Distributed Web Crawler

Samyak Shah, Uzair Islam, Nimisha Mathew

Dec 02, 2025

CONTENTS:

Add your content using `reStructuredText` syntax. See the reStructuredText documentation for details.

CHAPTER

ONE

ORCHESTRATOR

CSCI-651 Project: Distributed Web Crawler

Implementing a Distributed Web Crawler to crawl different websites, download all related files, calculate MD5 hash and send it back to the server. Uses Docker Container to host the server and MongoDB.

app.**connect_db**()

Method to lazily connect to DB, only 1 connection

Returns
DB connection to the collection

app.**get_urls**(*worker_id='worker0'*)

Method to distribute URLs to workers based on the worker ids.

Parameters
worker_id – ID of the worker requesting the URLs to crawl.

Returns
List of URLs for current worker to crawl.

app.**home**()

Basic method to check if server is up during healthcheck

Returns
JSON response indicating server is live.

app.**post_results**(*data=Body(PydanticUndefined)*)

Method to receive results from worker and post it into MongoDB.

Parameters
data – JSON dictionary of {url, file, md5, status} of all downloaded files received from workers.

Returns
None

CHAPTER

TWO

WORKER

CSCI-651 Project: Distributed Web Crawler

Implementing a Distributed Web Crawler to crawl different websites, download all related files, calculate MD5 hash and send it back to the server. Uses Docker Containers to host workers.

worker.**compute_md5**(*file_path*)

Method to compute MD5 of the downloaded file

Parameters
file_path – Path of downloaded file to calculate its MD5 hash

Returns
MD5 Hash of the file

worker.**crawl_arxiv_list_page**(*url*, *save_dir='downloads'*)

Method to crawl the arXiv website and download all PDFs

Parameters

- **url** – URL of the website to crawl
- **save_dir** – Directory name to download files and store them

Returns
JSON dictionary of {url, file, md5, status} of all downloaded files

worker.**crawl_mit_list_page**(*url*, *save_dir='downloads'*)

Method to crawl the MIT website and download all PDFs

Parameters

- **url** – URL of the website to crawl
- **save_dir** – Directory name to download files and store them

Returns
JSON dictionary of {url, file, md5, status} of all downloaded files

worker.**crawl_quanta_page**(*url*, *save_dir='downloads'*)

Method to crawl the Quanta website and download all articles :param url: URL of the website to crawl :param save_dir: Directory name to download files and store them :return: JSON dictionary of {url, file, md5, status} of all downloaded files

worker.**download_article_content**(*url*, *save_dir*)

Method to download the file and save it

Parameters

- **url** – URL of file to be downloaded

- **save_dir** – Directory name where to store it

Returns
Full file path where file has been stored

worker.**download_file**(*url*, *save_dir*)

Method to download the file and save it

Parameters

- **url** – URL of file to be downloaded
- **save_dir** – Directory name where to store it

Returns
Full file path where file has been stored

worker.**main**(*try_counter=0*)

Main method to trigger crawling based on worker ids. Retries 3 times on failures.

Parameters
try_counter – Current attempt of running main function

Returns
None

PYTHON MODULE INDEX