

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NHA TRANG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN HỌC
XỬ LÝ DỮ LIỆU LỚN (INT6216)**

**ỨNG DỤNG HADOOP VÀ SPARK TRONG XỬ LÝ
LOG MÁY CHỦ**

Sinh viên thực hiện: Nguyễn Hữu Nghĩa

MSSV: 63134775

Lớp: 63.CNTT-3

Giảng viên: TS. Nguyễn Đình Hưng

Khánh Hòa - 2025

Lời cam đoan

Em xin cam đoan rằng đồ án môn học do em thực hiện. Các nội dung nghiên cứu, số liệu và kết quả nghiên cứu và thực hiện của riêng em. Tất cả nội dung, số liệu và kết quả trong đồ án này đều trung thực. Tất cả phần mềm sử dụng trong đồ án này đều là mã nguồn mở. Nếu phát hiện có bất kỳ sự gian lận nào, tôi xin chịu hoàn toàn trách nhiệm.

Nguyễn Hữu Nghĩa

KẾT QUẢ ĐÁNH GIÁ ĐỒ ÁN MÔN HỌC

Họ tên sinh viên: Nguyễn Hữu Nghĩa
MSSV: 63134775
Lớp: 63CNTT-3

Nội dung	Trọng số	Điểm
1. Giải quyết vấn đề		
1.1. Phân tích bài toán; thu thập, khảo sát và chuẩn bị dữ liệu; thiết kế giải thuật	20%	
1.2. Cài đặt, triển khai ứng dụng trên Hadoop	20%	
1.3. Cài đặt, triển khai ứng dụng trên Spark	20%	
2. Báo cáo bài tập lớn		
2.1. Nội dung báo cáo	20%	
2.2. Vấn đáp	20%	
Điểm trung bình		

Giảng viên

Nguyễn Đình Hưng

Mục lục

1	GIỚI THIỆU	1
1.1	Tổng quan về dữ liệu lớn	1
1.2	Mục tiêu của đề tài	1
1.3	Cấu trúc của Đồ án	1
2	NỘI DUNG VÀ PHƯƠNG PHÁP THỰC HIỆN	2
2.1	Phân tích bài toán	2
2.2	Thu thập và chuẩn bị dữ liệu	2
2.3	Cài đặt và triển khai ứng dụng trên Hadoop	2
2.3.1	Cài đặt Hadoop	2
2.3.2	Xây dựng giải thuật	2
2.3.3	Lập trình ứng dụng	4
2.3.4	Thực thi ứng dụng	5
2.4	Cài đặt và triển khai ứng dụng trên Spark	6
2.4.1	Cài đặt Spark	6
2.4.2	Lập trình ứng dụng	7
2.4.3	Thực thi ứng dụng	8
3	KẾT LUẬN	9
3.1	Đánh giá chung	9
3.1.1	Những kết quả đạt được	9
3.1.2	Một số hạn chế	9
3.2	Hướng phát triển	9
	Tài liệu tham khảo	9

Danh sách hình vẽ

2.1	Giao diện cài đặt Hadoop	3
2.2	Giao diện hiện kết quả trên Terminal	5
2.3	Giao diện hiện kết quả trên Hadoop UI	6
2.4	Giao diện cài đặt PySpark	7
2.5	Giao diện hiện kết quả	8

Danh sách giải thuật

1	Pha Map xử lý liên kết đến trang web và log lỗi	4
2	Pha Reduce xử lý liên kết đến trang web	4

Chương 1

GIỚI THIỆU

1.1 Tổng quan về dữ liệu lớn

Trong thời đại công nghệ số, dữ liệu lớn (Big Data) đang trở thành một lĩnh vực quan trọng, thu hút sự quan tâm nghiên cứu và ứng dụng trong nhiều ngành nghề. Khai thác hiệu quả dữ liệu lớn không chỉ giúp khám phá những tri thức ẩn giấu mà còn mang lại lợi thế cạnh tranh cho doanh nghiệp và tổ chức. Nhờ vào việc phân tích dữ liệu một cách chính xác, các đơn vị có thể tối ưu hóa quy trình hoạt động, dự đoán xu hướng, cải thiện chất lượng dịch vụ và ra quyết định chiến lược hiệu quả hơn.

