GitHub 2020

Digital Insight Report

X-lab 开放实验室

osse 开源社会工程研究院

开源社 kaiyuanshe

DaSE Data Science & Engineering

SHOPEN 上海开源信息技术协会

Content

Abstract 02

Introduction 03

1. Overview 06

2. Developer Analysis 07

2.1. The Level of Activity of Global Developers
2.2. Usage of GitHub Apps
2.3. Global log time distribution
2.4. Typical Work Hour Distribution of Open Source Software Developers
2.5. Time Zone Distribution of All GitHub Developers
2.6. Language Distribution for Developers

3. Project Analysis 13

3.1. Overall Analysis of All Projects on GitHub
3.2. Top 20 Most Active Projects
3.3. Ranking of the Activity Metric of China's Projects
3.4. Language Distribution for Projects
3.5. OpenGalaxy

4. Case Study 22

4.1. Case Study Analysis Method
4.2. CNCF Foundation Projects
4.3. Linux Foundation AI & Data Foundation Projects
4.4. Apache Software Foundation Big Data Projects Analysis
4.5. VS Code Case Study

5. Star of the month 32

5.1. Evaluation method of the monthly star
5.2. 2020 Monthly Open Source Star

6. Summary and Outlook 34

7. Acknowledgements 34

Description 1.1 GH Archive 06

Description 1.2 GitHub Event Log 06

Description 1.3 Statistical Methods for Active Repositories and Developers 06

Description 2.1 Developer Activity Metric 07

Description 2.2 The number of developers' active repositories 08

Description 2.3 GitHub Apps 08

Description 2.4 UTC standard time 10

Description 2.5 Statistical method of working time distribution 10

Description 2.6 Understanding method of working time distribution diagram 10

Description 2.7 Developer time zone estimation method 11

Description 2.8 Measurement of the language used by developers 12

Description 3.1 Project Activity Metric 13

Description 3.2 Chinese-initiated project and corporate project ownership 15

Description 3.3 Description of how to build OpenGalaxy 18

Description 4.1 Construction method of developer collaboration network 31

Abstract

Open source software has become a cornerstone of our digital society and is the culmination of human endeavor. Open collaboration has played an enormous role in promoting the development of human digital civilization. GitHub is the world's largest open source platform for collaboration with countless open source communities. The massive amount of developer behavior data can reflect an abundance of individual contribution patterns, group collaboration models, community health status indicators, ecological development trends, as well as business value.

The *GitHub 2020 Digital Insight Report* is an open source project that explores the status of and global trends in open source. The project was initiated by X-lab and jointly developed and implemented by several scientific research institutions and open source communities. The report begins with a general analysis of the global state of open source today. Subsequent sections include discussion of in-depth developer analysis, project analysis, case studies, and star projects of the month. The purpose of the report is to produce a worldwide open source ecosystem map, promote the open source social innovation, and foster the open source digital civilization.

Introduction

2020 was destined to be a standout year for everyone, and open source was no exception.

In January, the Wuhan2020 Open Source Community, an initiative for combating COVID-19 launched by developers and supported by X-lab, inspired us to make the world a better place through open collaboration. With one small step in self-organization, open collaboration affords us a glimpse into humanity's collective future.

In July, GitHub completed the Arctic Code Vault project, which archived 21TB of code data on 186 reels of film in the Svalbard archipelago, located halfway between Norway and the North Pole, where the open source codes will be preserved for at least 1,000 years.

In September, China's first open source software foundation, the Open Atom Open Source Foundation, was officially established. The Open Atom Open Source Foundation is a global nonprofit organization dedicated to the open source industry. It provides neutral intellectual property custody services for various open source projects, as well as strategic consulting, legal consulting, project operations, and brand marketing services.

In December, the Linux Foundation-sponsored open source community CHAOSS officially landed in Shanghai, China. It began to promote measurement systems, tools, and methodology of the health of open source projects and communities, which made the community governance within the open source ecosystem more meaningful. It can be implemented more effectively through big data and digital tools.

Throughout and despite the pandemic, open source has been growing fast and is developing even more rapidly today. From various data indicators, we found that open source shows a growing global trend: the number of GitHub logs reached **860** million in 2020, an increase of **43%** compared to 2019; the number of active code repositories reached **54.21** million, an increase of **36%** compared to 2019; the number of active developers reached **14.54** million, an increase of **22%** over 2019.

Serving as public infrastructure for innovation and entrepreneurship in the digital economy, open source eats into traditional business dogma with its **Openness**, **Peering**, **Sharing**, **Transparency**, **and Acting Globally**.

