




InfoStacked

InfoGeeks

- **K Manohar**
- **M Vineeth**
- **Manthan Gupta**
- **Samarth Agarwal**
- **Pulkit Jain**

PROBLEM STATEMENT



- It is a convoluted process to ingest plethora of resources available on net.
- People spend most of their time searching for the same topic using various suffix keywords.
- Our project focuses on taking the input information and the required categories for which the search results are intended.

OBJECTIVES



- Our project focuses on taking the input information and the required categories for which the search results are intended.
- These queries would then be executed parallelly and the results would be displayed dynamically on the Web Application.
- The project aims to showcase the comparative study of the processing time in the serial and parallel execution of the application.

Methodology

- There are 4 modules: Github repos, YouTube videos, recent News articles and Tweets.
- The user needs to input the search keyword and the results would be fetched from the backend API and will be rendered on the screen.
- Instead of searching independently category wise, we have provided an interface to efficiently gather all the resources quickly.

Methodology (contd.)

GITHUB MODULE

The module takes the input query and the count of best results to be fetched.

NEWS ARTICLE MODULE

The news article module takes in the search query and uses selenium to scrape the articles and displays the links on screen.

TWITTER MODULE

Displays the most recent tweets for the particular search query. It also performs the sentiment analysis of the tweet and sends the URL using the Twitter API.

YOUTUBE MODULE

Returns the relevant youtube video links for the search query.

We use multi-processing to run these 4 modules in parallel and made use of a shared list to accumulate the results from all the above modules.

RESULTS



- Our project focuses on taking the input information and the required categories for which the search results are intended.
- These queries would then be executed parallelly and the results would be displayed dynamically on the Web Application.
- The project aims to showcase the comparative study of the processing time in the serial and parallel execution of the application.
- We are able to achieve 2x speed up.

CONCLUSION



- Through the help of our parallel interface user would be able to fetch the useful resources in just one click, thus saving their arduous efforts.
- Serial implementation takes a considerable amount of execution time.
- Using multiprocessing we are able to reduce the execution time considerably.
- The difference in execution time would exponentially increase if the number of search results is increased.

REFERENCES

- <https://medium.com/velotio-perspectives/web-scrapingintroduction-best-practices-caveats-9cbf4acc8d0f>
- <https://medium.com/prowebscraper/5-best-javascript-webscraping-libraries-and-tools-71f2459fcfd8>
- <https://www.slideshare.net/vimalsura/parallel-and-distributedinformation-retrieval-system>
- https://www.csee.umbc.edu/~nicholas/676/mir2edSlides/slides_cha p10.pdf
- https://www.cs.helsinki.fi/u/hahonen/irm07/lectures/irm07_12.pdf <https://ieeexplore.ieee.org/document/651488>
- https://link.springer.com/chapter/10.1007%2F978-1-4020-3005-5_7 <https://serpapi.com/>
- <https://developers.google.com/customsearch/v1/introduction>

THANK YOU!!