Milestone: Parallel Job Miner

Jason Saini, Joshua Byrd, Brendan Smith, Parker Waller

March 8, 2024

Abstract

The technology industry encompasses a diverse and continually evolving landscape, presenting a wealth of opportunities for computer science students on the cusp of graduation. For computer science students, the transition from academic study to professional practice is a critical phase of their careers. The job search process for students can be time-consuming and often inefficient, with relevant positions scattered across numerous platforms.

Our team decided to make it easier for students to find job applications online with a parallel "job miner", which will consolidate job application data from multiple popular job sites for computer science: LinkedIn, GitHub/Simplify, USAJobs (for those interested in government positions), Monster, and Indeed. We accomplished this by using a process called "web scraping" to gather job opportunity data from the web. In other words, our program gets necessary information about jobs from the sites listed above and provides that information to our users in the form of an xlsx (Excel) file. With this Excel file, users will be able to adopt this strategy of tracking positions they are interested in.

1 Introduction

Transitioning from studying in college to the ever-changing workforce requires finding jobs and internships within one's field. This process often involves sending many applications and receiving mostly rejections. To make the process easier for students, particularly those majoring in Computer Science and Software Engineering, we have created a web scraping application.

The project is open source, and therefore free, to all that would like to use it. Being open source enables all those who wish to contribute and expand/improve upon it to do so. To get started, users simply need to follow the download and setup instructions on the GitHub page, allowing them to run it as needed. We were able to achieve all of this through using pre-existing Application Programming Interfaces (APIs), creating our own web scraping functions, and storing collected info into a DataFrame from Python's Pandas library.

2 Problem Statement

Students encounter difficulty when applying to many different positions or companies trying to find relevant opportunities across various websites. After a website with relevant listings and the desired opportunities, they then must go into each card to read the details and click the actual application link if they are interested. Navigating through a multitude of cards on multiple sites for jobs that may not be relevant or applicable to you can quickly become overwhelming.

To help combat this, our project stores much of the needed information into a single file that is easy to read and contains the application links directly to all jobs found. From this spreadsheet, applicants can track their progress of individual applications using color coding, as demonstrated in the aforementioned commonplace strategy for tracking applications.

3 Related Work

There are many resources available that scrape the sites mentioned above, along with many others, to collect job data. Where our project differs is that we are the only resource which collects job data from multiple sites at once, providing users with a comprehensive list of opportunities without needing much technical knowledge.

To achieve this, we use existing APIs to access content and data. APIs allow our project to interact with the platforms efficiently and collect the needed information in a straightforward manner. This approach allows us to streamline data retrieval and adapt to anti-scraping measures used by major job finding sites.

As noted above, many job finding companies implement constantly evolving anti-scraping features. These features include bot detection, dynamic loading, and including unique job ids into the URL. Having these measures in place makes basic web scraping challenging, but APIs help combat that challenge and enable our data collection from various sites.

Additionally, our program is easy to use and designed to be efficient. The user only needs to download the files, follow the quick setup on the GitHub page, and run the file with a title to receive 100+ job opportunities from 4 sites related to what they are looking for. The program is made to be efficient by utilizing threads to scrape all the websites at once, instead of one by one.

While existing resources do exist, our project offers a unique version by scraping and consolidating information from multiple sites at once.

4 Technique

Our approach to a multi-threaded web scraper was using BeautifulSoup for the manual web scraping, RapidAPI to get our APIs, and Python's multiprocessing library to run each scraper on a different thread.

BeautifulSoup is a popular Python library used for web scraping. We use it in conjunction with the Requests library, which sends a get request to a provided link and returns the contents. After getting the content, you pass that through to BeautifulSoup to parse the content it was given, we choose to use the HTML parser. Now that you have the HTML-content, you can use BeautifulSoup to then traverse the page via its html tags.

RapidAPI is a service that allows users to publish their own API and use others' published APIs, either free or for a cost. Most published APIs tend to have multiple price plans which allow them a certain number of requests.

Python's multiprocessing library enables a program to run processes on separate threads concurrently. It allows one to assign a 'Process' to a function with any arguments it may need. When the thread is started, it executes that function with the given arguments.

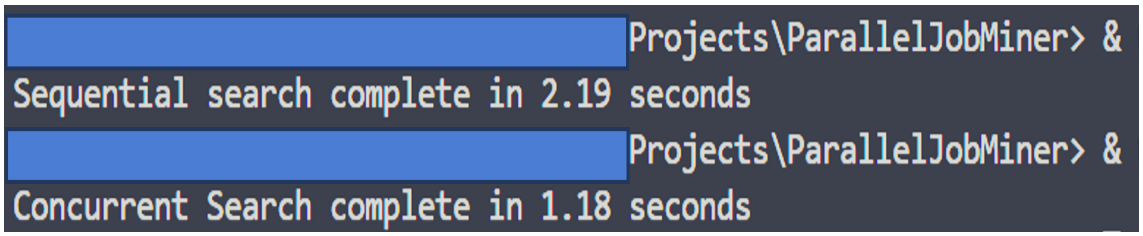
We use these in conjunction with each other to create our current application. BeautifulSoup allows us to find the needed content on the pages that we manually scrape, RapidAPI returns us the data of jobs that it gets, and multiprocessing allows us to run the scrapers concurrently instead of sequentially.