With more than 20 years of development, open source has evolved and generated digital infrastructures such as Linux, MySQL, Hadoop, Kubernetes, TensorFlow, React, VS Code, etc. The whole technology stack continues to grow upward, increasingly encompassing the entire software universe and forming an **open source ecosystem** as well as an open source civilization. Composed of all code, data, developer behavior, community norms, etc., the open source ecosystem is like a new species that is constantly evolving and growing, supporting the infrastructure of the entire digital world, and gradually changing the way humans collaborate and behave.

Open source is more akin to social innovation that can overcome various limitations and disadvantages of the traditional market, such as inefficient dissemination of information. For example, open source collaboration platforms have also iterated into a **massive data marketplace** where a large amount of transparent information will reside, beginning to have **automated digital tools** that enable decision making and collaboration. All of this will have a huge impact not only on organizations and their senior management teams, but also on all types of participants in the ecosystem, including developers, community maintainers, and consumers.

We think the free and open source software industry is over the conventional closed source software industry thanks to its inherent transparency, efficiency, massive data marketplace, and automated collaborative digital tools. The globally distributed collaboration, with its 24/7, uninterrupted coding relay, has enabled the software industry to advance at an unprecedented pace.

As the management guru Peter Drucker famously said, "**If you cannot measure it, you cannot manage it**," **and, consequently, you cannot improve it**. The software industry has yet to find a method that can effectively measure the productivity of software development. This holds true especially for the entire open source ecosystem. How individuals and communities are measured and how managers can use these metrics to make better decisions are still open questions. In our view, these are both challenges and opportunities. When carrying out the work of open source governance, it is difficult to circumvent the problem of measurement, but all GitHub event logs offer an excellent opportunity to solve the problem.

Measurement is also a double-edged sword. Given its strong guiding nature, it motivates you to pay attention to and improve the elements that can be measured. However, measurement may also cause you to ignore and worsen the elements that cannot be measured. How can you find the right metrics and use them wisely in the process of building large-scale open source communities and ecosystems around the world? We hope the present report brings you some insights.

Based on last year's GitHub 2019 Digital Annual Report, the main changes in this year's *GitHub 2020 Digital Insight Report* include:

- The iteration of the entire report was completed collaboratively in the form of an open source project, involving data, code, and text content;
- More comprehensive metrics and more scientific calculation methods are proposed;
- More specialized and rich methods for data visualization and insight are used;
- On the basis of activity metric defined last year, more attention is paid to information in the time dimension, diversity dimension, and collaborative network dimension in the present report;
- The concepts of OpenGalaxy and OpenQuadrant are put forward for the first time and implemented on the ground;
- An in-depth case study of developer collaboration network within specific projects is included;
- A Star Project of the Month that has received a lot of attention over a short period of time is reviewed.

This year, we changed the report title from GitHub Digital Annual Report to GitHub Digital Insight Report. One important reason is that we no longer merely list some statistics, but will also dig into patterns behind the data, and combine scientific explanations with expertise.

The main insights of this report are as follows:

- The global open source industry has developed greatly. The active behavior of the community, the number of developers, and the number of open source repositories have all increased significantly;
- The automation of open source software pipeline has been greatly improved, and diversified digital collaborative robots have begun to enter the mainstream;
- An activity model based on massive data can effectively and continuously reflect the overall status of developers and communities;
- The working hours of mainstream developers in GitHub communities show clear patterns and gradually overlap with developers' working hours in companies;
- Corporate open source has become the absolute mainstream, and '996 working hour open source projects' have begun to crop up;
- The Americas have the largest distribution of developers; Europe has the highest percentage of developers in a single time zone, while the number of Asian developers is still on the low end. Compared with other Asian countries, China boasts a higher level of open source activities;
- JavaScript and Python still occupy the first and second place, respectively, in the language rankings; HTML and CSS are more popular in the context of global developers, while TypeScript and Rust have significantly gained in prominence;
- Established companies such as Google and Microsoft remain major contributors to open source. China-headquartered e-commerce giant Alibaba ranks topmost in terms of activity metric, while PingCAP has recorded an impressive performance;
- For the first time, a more complete picture of GitHub open source projects is revealed through the OpenGalaxy, which we introduce in this report. An open source ecosystem has formed in the mainstream technology field. New communities are constantly emerging, while a very small number of projects are still isolated collaboration islands;
- Cloud Native Computing Foundation (CNCF), Linux Foundation (LF), Apache Software Foundation (ASF), as well as other foundations, are noted for their own strengths in the technical field. We also introduce the Open Source Quadrant, which can further distinguish different development stages and maturity levels of similar open source projects;
- Developer time zone distribution maps and developer collaboration networks have become effective means of reflecting the diversity and robustness of open source communities, and can better guide the open source community governance.