1.2 Mục tiêu của đề tài

Các mục tiêu chính của đề án:

- Tìm hiểu tổng quan về dữ liệu lớn và các ứng dụng trong thực tế
- Tìm hiểu các phương pháp, công nghệ và công cụ tiêu biểu trong xử lý dữ liệu lớn, đặc biệt là Hadoop và Spark
- Áp dụng kiến thức đã học để xây dựng một chương trình phân tích log máy chủ bằng công nghệ Big Data
- Thực hiện các thống kê về số lượng dòng nhật ký và số dòng báo lỗi theo từng ngày nhằm hỗ trợ giám sát hệ thống.

1.3 Cấu trúc của Đề án

Đề án gồm các phần như sau:

- Chương 1: Giới thiệu.
- Chương 2: Nội dung và phương pháp thực hiện.
- Chương 3: Kết luận.

Chương 2

NỘI DUNG VÀ PHƯƠNG PHÁP THỰC HIỆN

2.1 Phân tích bài toán

Bộ dữ liệu log máy chủ được lưu dưới dạng tệp văn bản, trong đó mỗi dòng nhật ký chứa các thông tin sau:

- **Thời điểm:** Ghi nhận ngày, giờ của sự kiện log.
- **Loại nhật ký:** Phân loại log theo các mức như `[notice]`, `[error]`, `[warning]`, v.v.
- **Nội dung:** Chứa thông tin chi tiết về sự kiện xảy ra.

Yêu cầu: Với mỗi ngày, thực hiện tổng hợp:

1. Tổng số dòng nhật ký theo từng ngày.
2. Số lượng dòng báo lỗi (`[error]`) theo từng ngày.

2.2 Thu thập và chuẩn bị dữ liệu

Trong đề tài này em sử dụng dữ liệu từ **Apache**.

