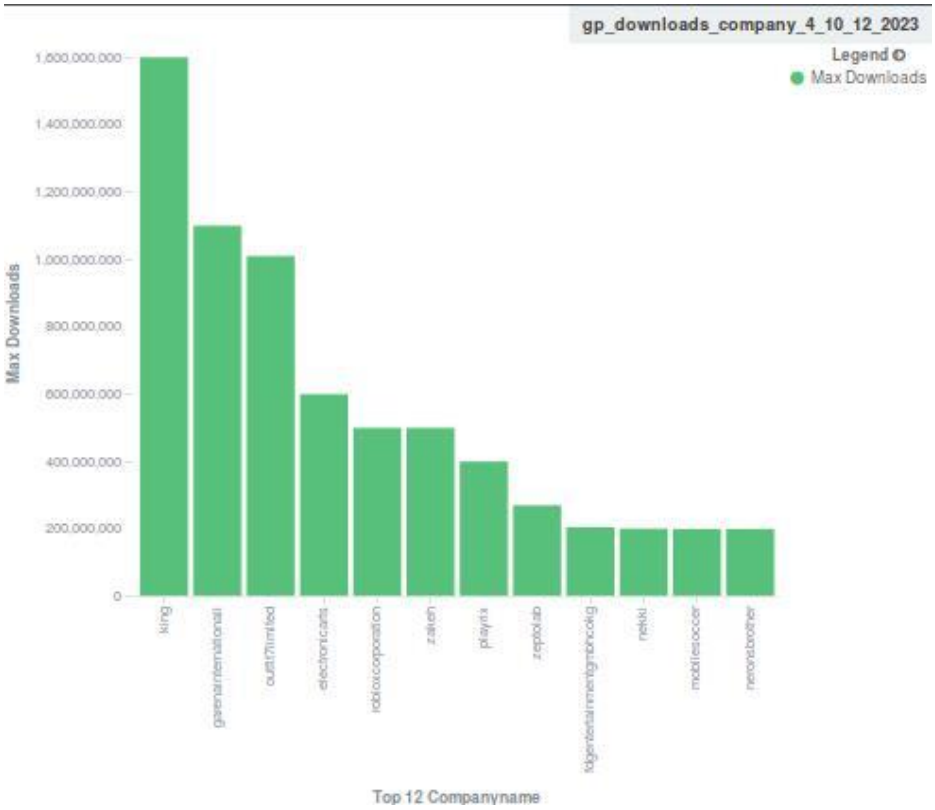



*Hình 3. 12 Phân bố game theo nhóm tuổi*

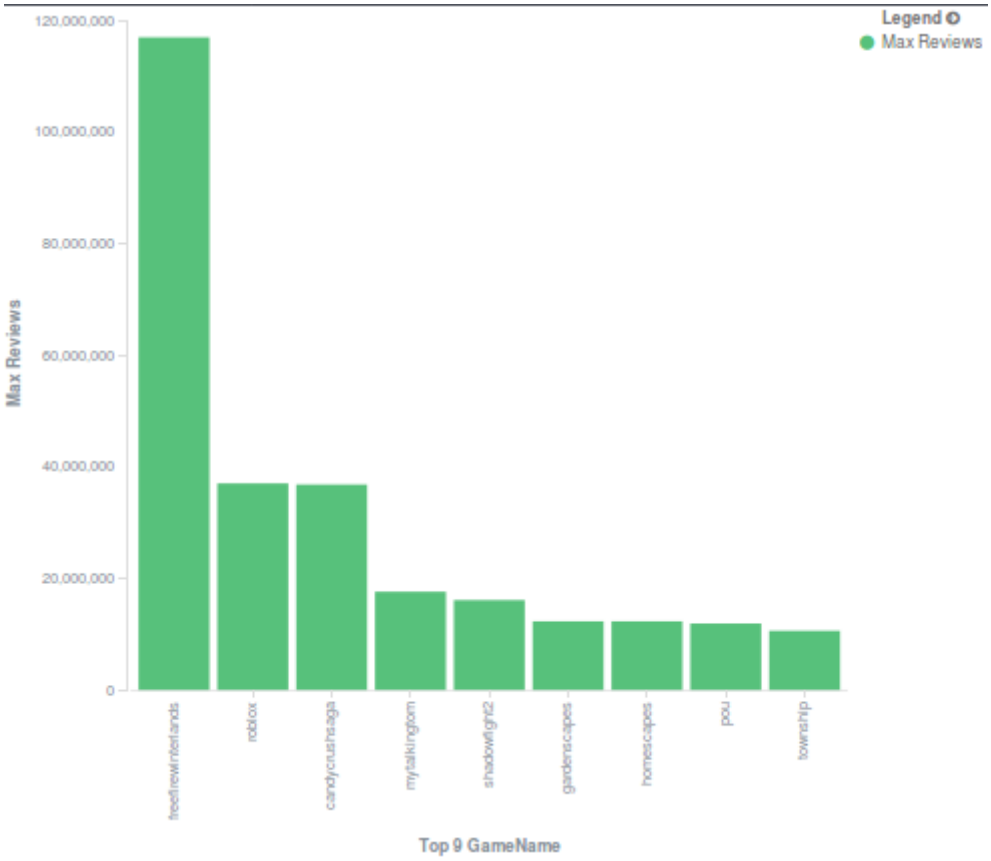



*Hình 3. 13 Thống kê số lượng game theo loại trò chơi*