The complete *GitHub 2020 Digital Insight Report* is presented below.

1. Overview

The total number of GitHub global event logs was about 860 million in 2020, representing an increase of about 42.6% from 610 million in 2019, which was the fastest-growing year in the past five years. In this report, through project and developer behavior data, statistics show that the number of GitHub global active projects in 2020 was about 54.21 million, while the number of active developer accounts was about 14.54 million, marking an increase of 36.4% and 21.8%, respectively, over 2019.

Figure 1.1 An overview: the number of event logs, active repositories, and active developer accounts on GitHub from 2015 to 2020

Description 1.1: GH Archive

Open source software developers all over the world are contributing to millions of projects in parallel, writing code and documentation for projects, fixing and submitting bugs, and so on. GH Archive is a project used to record the public GitHub timeline (that is, a log of all events on GitHub in chronological order), archive and make it easy to access for further analysis.

Description 1.2: GitHub Event Log

GitHub provides more than 20 event types, ranging from new Commit and Fork events, to submitting new Pull Requests, Comments, and adding members to the project. These events are aggregated into hourly archives, which can be accessed using HTTP clients.

Description 1.3: Statistical Methods for Active Repositories and Developers

A repository is defined as active if it contains at least one event log during the period. Similarly, a developer is considered active if the developer has any repository containing event logs.

2. Developer Analysis

At the core of the open source world are developers. Following the spirit of the Apache maxim ‘Community Over Code’ promoted by The Apache Way, the developer community is crucial to open source vitality. The objective of this report is to present a comprehensive analysis of all GitHub developers in 2020 from multiple perspectives such as the activity level of developers, GitHub Apps usage, typical working hours, global developer time zone distribution, and developer language distribution.

2.1. The Level of Activity of Global Developers

Through analyzing the developer activity metric and the number of active repositories across all of GitHub, we obtain the distribution of the activity metric of global developers on GitHub and the distribution of the number of repositories a single developer actively worked on.