2.3 Cài đặt và triển khai ứng dụng trên Hadoop

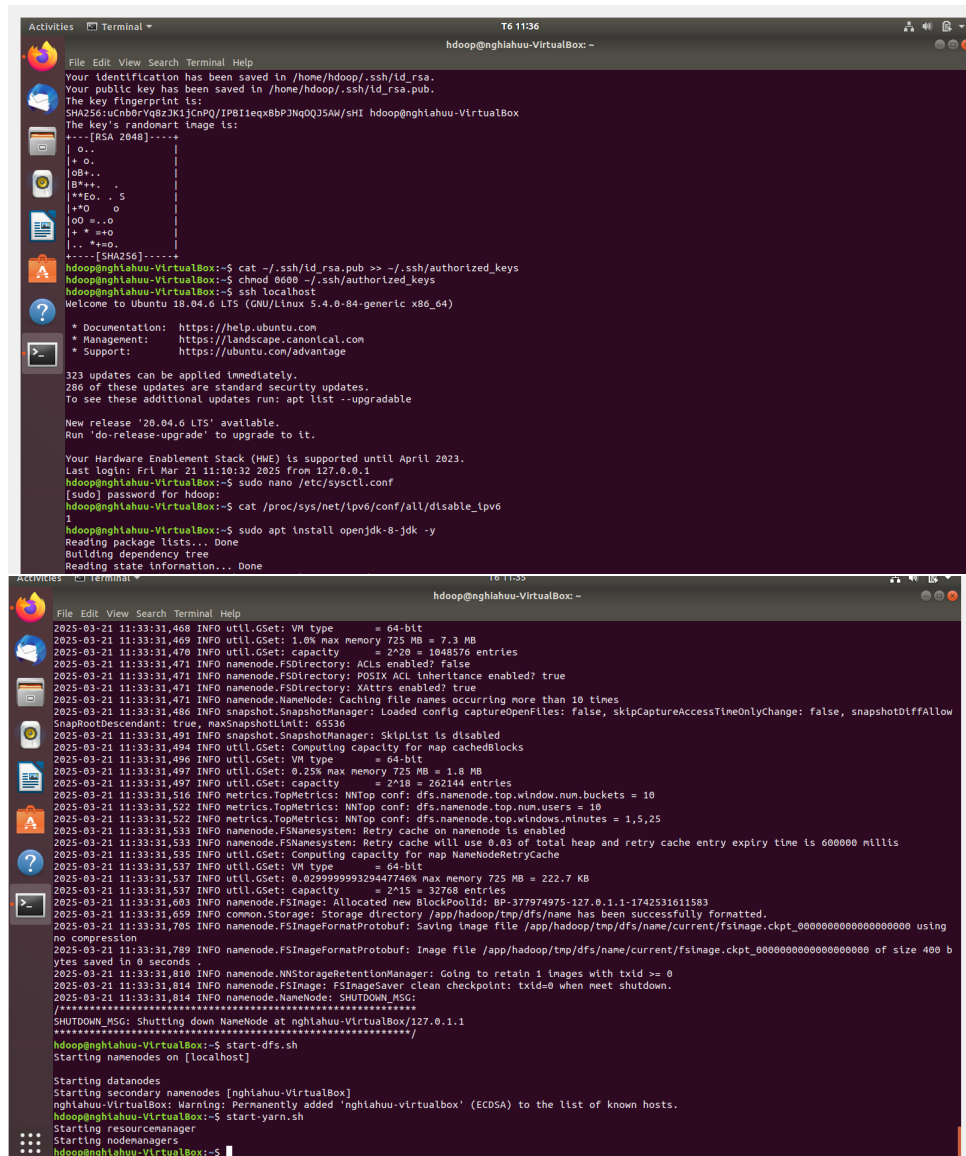
2.3.1 Cài đặt Hadoop

Quá trình cài đặt Hadoop bao gồm các bước sau:

1. Tải về và giải nén Hadoop.
2. Thiết lập các biến môi trường cho Hadoop.
3. Thiết lập cấu hình Hadoop.
4. Thiết lập cấu hình Hadoop core.
5. Thiết lập YARN.
6. Vận hành Hadoop.

2.3.2 Xây dựng giải thuật

Giải thuật MapReduce tổng hợp liên kết đến trang Web.



Hình 2.1: Giao diện cài đặt Hadoop

Pha Map

Giải thuật 1: Pha Map xử lý liên kết đến trang web và log lỗi

```
1: for mỗi dòng trong stdin do
2:   Tách dòng thành các phần bằng dấu cách
3:   if số phần nhỏ hơn 4 then
4:     Bỏ qua dòng
5:   end if
6:   Tìm chuỗi phù hợp với định dạng thời gian bằng regex
7:   if tìm thấy then
8:     Trích xuất ngày tháng từ chuỗi khớp
9:     print(ngày, 1)
10:    if dòng chứa từ khóa "error" then
11:      print(ngày_error, 1)
12:    end if
13:  end if
14: end for
```

Pha Reduce

Giải thuật 2: Pha Reduce xử lý liên kết đến trang web

```
1: Khởi tạo: current_key ← None, current_count ← 0
2: for mỗi dòng trong sys.stdin do
3:   Chia dòng thành key, count
4:   count ← ép kiểu thành số nguyên
5:   if current_key bằng key then
6:     Cộng count vào current_count
7:   else
8:     if current_key không rỗng then
9:       In kết quả: current_key và current_count
10:    end if
11:    Cập nhật current_key ← key
12:    Cập nhật current_count ← count
13:  end if
14: end for
15: if current_key không rỗng then
16:   In kết quả: current_key và current_count
17: end if
```