To ensure the scalability of our program we have a clean and organized file structure that is modular. For example, you can add a scraper to another site by simply adding a file for it and adding the function to the main file.

5 Evaluation

So far, our evaluation has shown that the sequential and concurrent methods of web scraping result in very similar execution times, with the sequential method having the faster runtime around 70% of the time. This, however, is mainly due to the need to finish adding scraping/querying functionalities for multiple job sites such as Monster, Glassdoor, and Indeed.

We expect that once we have more sources for jobs and more job listings in general, we should see the execution times start to deviate away from each other, with the concurrent method having a faster runtime.



After running our concurrent and sequential algorithms separately, we observed a nearly 86% decrease in runtime from end to end, from processing the user specified job title to generating the spreadsheet. While a noticeably small increase in the quantifiable runtime, a decrease of this magnitude on a much larger scale will be incredibly beneficial to those compiling job applications in larger quantities.

6 Limitations and Challenges

In the early stages of the project, we discovered that websites (i.e. LinkedIn) are weary against web scrapers for reasons including protecting intellectual property, preserving bandwidth and server resources, maintaining data accuracy and quality, protecting user privacy, preventing competitive advantage, avoiding legal complications, etc.

Our quick workaround was to use the official APIs, but of course these come with their own set of limitations. For example, the free LinkedIn API we are using allows 25 calls per month, which will not be serviceable for a full production launch. This is another challenge for scalability and maintenance, and a significant consideration when incorporating a data source into our parallel job scraper.

Another considerable challenge is accessing our pandas dataframe through threads, which requires a thread-safe extended class of the dataframe. A final challenge is finding ways to elevate the user experience from a heavily manual setup process to a more direct way of accessing our application.

7 Milestone Progress and Next Steps

We have reached a significant milestone in our project at this point, using a modular approach to thread different scrapers and API requests, we can quickly compile relevant job application information for prospective applicants. At the time of writing, these are the following tasks left at hand:

1. Finish scraping/querying remaining sites (Monster, Glassdoor, Indeed)
2. Investigate appending Excel data in multi-threaded way for potential performance improvements

3. Filter out duplicate positions from multiple sites (most likely using the application link to check for duplicate entries)
4. Navigate API token limits (LinkedIn only allows 25 queries a month)
5. Add legend to spreadsheet to allow color-coded status tracking for job applications
6. Adding more parameters to further customize job search
7. Reach goal: Implement a better UI or general UX improvements

8 Conclusion

In conclusion, our job miner project represents a significant effort towards easing the job search process for computer science students and professionals. By consolidating job application data from multiple popular job sites into a single Excel file, our tool empowers users to efficiently track and manage their job applications.

Throughout the development process, we have encountered various challenges and limitations, including navigating anti-scraping measures, managing API token limits, and enhancing the user experience. However, through our innovative approach and collaborative efforts, we have made substantial progress towards overcoming these obstacles.

Moving forward, our focus will be on refining the project, addressing remaining tasks, and incorporating user feedback to enhance usability and functionality. We are committed to finishing the scraping/querying functionalities for remaining sites, optimizing performance through multi-threading, implementing duplicate position filtering, and improving the user interface.

Ultimately, our goal is to provide a valuable resource for the computer science community, facilitating seamless access to job opportunities and empowering individuals in their career pursuits. We are excited about the future of the Parallel Job Miner project and look forward to its continued growth and impact.