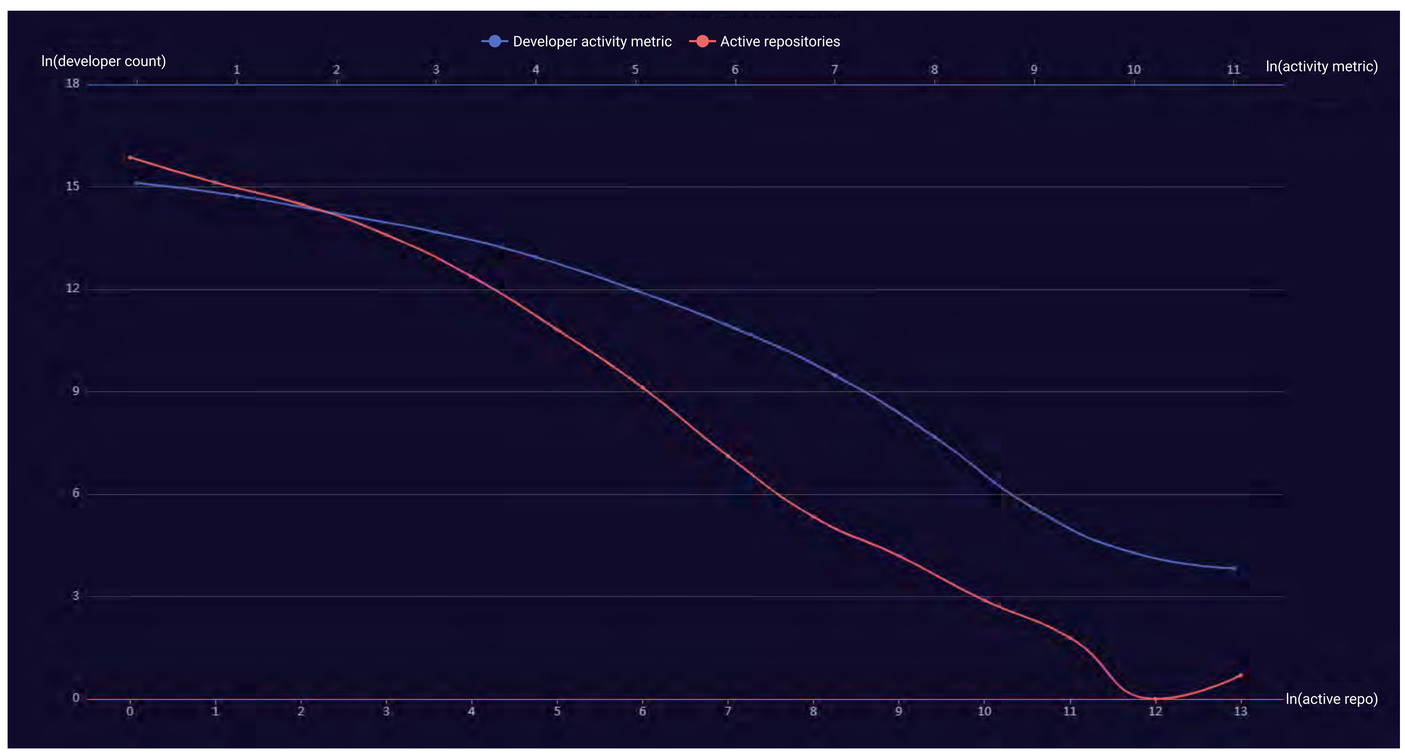

Figure 2.1 The distribution of developer activity metric and the number of active repositories

The figure above is drawn on a double logarithmic coordinate system. The abscissa at the top of the figure represents the activity metric of developers, the abscissa at the bottom represents the number of developers participating in the project, and the ordinate represents the number of developers. The axes are natural logarithms. It can be seen that the distribution of developer activity metric and the number of active repositories follow a power-law distribution. Statistics suggest that 5,445 developers’ activity metric is over 2,000, equivalent to less than 6 out of 10,000 Github developers. Most of the developers’ activity metric falls into the range of [0,500], accounting for 99.45% of the total number of Github

Description 2.1: Developer Activity Metric

Developer activity is a metric that reflects the level of activity of a specific GitHub account in a specific GitHub project over a period of time. Developer activity metric is determined by the account behavior data in the project. The behaviors detailed in this report include:

- Issue comment: Participating in a discussion on an issue is the most basic behavior, whereby each comment counts as one point.
- Open issue: Initiating an issue in a project, whether it be a discussion, a bug report or a question, is active to the project; each issue initiated counts as one point.
- Open pull request: Submit a PR for the project, indicating that the source code has been contributed to the project; each time a PR is initiated it counts as one point.
- Pull request review comment: To review and discuss the PR in the project, you need to have a considerable understanding of the project and greatly help the quality of the project source code. Each comment counts as one point.

  Note: The discussion of a specific line of code only through code review is recorded as a review, and the comment reply directly to the PR is recorded as an issue comment event.
- Pull request merged: If a PR is merged into the project, even a small change requires a deeper understanding of the project. It is a tangible contribution to expedite the project implementation progress. Each PR is merged one time. At the same time, the PR merge event increases the number of lines according to the code of the PR.
- Watch: The user's star item is counted once. Note: Star operations are recorded as Watch events in GitHub log events.
- Fork: The developer forks the project, which counts as one point.

developers, which suggests that most developers still have a low level of activity. At the end of the curve, the number of active repositories shows an increase, likely attributable to the huge number of active repositories for some unfiltered automated collaboration accounts, which far exceed the number of human developers; hence, the tail end of the curve is V-shaped.

In addition, we find 8 of the top 10 most active developers are GitHub Apps, and the other two are developer accounts for automated collaboration.

Table 2.1 Top 10 Developer Account Activity Level Across all of GitHub in 2020

#	actor login	activity metric	issue comment	open isuue	open pull	pull review comment	merge pull	star	fork
1	dependabot[bot]	36,082,423.2	3,062,227	0	17,914,723	0	1,311,348.0	0	0
2	dependabot-preview[bot]	10,169,281.1	2,097,036	38,354	3,547,903	0	1,214,518.7	0	0
3	pull[bot]	9,591,558.7	0	0	3,204,996	0	2,585,151.0	0	0
4	renovate[bot]	2,668,048.6	57,090	2,450	949,411	0	620,122.0	0	0
5	github-learning-lab[bot]	1,988,666.3	1,578,947	727,017	93,468	53,231	66,450.2	0	0
6	github-actions[bot]	984,674.2	632,314	60,492	128,594	39,412	72,411.8	0	0
7	direwolf-github	894,975.0	0	0	516,714	0	0	0	0
8	codecov[bot]	826,249.8	838,764	0	0	0	0	0	0
9	snyk-bot	654,743.4	0	0	346,908	0	34,277.0	0	23
10	sonarcloud[bot]	527,040.0	755,729	0	0	0	0	0	0

As shown in Table 2.1, the automated collaborative robots can serve many projects concurrently because they run on the server side, which results in an extremely high level of activity and a huge number of collaborative repositories. The above statistics of developer activity and active repositories exclude the collaboration behavior of GitHub Apps related accounts.