2.3.3 Lập trình ứng dụng

Lập trình pha Map

```
#!/usr/bin/env python3
import sys
import re

for line in sys.stdin:
    parts = line.strip().split()
    if len(parts) < 4:
        continue # Bỏ qua dòng không hợp lệ

    # Trích xuất thời gian (dạng: [Sun Dec 04 04:47:44 2005])
    match = re.search(r'\[(\w+)\s+(\w+)\s+(\d+)\s+(\d+):(\d+):(\d+)\s+(\d+)\]', line)
    if match:
        day = f"{match.group(3)}-{match.group(2)}-{match.group(7)}" # Định dạng: 04-Dec-2005
```

```
print(f"{day}\t1") # Xuất dữ liệu cho reducer

# Kiểm tra nếu là dòng báo lỗi (chứa "error")
if "error" in line.lower():
    print(f"{day}_error\t1")
```

Lập trình pha Reduce

```
#!/usr/bin/env python3
import sys

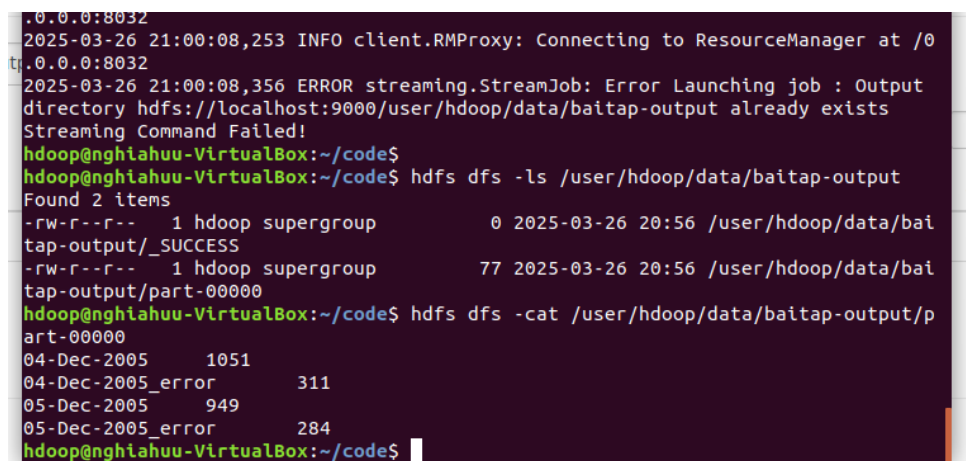
current_key = None
current_count = 0

for line in sys.stdin:
    key, count = line.strip().split("\t")
    count = int(count)

    if current_key == key:
        current_count += count
    else:
        if current_key:
            print(f"{current_key}\t{current_count}")
            current_key = key
            current_count = count

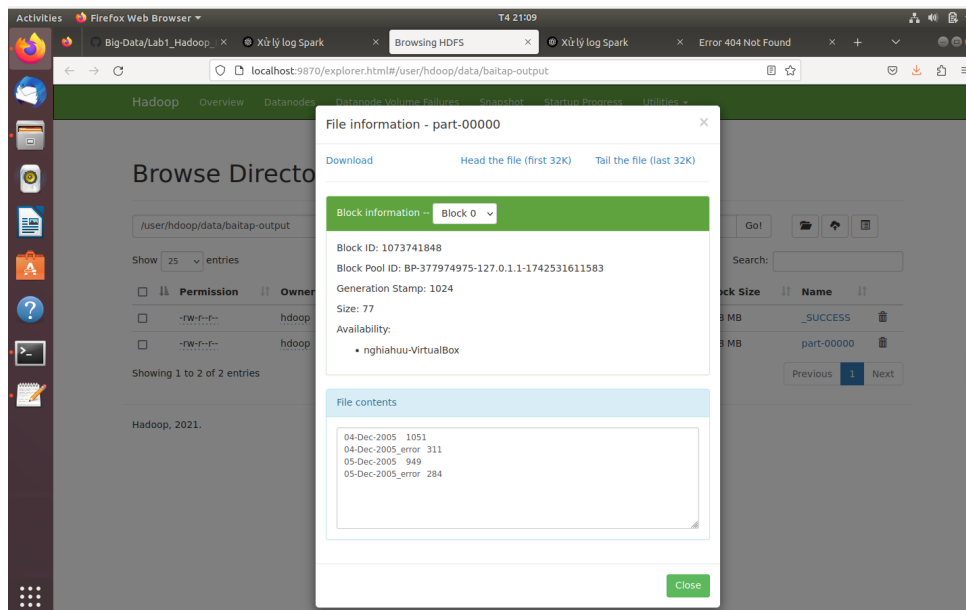
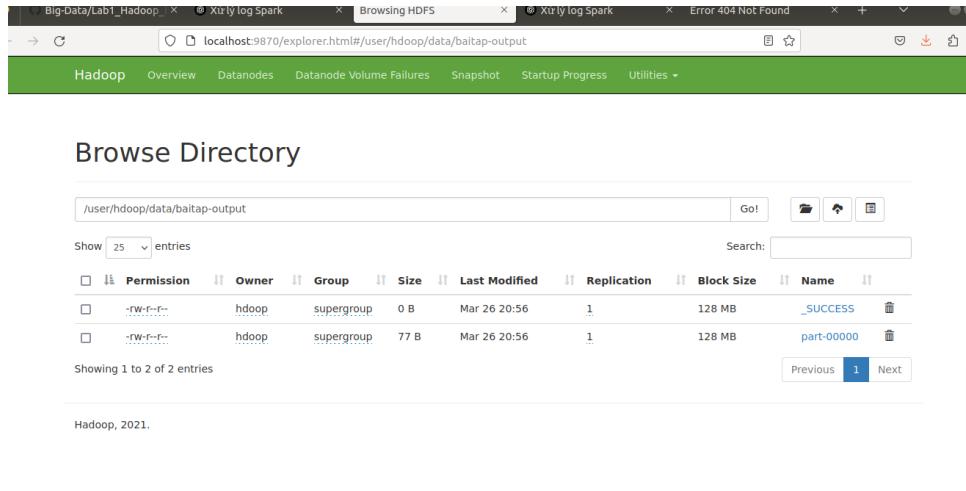
# In kết quả của dòng cuối cùng
if current_key:
    print(f"{current_key}\t{current_count}")
```

