SECP3133: HIGH PERFORMANCE DATA PROCESSING

SECTION 02

Project Proposal

Optimizing High-Performance Data Processing for Large-Scale Web Crawlers

Faculty of Computing

No	Nama	No Matrik
1	Soh Han Wei	A22EC0273
2	Loo Jia Chang	A22EC0074
3	Muhammad Nur Azhar Bin Mohd Yazid	A22EC0220
4	Nur Arini Fatihah Binti Mohd Sabir	A22EC0244

Table of Content

1.0 Introduction

1.1 Background

Today, data is growing and expanding at an alarming rate, especially on the Internet. Millions of websites currently provide valuable information across different fields. E-commerce websites such as Lazada are one of the most data-rich platforms, providing a variety of product lists, prices, reviews and other important details. Crawling or extracting these data on a large scale is very important for various commercial purposes, including price comparison, market analysis and consumer sentiment analysis. However, large-scale web crawling brings great challenges, such as managing server load, ensuring the efficiency of crawling and abiding by moral principles.

In this project, we aim to optimize the web crawling process for large-scale data extraction by using high performance computing (HPC) technology. The main focus in this project is to efficiently crawl websites and extract structured data while overcoming performance bottlenecks and moral problems.

1.2 Objective

To design and implement an optimized high-performance web crawler that can extract large-scale structured data from Lazada efficiently without overloading the server.

To analyze and evaluate the performance of different data extraction frameworks (Scrapy, BeautifulSoup, Selenium, Request-HTML) in terms of speed, scalability, and resource consumption for real-time and bulk crawling tasks.

1.3 Target Web - Lazada

For our project, we've chosen Lazada Malaysia as the primary website for data extraction. Lazada dominates Malaysian markets via its extensive catalog of available listings covering various categories

such as electronics, fashion, home appliances, and beauty products. The platform stands out to users because its streamlined product pages enhance bulk data extraction processes.

One of the main reasons we picked Lazada is due to the features it offers that are relevant to our project. Each product listing on Lazada usually includes detailed information such as product id, product name, current and original pricing, discount percentages, seller names and ratings, stock status, shipping options, and average rating. These fields are exactly the kind of data we aim to collect, as they are useful not just for testing our crawler's performance but also for practical data analysis tasks later on.

The product data through Lazada becomes more reliable because the platform features both official stores and verified sellers. User-generated reviews at Lazada operate at a high pace because a large number of customers provide ratings and feedback that prove useful for future analysis projects.

The platform lets users control their product crawl by searching and filtering to maintain focus on targeted data while avoiding server congestion. Our implementation of crawling ethics becomes possible through these system controls that allow us to follow both page crawl delays and reduce wasteful page requests.

Overall, Lazada provides a rich and structured data environment that aligns well with our technical goals.The platform offers genuine web data processing difficulties and consistent structured information that enables performance assessment of our high-speed crawler system.

1.4 Data To Be Extracted

1. Core Product Data

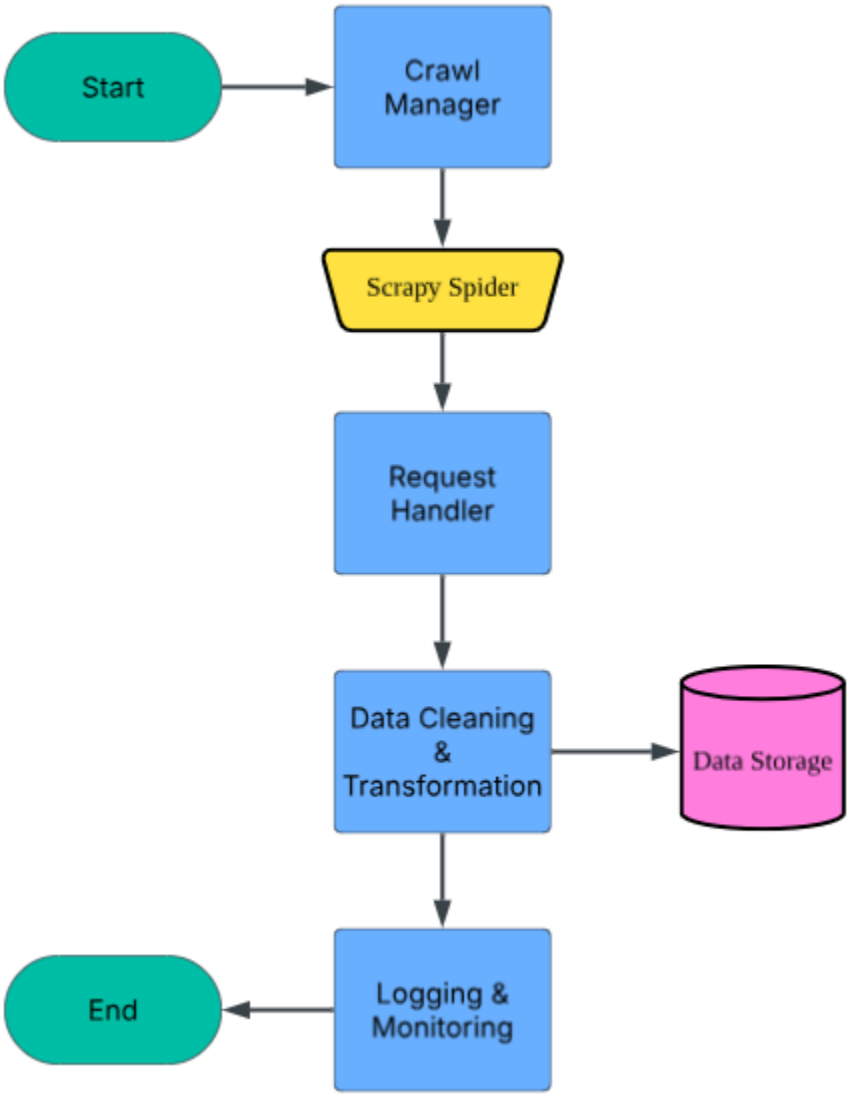
To make sure we gather over 100,000 structured data points without overwhelming the site with too much crawling, we'll zero in on the most valuable fields from product listings, reviews, and seller profiles.