2.2 Usage of GitHub Apps

As shown in Table 2.1, the majority of the most active developer accounts across GitHub are GitHub Apps. This report analyzes relevant GitHub Apps data. Figure 2.2 shows the trend of both the number of active GitHub Apps (total active accounts) and the ratio of logs generated by GitHub Apps to all GitHub event logs (proportion of logs) from 2015 to 2020.

Figure 2.2 shows that GitHub Apps have developed rapidly in recent years since their launch in 2016. In terms of the proportion of logs, the total number had increased by 288% in 2019 compared to 2018, and it increased by 141% in 2020 compared to 2019 to as high as over 12%.

However, up until 2020, there were only 3,058 active GitHub Apps

The aforementioned seven kinds of behaviors are counted independently in the report model and are assigned different weights. According to expert experience, the weighted values are 1, 2, 3, 4, 2, 1, 2, namely:

$$\begin{aligned} A_{u\_d} = {} & C_{issue\_comment} + 2C_{open\_issue} \\ & + 3C_{open\_pr} + 4C_{review\_comment} \\ & + 2C_{pr\_merged} + C_{watch} \\ & + 2C_{fork} \end{aligned}$$

Among them, the count of pull request merged is determined by the piecewise function:

$$C_{pr\_merged} = \begin{cases} 0.8 + 0.002 \times loc & loc < 100 \\ 1 & 100 \le loc < 300 \\ 2.5 - 0.005 \times loc & 300 \le loc < 400 \\ 0.5 & loc \ge 400 \end{cases}$$

where *loc* represents the number of new lines of code.

According to the classic statistics of software metrology, a single code change is best within 200 lines. Code changes exceeding 400 lines will cause review difficulties. Therefore, the number of new lines of code and the weight index of a PR are associated through a piecewise function.

Description 2.2: The number of developers' active repositories

The number of developer's active repositories is defined as the number of repositories with activeness greater than 0 generated by each developer under the above definition of Developer Activeness.