2.3.4 Thực thi ứng dụng



```
hadoop@ngghiahuu-VirtualBox:~/code$ hdfs dfs -ls /user/hadoop/data/baitap-output
Found 2 items
-rw-r--r-- 1 hadoop supergroup 0 2025-03-26 20:56 /user/hadoop/data/baitap-output/_SUCCESS
-rw-r--r-- 1 hadoop supergroup 77 2025-03-26 20:56 /user/hadoop/data/baitap-output/part-00000
hadoop@ngghiahuu-VirtualBox:~/code$ hdfs dfs -cat /user/hadoop/data/baitap-output/part-00000
04-Dec-2005 1051
04-Dec-2005_error 311
05-Dec-2005 949
05-Dec-2005_error 284
```

Hình 2.2: Giao diện hiện kết quả trên Terminal



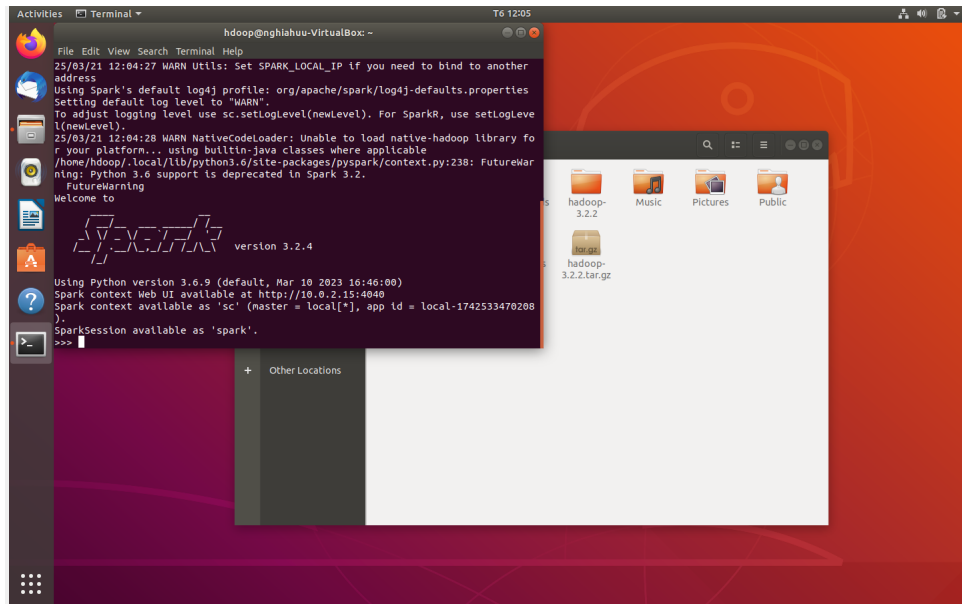
Hình 2.3: Giao diện hiển thị kết quả trên Hadoop UI