## Google play



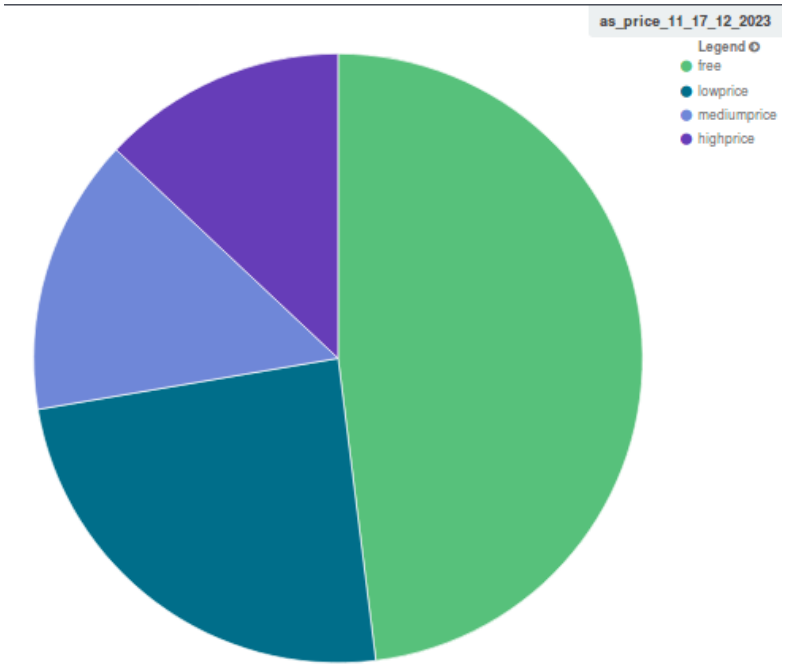

*Hình 3. 14 Top 12 công ty game có số lượt tải nhiều nhất*




*Hình 3. 15 Top 9 game có số lượt phản hồi nhiều nhất*

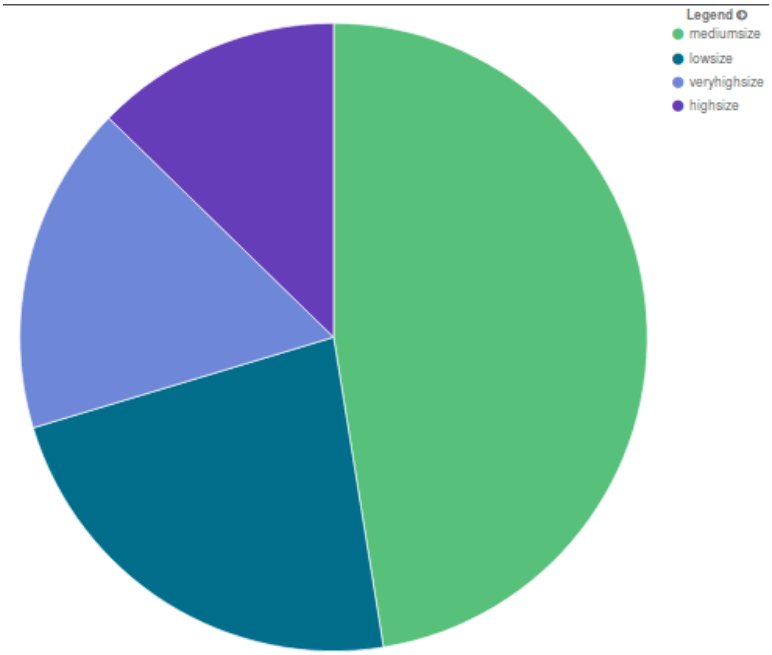
