Analytical Approach In Making Technical Stack Selection

Apoorv Yadav
Binghamton University
Binghamton, USA
ayadav7@binghamton.edu

Akash Rasal
Binghamton University
Binghamton, USA
arasal2@binghamton.edu

Abstract

Social Media is an integral part of our lives. It is omnipresent and influences our day-to-day decisions. We can collect a lot of this data and perform analysis on it. Using this data, we can gain insights on various situations. Web repository platforms are easy to use and manage open-source projects. Posting issues and getting them resolved is a part and parcel of open-source projects. On the other hand, for development related issues, support, any interesting findings and geeky discussions, users use social media platforms. We intend to collect data from two sources, Reddit[1] and GitHub[2]. Reddit is a community based social-media platform and GitHub is a popular repository and version control platform. The project collects Reddit post and comments for 7 subreddits. Also, GitHub issues and their comments are stored in database. These data points will be used to see how fast issues are resolved and how helpful is the GitHub community. This analysis can be further extended to facilitate decision making while choosing any technology stack.

ACM Reference Format:
Apoorv Yadav and Akash Rasal. 2023. Analytical Approach In Making Technical Stack Selection. In *Proceedings of Project Proposal (CS 515 '23).* BU, Binghamton, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

With user adoption of different frameworks, new technologies and languages, online communities are increasing. Online communities help resolve issues and have healthy discussions. Subreddits are very important when resolving an issue while working on a technology/developing an application. Users post their application development issues on subreddits where other users from the globe help resolve it. They often ask follow-up questions to get a clear understanding of the issue and try to provide solution. The comments can be upvoted, downvoted and can be shared. The development community enjoy helping each other over social media platforms. For open-source projects, a repository of the code is maintained and users could run the code and contribute by posting issues, resolving the issues and keep the project building. Many open-source projects are developed together using version management platform GitHub. In GitHub, an issue can be raised in a repository using the issue section. Comments can be added on issues, reactions can be given to a comment, and issues can be closed. An open-source community can be analyzed for its engagement, comments on issues and closure of issues.

2 Data Source

For gathering data, we have selected few GitHub repositories and their Reddit counterparts. The selection criteria that we used were primarily the popularity of the technology, contributions on the GitHub repository and active community on Reddit.

Following are the repositories on GitHub:

- `TypeScript` [14]
- `nixpkgs` [8]
- `rust` [10]
- `flutter` [1]
- `go` [3]
- `kubernetes` [5]
- `swift` [12]

Following are the most popular subreddits (in terms of size) on Reddit:

- `r/typescript` [15]
- `r/NixOS` [7]
- `r/rust` [11]
- `r/FlutterDev` [2]
- `r/golang` [4]
- `r/kubernetes` [6]
- `r/swift` [13]

3 Data Collection

The project is implemented using Python and Postgres [9] database is used to store the data.

3.1 Reddit

The design of the system which collects data from reddit is made modular to accommodate new subreddits. We import required libraries. Create a database connection and pass it as dependency to the RedditCrawler object. The crawler

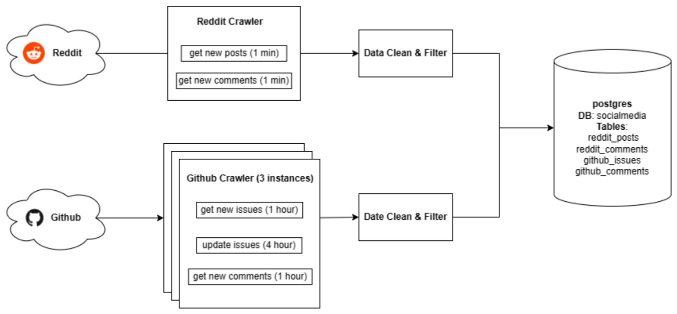

Figure 1. System Design

reads the config object, listing all the subreddits. The config itself can be extended for multiple reddit crawlers. The crawler then runs for fetching new posts and new comments for all the subreddits listed. Once records are fetched, the data collection system selects and perform necessary casting. Further, we insert the records into the postgres database until we find the record existing, at this point, it means we have already collected the records and hence stops the function. We were able to use this logic for early stopping due to the specifications of the response from reddit which states that the records are sorted with time.

The reddit data collection system is scheduled to run every one minute. We estimate collecting about 100 posts per day and 1250 comments.