2.4 Cài đặt và triển khai ứng dụng trên Spark

2.4.1 Cài đặt Spark

Quá trình cài đặt Hadoop bao gồm các bước sau:

1. Cài đặt PIP
2. Cài đặt PySpark với PIP



Hình 2.4: Giao diện cài đặt PySpark

2.4.2 Lập trình ứng dụng

```
import os
import shutil
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, count, regexp_extract

# Khởi tạo SparkSession
spark = SparkSession.builder.appName("LogAnalysis").getOrCreate()

# Đọc file log
log_file = "/home/hdoop/Apache/Apache_2k.log.txt" # Đổi tên file nếu cần
df = spark.read.text(log_file)

# Trích xuất ngày từ dòng log
df = df.withColumn("date", regexp_extract(col("value"), r'\[(\w{3} \w{3} \d{2})', 1))

# Nhóm theo ngày và đếm tổng số dòng nhật ký
log_count = df.groupBy("date").agg(count("*").alias("total_logs"))

# Trích xuất loại nhật ký (notice, error, warning, v.v.)
df = df.withColumn("log_type", regexp_extract(col("value"), r'\[(\w+)\]', 1))

# Lọc các dòng có chứa "error"
error_count = df.filter(col("log_type") == "error").groupBy("date").agg(count("*").alias("error_lo"))

# Hợp nhất hai bảng bằng phép JOIN trên cột "date"
result_df = log_count.join(error_count, on="date", how="left").fillna(0)

# Thư mục lưu kết quả
```

```

output_dir = "/home/hadoop/log_analysis_result"

# Lưu kết quả vào thư mục (Spark sẽ tạo file part-00000)
result_df.coalesce(1).write.csv(output_dir, header=True, mode="overwrite")

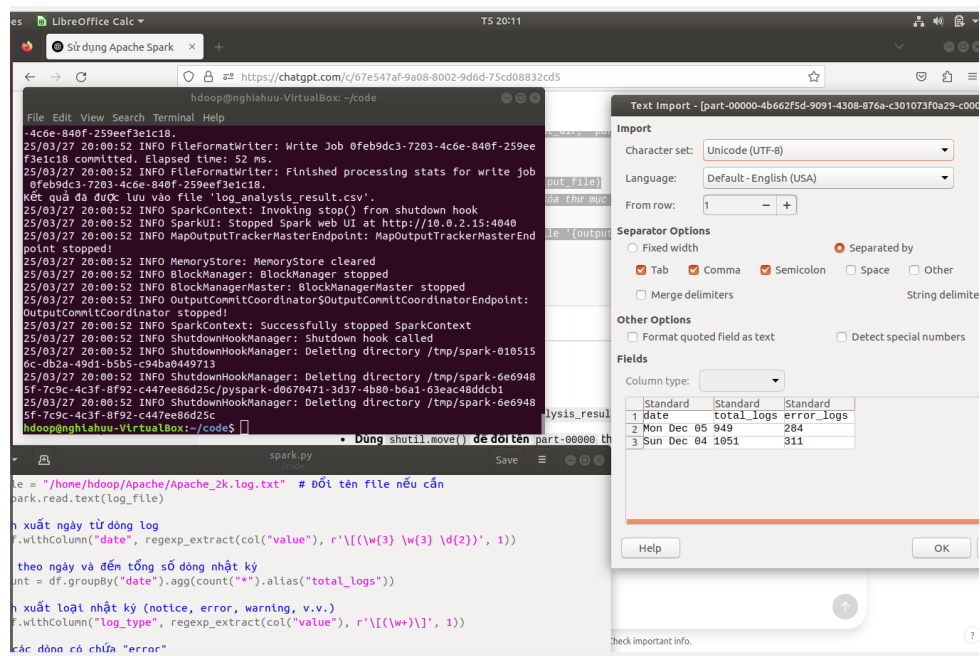
# Đổi tên file kết quả
output_file = "log_analysis_result.csv"
csv_file_path = os.path.join(output_dir, "part-00000")

if os.path.exists(csv_file_path):
    shutil.move(csv_file_path, output_file)
    shutil.rmtree(output_dir) # Xóa thư mục cũ sau khi đổi tên

print(f"Kết quả đã được lưu vào file '{output_file}'.")