## Các biểu đồ dữ liệu 11-17/12/2023

## App store




*Hình 3. 16 Phân bố game theo giá thành*

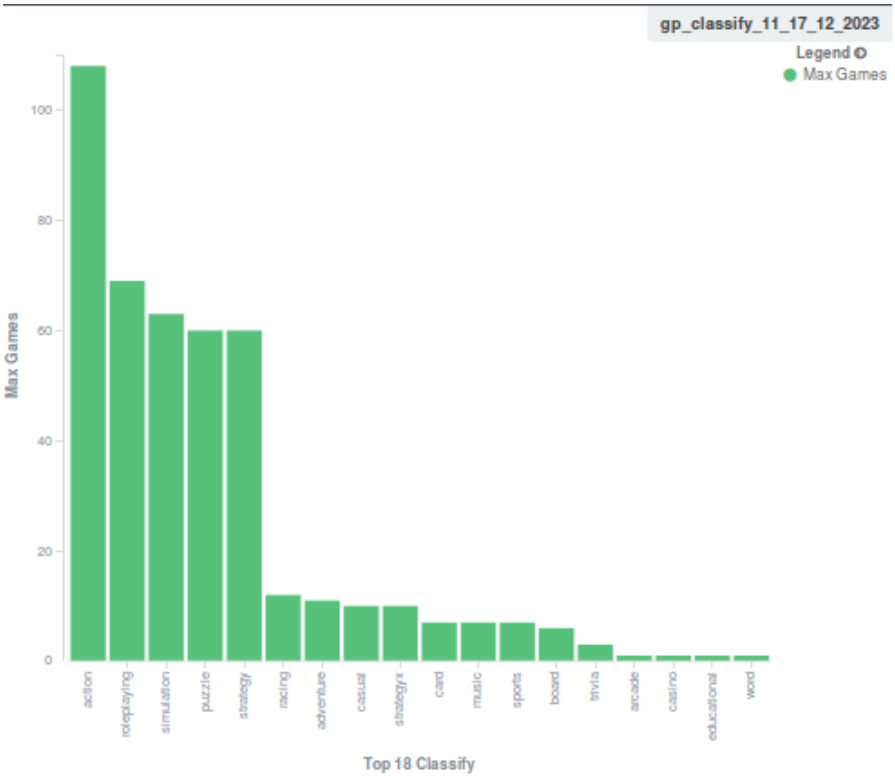
(free: 0đ; lowprice: 0-50.000đ; mediumprice: 50.000-100.000đ; highprice: >100.000đ)