Total Expected Posts while running the systems from Oct 20 to Dec 15:

$$\begin{aligned} \text{Total} &= \text{Number of Posts/Day} \times \text{Number of Days} \\ &= 100 \text{ posts/day} \times 55 \text{ days} \\ &= 5,500 \text{ posts} \end{aligned} \tag{1}$$

Total Expected Comments while running the system from Oct 20 to Dec 15:

$$\begin{aligned} \text{Total} &= \text{Number of Comments/Day} \times \text{Number of Days} \\ &= 1250 \text{ comments/day} \times 55 \text{ days} \\ &= 68,750 \text{ comments} \end{aligned} \tag{2}$$

3.2 Github

There are three tasks done for GitHub data collection system.

3.2.1 Get new Issues. This task collects new issues from the GitHub repository. As done for reddit, this task is designed to accommodate new GitHub repos form the configuration file. After importing required libraries, we read the config file to get repositories and tokens. A connection is made to the database and using API calls, issues are fetched. New issues from the response are inserted into the database if they do not exist in the database. We estimate to collect 400 issues daily.

Total Expected Issues to be collected from Oct 20 to Dec 15:

$$\begin{aligned} \text{Total} &= \text{Number of Issues/Day} \times \text{Number of Days} \\ &= 400 \text{ Issues/day} \times 55 \text{ days} \\ &= 22,000 \text{ Issues} \end{aligned} \tag{3}$$

3.2.2 Update issue state. To update issue state, all the issues are fetched from the database whose state is open. Then on every issue_number from the database, we send API request to get details of the issue. If the issue is closed, an update is made to the database with the new state of the issue.

3.2.3 Get new comments. Like GitHub issues, fetching comments on issues is also made modular. We collect comments on an all open issue until it is 30 days old. As we estimate to have 9000 (= 300 issues/day * 30 days) issues open in worst case, and with 5000 requests/hour per access token, we have made three groups of GitHub repos each summing to have almost same number of open issues at a time. To accommodate three groups and the rate limit, three access tokens are created from three different accounts. According to the input from command line, the program reads the token and repos form the configuration file. For the current repo in consideration, all open issues which may be 30 days old, are fetched from the github_issues table. For each issue_number, all comments are fetched using the API with a filter of time whose value is also taken from the configuration file. All the new comments are inserted in the github_comments table. After all the comments for all the issues are fetched and inserted, the current time is saved in the configuration file to be used for next run.

We are collecting 1000 comments daily on average. Total Expected comments to be collected from Oct 20 to Dec 15:

$$\begin{aligned} \text{Total} &= \text{Number of Comments/Day} \times \text{Number of Days} \\ &= 1000 \text{ Comments/day} \times 55 \text{ days} \\ &= 55,000 \text{ Comments} \end{aligned} \tag{4}$$

4 Total Data Collection

Overall, we expect around 150K data records till the end of the semester from the seven subreddits and github repos listed in the original proposal.

Separate tables are created: reddit_posts, reddit_comments, github_issues and github_comments. Reddit_comments are linked to reddit_posts using post_name. On the other hand, Github_comments are linked to github_issues using issue_number.

Figure 2 and 3 shows data collected for Reddit posts and their comments and GitHub issues and their comments respectively over the period of 10 days.