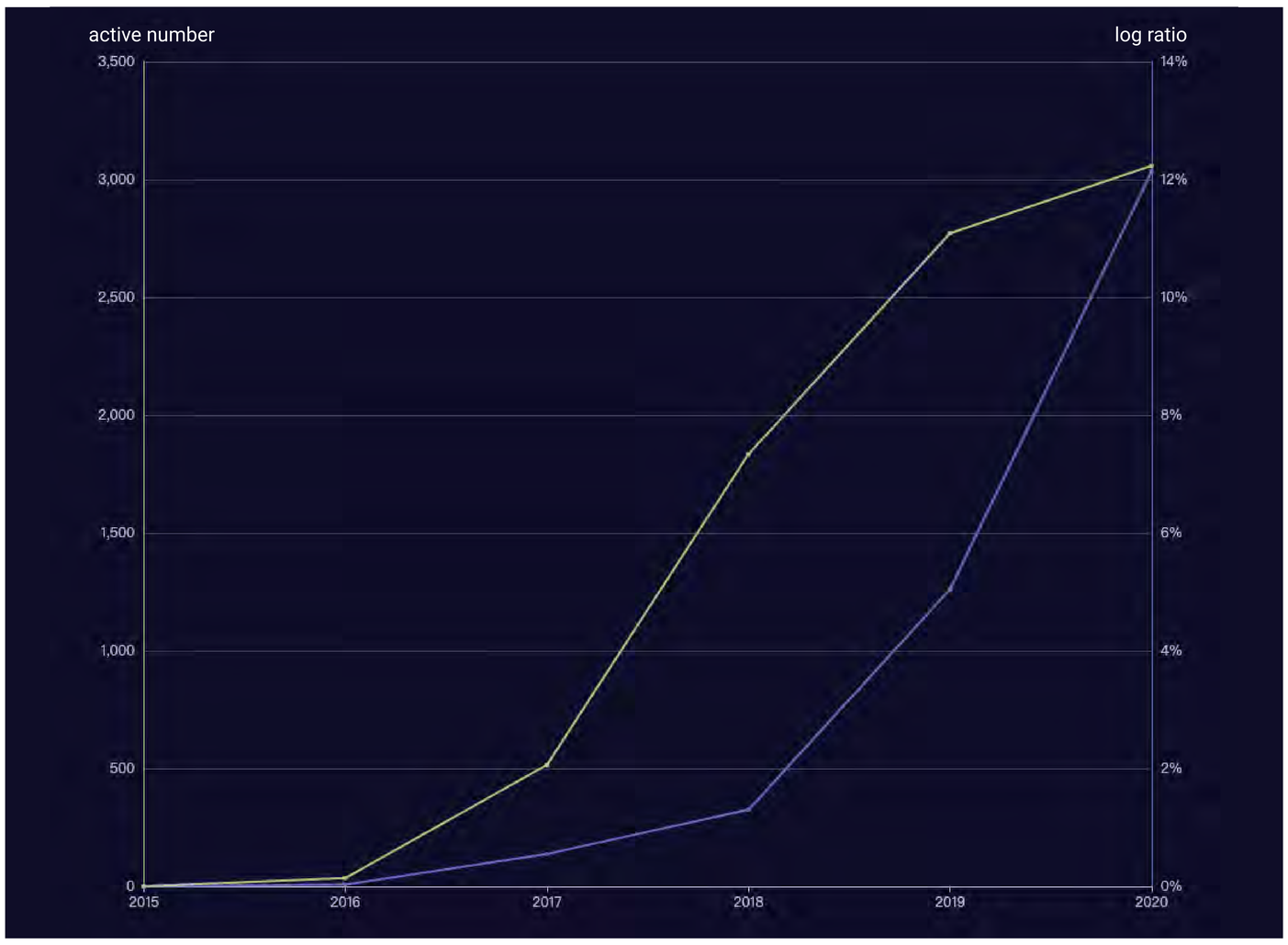

Figure 2.2 The number of active GitHub Apps and the proportion of logs

accounts, indicating that few GitHub Apps developers were registered on the platform. Among them were only 165 GitHub Apps serving more than 100 repositories, and only 59 serving more than 1,000 repositories, suggesting that most of GitHub Apps are developed for small-scale use by individuals, while the top GitHub Apps occupy a large number of markets. For example, the number of repositories that apply the most widely used *depenabot[bot]* and *dependabot-preview[bot]* exceeds 4.5 million, which is greater than the total number of repositories that other GitHub Apps serve, becoming the most active batch of accounts on GitHub.

At the same time, we have classified and listed some of the most active GitHub Apps on the entire platform, as shown in the following table.

Table 2.2 GitHub 2020 most active GitHub Apps account descriptions

actor login	actor acitivity	involved areas	functionality
dependabot[bot]	Automated dependency updates built into GitHub	Upgrading dependency	pulling down your dependency files and looking for any outdated or insecure requirements.
dependabot-preview[bot]		Upgrading dependency; Security Monitoring	Great PRs that stay up-to-date; Compatibility scores for each update; Security advisories handled automatically; Simple getting started flow
pull[bot]	Keep branches up-to-date with upstream through automatic pull requests	Up-and down stream update	Ensure branch updates; automatically integrate new changes from upstream; automatically merge or hard reset pull requests to match upstream; assign reviewer to pull requests
renovate[bot]		updating dependencies; customing grouping and schedules	Automatically update dependencies; Supports a multitude of languages; Extensive configurability; Supports shared presets as code
github-learning-lab[bot]		using guide	helping users learn how to use GitHub; getting interactive instructions and activities by the bot
github-actions[bot]	Automate your workflow from idea to production	automating workflow	automating all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub
codecov[bot]	Empower developers with tools to improve code quality and testing.	improving code quality and testing	group, merge, archive and compare coverage reports.
sonarcloud[bot]	the leading product for Continuous Code Quality & Code Security online, totally free for open-source projects.	improving code quality and secutrity	supporting all major programming languages; detecting bugs, vulnerabilities and code smells

Description 2.3: GitHub Apps

Below is a brief introduction to GitHub Apps and how to use them.

First of all, the official definition of GitHub Apps is as follows:

“GitHub Apps are first-class actors within GitHub. A GitHub App acts on its own behalf, taking actions via the API directly using its own identity, which means you don't need to maintain a bot or service account as a separate user.”

GitHub Apps are a type of product officially provided by GitHub. When a GitHub App is running, it runs under its own identity and does not represent any other users. GitHub Apps have user names ended with [bot], distinguishing them from normal users. This helps to filter them out from a large number of logs.