```

2.4.3 Thực thi ứng dụng



Hình 2.5: Giao diện hiện kết quả

Chương 3

KẾT LUẬN

3.1 Đánh giá chung

Sau quá trình nghiên cứu và triển khai, đề tài đã đạt được những kết quả đáng khích lệ. Việc ứng dụng công nghệ vào thực tế đã mang lại nhiều lợi ích, giúp tối ưu hóa quy trình và nâng cao hiệu suất làm việc. Trong phần này, chúng tôi sẽ tổng kết những kết quả đạt được cũng như một số hạn chế còn tồn tại.

3.1.1 Những kết quả đạt được

Trong quá trình thực hiện, đề tài đã đạt được một số kết quả quan trọng như sau:

- Xây dựng được hệ thống đáp ứng các yêu cầu đề ra ban đầu.
- Ứng dụng công nghệ tiên tiến vào quá trình xử lý, giúp cải thiện hiệu suất.
- Hệ thống hoạt động ổn định, đáp ứng tốt các bài kiểm thử.
- Cung cấp giao diện thân thiện với người dùng, giúp dễ dàng thao tác và sử dụng.
- Tài liệu hướng dẫn chi tiết, giúp người dùng dễ dàng triển khai và bảo trì hệ thống.

3.1.2 Một số hạn chế

Bên cạnh những kết quả đã đạt được, vẫn còn một số hạn chế cần khắc phục trong tương lai:

- Hệ thống chưa được tối ưu hóa hoàn toàn về hiệu suất khi xử lý lượng dữ liệu lớn.
- Một số tính năng chưa hoàn thiện do giới hạn về thời gian và tài nguyên.
- Việc triển khai thực tế có thể gặp một số khó khăn do môi trường hệ thống khác nhau.
- Chưa có cơ chế mở rộng linh hoạt để tích hợp thêm các tính năng mới.

3.2 Hướng phát triển

Để hoàn thiện và nâng cao chất lượng hệ thống, trong tương lai có thể thực hiện các hướng phát triển sau:

- Tối ưu hóa thuật toán và kiến trúc hệ thống để cải thiện hiệu suất xử lý.
 - Bổ sung thêm các tính năng nâng cao nhằm đáp ứng tốt hơn nhu cầu thực tế.
 - Tích hợp với các hệ thống khác để mở rộng khả năng ứng dụng.
 - Nâng cấp giao diện người dùng để tăng tính tiện dụng và trải nghiệm tốt hơn.
 - Xây dựng cơ chế bảo mật tốt hơn nhằm đảm bảo an toàn dữ liệu.
- Những định hướng trên sẽ giúp hệ thống phát triển bền vững và đáp ứng tốt hơn các yêu cầu trong tương lai.

Tài liệu tham khảo

[1] Bài giảng xử lý dữ liệu lớn, Nguyễn Đình Hưng, ĐHNT .