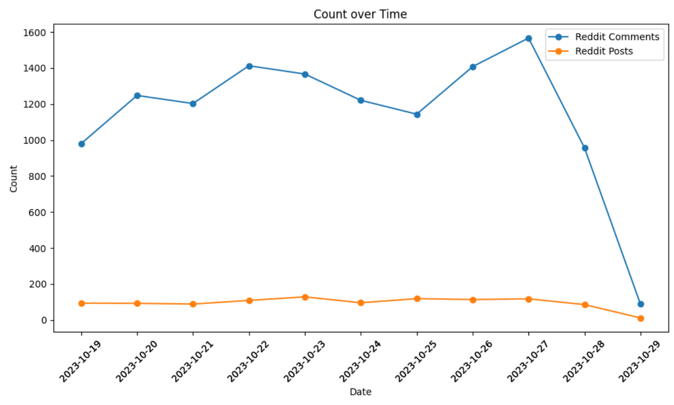

Figure 2. Data for GitHub issue and comments as of 29 Oct 2023 2:45 am UTC

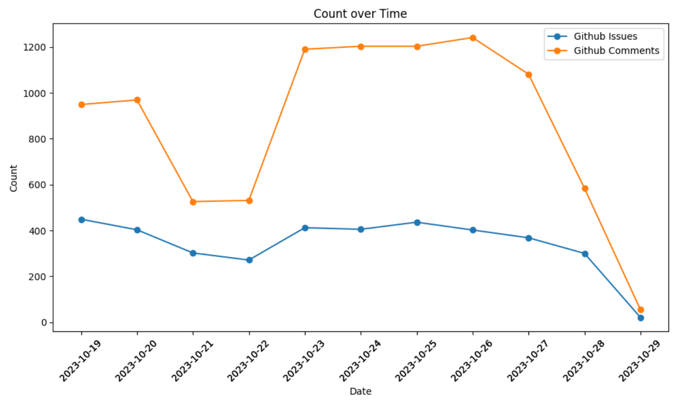

Figure 3. Data for GitHub issue and comments as of 29 Oct 2023 2:45 am UTC

5 Changes from proposal

In the project 1 proposal, we mentioned a slightly different method to collect reddit comments. We intended to fetch comments from individual posts using /api/subreddit/comments/article. However, that posed us with a new challenge to how frequently to fetch individual posts to not miss comments which are removed or deleted. We updated this approached to professor's suggestion to use /api/subreddit/comments/ which gives us new comments from the entire subreddit. Another addition to our initial decision was to mark issues in github as closed. This was essential to ensure we do not violate the api rate limits of github (5000 Queries per hour).

We also observed changes in the amount of records we are gathering. We stated the following in the proposal for our sources.

- Reddit: 4900 rows per week
- Github: 735 rows per week

After collecting data from oct 20, our observations are as

- Reddit: 9450 rows per week
- Github: 9800 rows per week

Hence, we are collecting more data than estimated in the proposal. We have listed the expected total data in section 3.

6 Challenges

Another problem that we could have potentially faced was reaching api limits for github api. Since we are only allowed 5000 Queries per hour, we can only check for 2500 open issues for closing status and 2500 issues for comments. This was fairly low number from what we expect for open issues. Therefore, we had to use three different configurations to fetch data for all the subreddits due to this limit. As the number of issues reach towards 5000, we might risk hitting the limit and therefore decided to use three separate github tokens to collect data. This allows us to query the api 15000 times per hour which is far higher than what we expect it in worst case.

7 Preliminary Exploration

We performed some initial exploratory data analysis on collected data so far. We observed a similar trend in posts vs comments for both the sources. A dip in the count can be seen for weekends (Oct 21, Oct 22, Oct 28 and Oct 29) for github issues and github comments. This behaviour is not observed in the reddit. With respect to activity, NixOS is most active repository amongst the seven. In reddit, The most records belong to r/rust followed by r/golang.

8 Analysis

For this project, we have few prospects of how we can analyze this data and present it to users. GitHub data can be utilized to get insights regarding number of issues per week, average time for an issue to be resolved, comments per issues. An issue can be tracked for quickest resolution. Also, issues can be used to see how naïve or stable the technology is. Similarly, for reddit, we will analyze the time frame for a post to have an active discussion, i.e., time taken for the first interaction, average number interactions, unique users, user engagement.

9 Future Work

We will be performing sentiment analysis on comments to monitor how positive and uplifting a community is towards a post in the next project. This influences new users to adapt a tool more easily and hence can be a good measure for us to explore.

References

[1] 2023. Flutter GitHub. https://github.com/flutter/flutter
[2] 2023. Flutter subreddit. https://www.reddit.com/r/FlutterDev/
[3] 2023. Go GitHub. https://github.com/golang/go
[4] 2023. Go subreddit. https://www.reddit.com/r/golang/

[5] 2023. Kubernetes GitHub. https://github.com/kubernetes/kubernetes
[6] 2023. Kubernetes subreddit. https://www.reddit.com/r/kubernetes/
[7] 2023. NixOS subreddit. https://www.reddit.com/r/NixOS/
[8] 2023. Nixpkgs GitHub. https://github.com/nixos/nixpkgs
[9] 2023. Postgres Database. https://www.postgresql.org/
[10] 2023. Rust GitHub. https://github.com/rust-lang/rust
[11] 2023. Rust subreddit. https://www.reddit.com/r/rust/
[12] 2023. Swift GitHub. https://github.com/apple/swift
[13] 2023. Swift subreddit. https://www.reddit.com/r/swift/
[14] 2023. TypeScript GitHub. https://github.com/microsoft/TypeScript
[15] 2023. TypeScript subreddit. https://www.reddit.com/r/typescript/