9	USAJobs	Civil Engineer	U.S. Army Corps of Engineers	https://www.usajobs.gov:443/GetJob/ViewDetails/753917600
10	USAJobs	Interdisciplinary Engineer	Defense Contract Management Agency	https://www.usajobs.gov:443/GetJob/ViewDetails/779781900
11	USAJobs	Interdisciplinary Engineer	Defense Contract Management Agency	https://www.usajobs.gov:443/GetJob/ViewDetails/780704800
12	USAJobs	Computer Engineer (Cybersecurity) (Direct Hire)	Bureau of Industry and Security	https://www.usajobs.gov:443/GetJob/ViewDetails/779987100
13	USAJobs	TELECOMMUNICATIONS SPECIALIST	Air National Guard	https://www.usajobs.gov:443/GetJob/ViewDetails/778828100
14	USAJobs	SERVICE CONTRACT MONITOR	Department of the Air Force - Agency Wide	https://www.usajobs.gov:443/GetJob/ViewDetails/779252500
15	USAJobs	Civil Engineer (Hydraulics) - DIRECT HIRE	Federal Highway Administration	https://www.usajobs.gov:443/GetJob/ViewDetails/750738000
16	USAJobs	Engineer	Air Force Materiel Command	https://www.usajobs.gov:443/GetJob/ViewDetails/752256200
17	USAJobs	Computer Engineer	Forest Service	https://www.usajobs.gov:443/GetJob/ViewDetails/763513500
18	USAJobs	Civil/Agricultural Engineer (Design Engineer)	Natural Resources Conservation Service	https://www.usajobs.gov:443/GetJob/ViewDetails/779733700
19	USAJobs	Senior Electronics Engineer	Nuclear Regulatory Commission	https://www.usajobs.gov:443/GetJob/ViewDetails/779336800
20	USAJobs	INTERDISCIPLINARY ENGINEER/SCIENTIST	Naval Air Systems Command	https://www.usajobs.gov:443/GetJob/ViewDetails/764689900
21	USAJobs	INTERDISCIPLINARY ENGINEER/SCIENTIST	Naval Air Systems Command	https://www.usajobs.gov:443/GetJob/ViewDetails/760761400
22	USAJobs	CIVIL ENGINEER	Air Force Civilian Career Training	https://www.usajobs.gov:443/GetJob/ViewDetails/752235400
23	USAJobs	General Engineer	Office of Naval Research	https://www.usajobs.gov:443/GetJob/ViewDetails/766089500
24	USAJobs	Interdisciplinary - Cost Engineer	U.S. Army Corps of Engineers	https://www.usajobs.gov:443/GetJob/ViewDetails/765473200
25	USAJobs	Interdisciplinary	United States Army Futures Command	https://www.usajobs.gov:443/GetJob/ViewDetails/779548400
26	Github	Software Engineer – Intern - Summer 2024	Visa	https://jobs.smartrecruiters.com/Visa/743999972605824?utm_source=Simplify&ref=Simplify
27	Github	Software Engineer – Intern	Snackpass	https://boards.greenhouse.io/snackpasspt/jobs/4658696004?utm_source=Simplify&ref=Simplify
28	Github	Remote Backend Engineer Intern	Autodesk	https://autodesk.wd1.myworkdayjobs.com/Ext/job/Arizona-USA---Remote/Remote-Backend-Engine
29	Github	Software Engineer Intern	Appian	https://boards.greenhouse.io/appian/jobs/5785341?utm_source=Simplify&ref=Simplify
30	Github	Full Stack Engineer Intern	WeaveGrid	https://www.weavegrid.com/careers/job-openings?gh_jid=5118006004&utm_source=Simplify&ref=
31	Github	Software Engineer – Intern - Summer 2024	Visa	https://jobs.smartrecruiters.com/Visa/743999971543963?utm_source=Simplify&ref=Simplify
32	Github	Product Manager Intern - Data Science	Unite Us	https://boards.greenhouse.io/uniteus/jobs/7274221002?utm_source=Simplify&ref=Simplify
33	Github	Undergraduate Engineering Internship	Truveta	https://boards.greenhouse.io/truveta/jobs/5069430004?utm_source=Simplify&ref=Simplify
34	Github	Data Scientist Summer Intern - Undergraduate	Proofpoint	https://proofpoint.wd5.myworkdayjobs.com/en-US/ProofpointCareers/job/Draper-UT/Data-Scientis
35	Github	Intern – Software Developer - Summer 2024	Plexus	https://plexus.wd5.myworkdayjobs.com/en-US/Plexus_Careers/job/Neenah-WI/Intern---Software-D
36	Github	Software Engineer Intern - AI	Modernizing Medicine	https://www.modmed.com/company/careers/?gh_jid=7261630002&utm_source=Simplify&ref=Sim
37	Github	Data Engineering Intern	Jackpocket	https://boards.greenhouse.io/jackpocket/jobs/7268024002?utm_source=Simplify&ref=Simplify
38	Github	AI – Data Engineering Intern	Curai	https://jobs.lever.co/curai/9f669138-e676-4a70-b10f-51e4e72a33e2/apply?utm_source=Simplify&
39	Github	Software Engineering Intern - Summer 2024	CACI	https://caci.wd1.myworkdayjobs.com/External/job/US-VA-Sterling/Software-Engineering-Intern---Su
40	Github	Software Engineer – Intern - Summer 2024	Visa	https://jobs.smartrecruiters.com/Visa/743999971543683?utm_source=Simplify&ref=Simplify