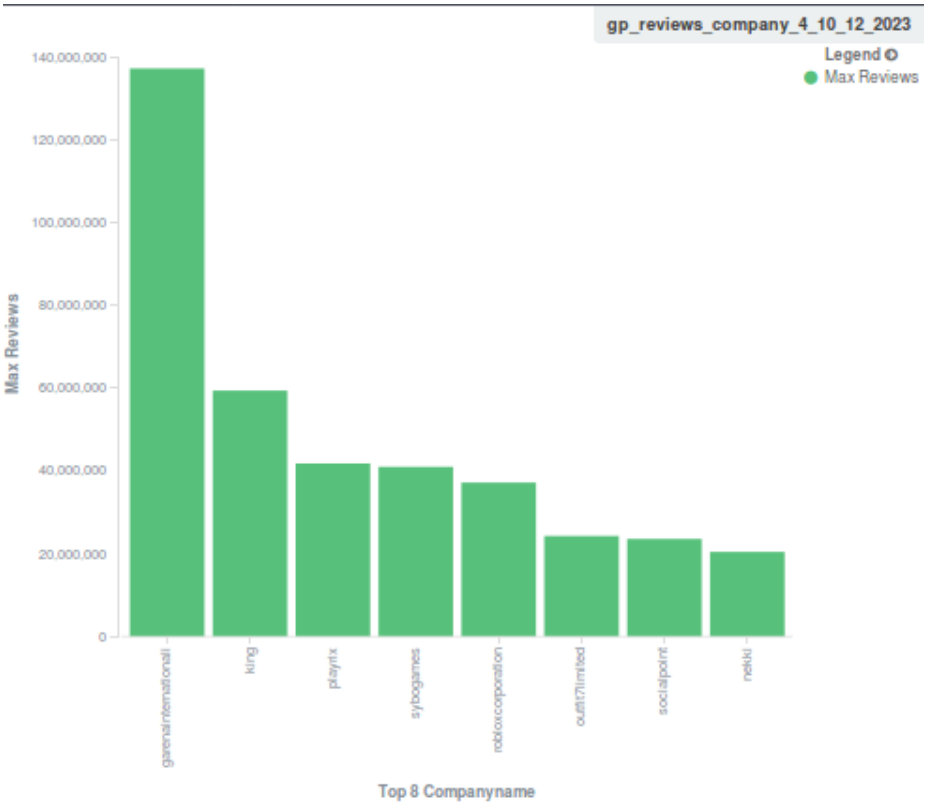

*Hình 3. 17 Phân bố game theo dung lượng*

(lowsize: <100MB, mediumsize: 100-500MB, highsize: 500-1000MB, veryhighsize: >1GB)

## Google play




*Hình 3. 18 Thống kê số lượng game theo loại trò chơi*




*Hình 3. 19 Top 8 công ty game có số lượt phản hồi nhiều nhất*

# CHƯƠNG 4: NHẬN XÉT, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

## 4.1 Nhận xét, đánh giá

Hệ thống cho thấy những lợi ích mà một hệ thống BigData đem lại như khả năng lưu trữ, tìm kiếm, biểu diễn lượng lớn dữ liệu, khả năng mở rộng khi lượng tài nguyên hiện tại không đủ, khả năng chịu lỗi trong một mạng phân tán khi có những thành phần trong mạng gặp trục trặc. Đây là những khả năng mà các hệ thống truyền thống không có hoặc khả năng đáp ứng còn hạn chế.

Bên cạnh đó, hệ thống của nhóm có một số nhược điểm. Việc sử dụng spark của nhóm không khai thác được tối đa hệ thống. Ngoài ra luồng thực hiện của hệ thống vẫn khá rời rạc, một số bước tải dữ liệu vẫn thực hiện bằng cách gõ code thủ công mà chưa được tự động hóa.

## 4.2. Hướng phát triển

Sử dụng Spark Streaming để phân tích và cải thiện tốc độ ghi dữ liệu. Nâng cấp phần cứng, tăng số node trong cụm. Thu thập dữ liệu game từ máy tính.

## DANH MỤC TÀI LIỆU THAM KHẢO


1. Bài giảng "Lưu trữ và xử lý dữ liệu lớn" – TS. Trần Việt Trung
2. Cấu hình hệ thống AnalyzeGameData/installation_instructions at master · Tran-Ngoc-Bao/AnalyzeGameData (github.com)
3. Giáo trình "Tổng quan về dữ liệu lớn (Big Data)" – Ks. Nguyễn Công Hoan – Trung Tâm Thông Tin Khoa học thống kê (Viện KHTK)