GitHub Apps are designed to provide numerous advantages, the most prominent of which is the ability to achieve fine-grained permission management. For instance, the GitHub Apps responsible for continuous integration can request read permissions for repository content and write permissions for states. Moreover, a GitHub App may not have read or write access to the code but can still manage issues, labels, and milestones.

We open-sourced this report project on GitHub. The repository is github-analysis-report. To automate the code development and data analysis for this project, we developed *analysis-report-bot[bot]*, a GitHub collaborative robot, in August 2020. It is worth mentioning that *analysis-report-bot[bot]* ranks 289th in activity ranking across all GitHub Apps as of today.

Having benefited from this automated collaborative bot, we believe bots that are based on GitHub Apps will be more widely used to help better manage large-scale collaborations for open source projects.

2.3. Global log time distribution

Since the GitHub event log has detailed time stamp information, insights can be gleaned through statistical analysis of the time dimension. For example, in UTC standard time, the global working time distribution is shown in Figure 2.3.

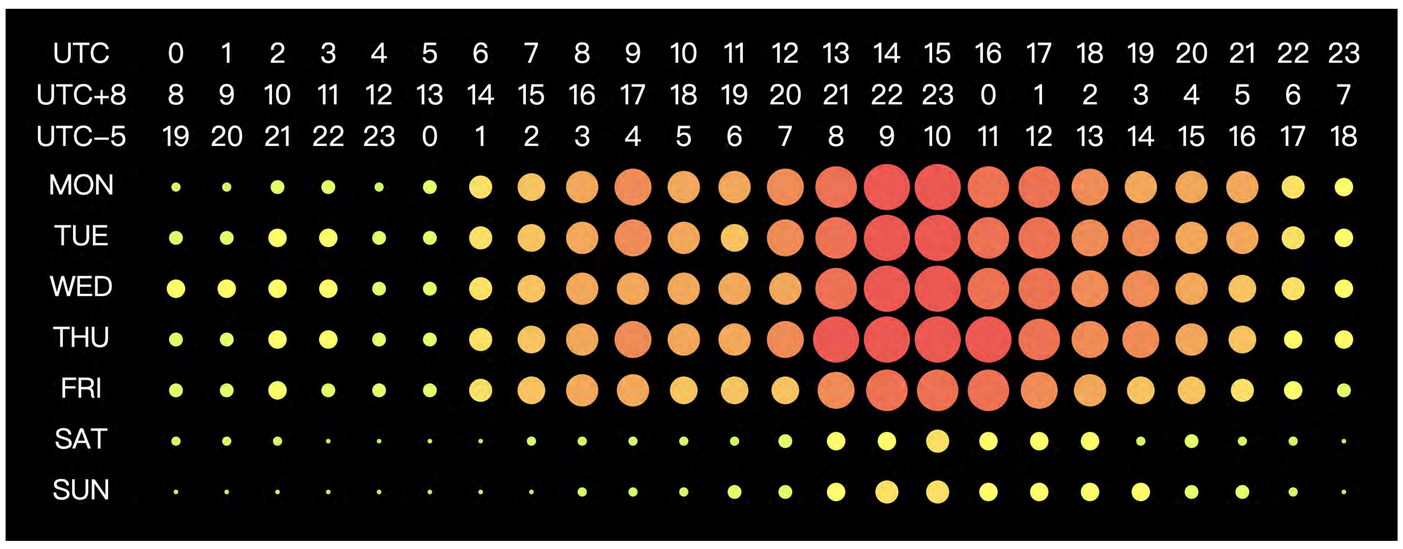

Figure 2.3 GitHub's global log time distribution in 2020

Figure 2.3 shows the table bubble plot (punch card chart) where the horizontal axis represents the 24 hours in a day and the vertical axis represents the 7 days in a week. The size of the dot indicates the relative size of the log volume generated during the period. In this figure, 0 o'clock is UTC standard time.

As shown in Figure 2.3, if we think that normal working hours of mainstream developers are from 9:00 to 21:00 every day, from a global perspective, through the log volume, we can see that the developers on the GitHub platform are dominated by Europe and the United States. And the activity on weekends is significantly lower than on weekdays, which is consistent with the GitHub Octoverse 2020 report that more developers use GitHub to work instead of developing merely based on interests.

In fact, the punch card chart is more effective when used in specific

Description 2.4: UTC standard time

UTC stands for Coordinated Universal Time. It is an internationally accepted time standard that synchronizes the time of all parts of the world. UTC time is obtained by comprehensive actuarial calculations through average solar time (based on Greenwich Mean Time GMT), new time scale corrected by earth axis movement, and International Atomic Time in seconds.