We've crafted a streamlined extraction plan aimed at collecting data on 10,000 to 20,000 products, focusing on 10 to 15 key fields for each. We'll also include additional seller and review information to help us scale effectively.

Fields to Extract (Per Product)		
Field	**Example**	**Data Type**
product_id	"123456789"	String
product_name	"Xiaomi Redmi Note 12"	String
category	"Mobile Phones"	String
brand	"Xiaomi"	String
current_price	"RM 799.00"	String/Float
original_price	"RM 999.00"	String/Float
discount_percentage	"20%"	String
stock_status	"In Stock"	String
seller_name	"Xiaomi Official Store"	String
seller_rating	"4.9/5"	String/Float
total_reviews	"2,540"	Int
avg_rating	"4.7"	Float
shipping_info	"Free Shipping"	String
estimated_delivery	"2-4 days"	String
warranty	"1 Year"	String
product_specs	{"RAM": "8GB", "Storage": "128GB", "Battery": "5000mAh"}	JSON

2.0 System Design & Architecture

2.1 Crawler's Architecture



Description of Components

1. Crawl Manager
    a. Coordinating the entire crawling process, manages flow of tasks, schedules URLs to be crawled, and controls request rate.
    b. Ensures respect for rate limits and robots.txt

2. Crawling Tools - Scrapy Spider
    a. Use For: Crawling category pages with product cards
    b. Example: Get product URLs, prices, and short titles
    c. Why: Scrapy is an asynchronous framework and is best used for crawling pages in a fast and efficient manner

3. Request Handler
    a. Handles complex headers and user-agent rotation for Lazada.
    b. Manages cookies and session tokens (important for logged-in-only data)
    c. Captures anti-bot flags and implements fallback (retry or proxy)

4. Data Cleaning & Transformation Plan
    a. Standardize data types (e.g., convert prices to float)
    b. Remove formatting (e.g., commas from numbers, "RM" from prices)
    c. Handle missing/optional values gracefully
    d. Parse nested fields (like specs in JSON)
    e. Ensure output is clean and structured for storage/analysis

5. Data Storage
    a. CSV file (for batch processing offline)

6. Logging & Monitoring
    a. Tracks crawling health: number of pages hit, success/fail ratio, time taken per batch
    b. Alerts on potential bans, CAPTCHAs, or structure changes
    c. Example logs:

```
[INFO] Crawled 3,500 product pages - 2,890 success, 610
blocked, 12 retries
[WARN] Review section structure changed on 20 Apr
```

2.2 Libraries and Framework

Softwares / Tools	Usage / Functionality
Google Docs	Documentation & Report
Lucidchart	Architecture Design
Google Colab	Coding Environment
Python	Data Crawling, Data Cleaning and Transformation, Data Visualization
Scrapy Spider	Web Scraping
.csv File	Data Storage

2.4 Role of each members

Member	Responsibility
Soh Han Wei (Group Leader)	● Coordinate weekly meetings & track milestones in Trello ● Seek lecturer approval and handle external communications ● Maintain master project timeline & risk register ● Resolve roadblocks and re-allocate resources when needed ● Perform final review of all GitHub pull-requests and deliverables
Loo Jia Chang	● Design crawler architecture ● Implement core crawling logic with robots.txt & rate-limiting ● Set up progressive data checkpoints (CSV / SQLite) ● Document crawler API for team reuse ● Collaborate on introducing threading / asyncio for speed-up
Muhammad Nur Azhar Bin Mohd Yazid	● Draft planning brief and documentation templates ● Write unit & integration tests for crawler and processing modules ● Validate dataset integrity after each run (duplicates, missing values) ● Compile Turnitin-ready report and design presentation slides ● Verify submission checklist (code, dataset, report, slides) and package for upload
Nur Arini Fatihah Binti Mohd Sabir	● Clean & transform raw data, ensuring ≥ 100, 000 records ● Apply at least two optimization techniques (asyncio, threading, Spark/Dask) ● Benchmark CPU, memory, and runtime pre- vs post-optimization

	● Integrate logging dashboard to display progress percentages ● Provide performance metrics and graphs for the final report

8.0 Reference