The world time zone is expressed using a positive or negative offset from UTC. The westernmost time zone uses UTC-12, which is twelve hours behind UTC; the easternmost time zone uses UTC+14, which is 14 hours ahead of UTC. The reasons for UTC+13 and UTC+14 are as follows:

1. Established in 1884, the International Date Line (IDL) was drawn to avoid the existence of two dates in one country. However, the territory of Kiribati, established in 1979, crossed the International Date Line. Kiribati's UTC includes: Line Islands (UTC+14), Phoenix Islands (UTC+13), Gilbert Islands (UTC+12), so that the dates in a country are guaranteed to be the same day.

2. Because of the Daylight Saving Time plan, in spring, the time is artificially adjusted forward by one hour. For example, New Zealand is UTC+12, and UTC+13 is used in the daylight saving time.

Beijing time is UTC+8, which is 8 hours earlier than UTC.

Description 2.5: Statistical method of working time distribution

Work time distribution statistics are based on the developer's daily and hourly log volume each week, and the linear maximum and minimum are normalized to the range 1-10, excluding the activity metric information.

community or project analysis, and can effectively reflect the distribution of project working time. As shown in Figure 2.4, in the distribution of working hours for a project throughout 2020, it can be clearly seen that the activity of a project on a Saturday is equivalent to that on working days, which is a typical 996 project.

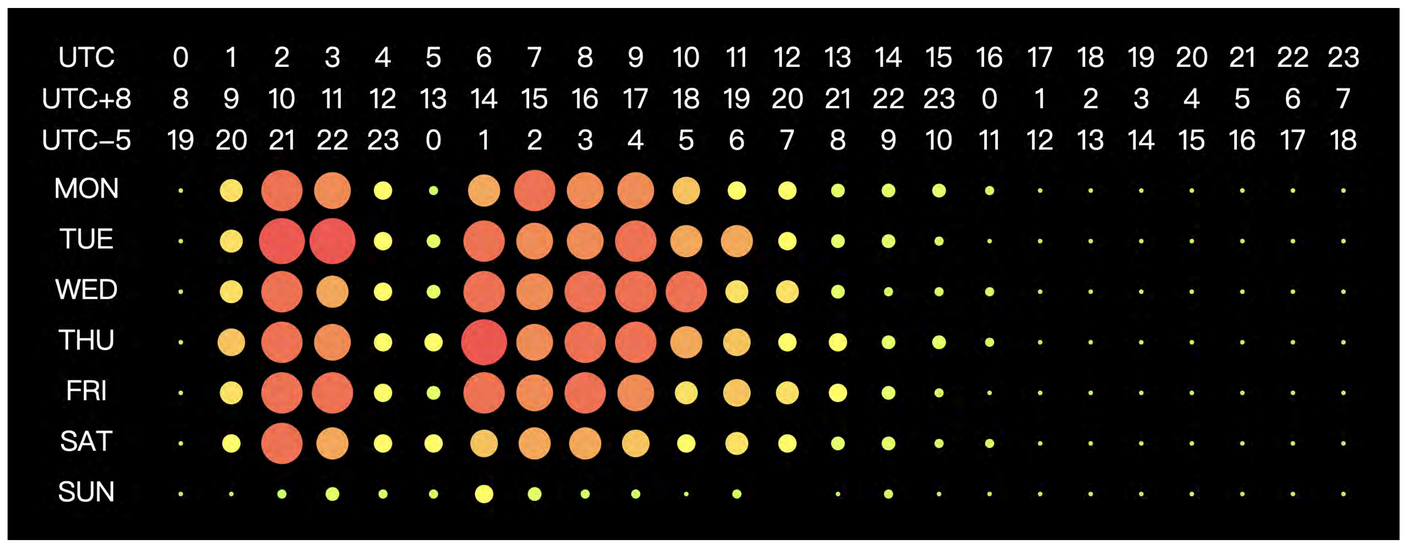

Figure 2.4 The distribution of log time of a project in GitHub in 2020

2.4. Typical Work Hour Distribution of Open Source Software Developers

We can also extract the distribution of each developer's independent log over different time periods of the day from the log data across all of GitHub, and then move and merge it into the same time zone by estimating the developer's time zone, thereby obtaining the typical work hour distribution values of open source software developers. Since this estimate requires developers to have enough behavior logs, we only extracted data of the top 50,000 developer accounts according to the global activity rankings after excluding GitHub Apps. The results are shown in Figure 2.5.

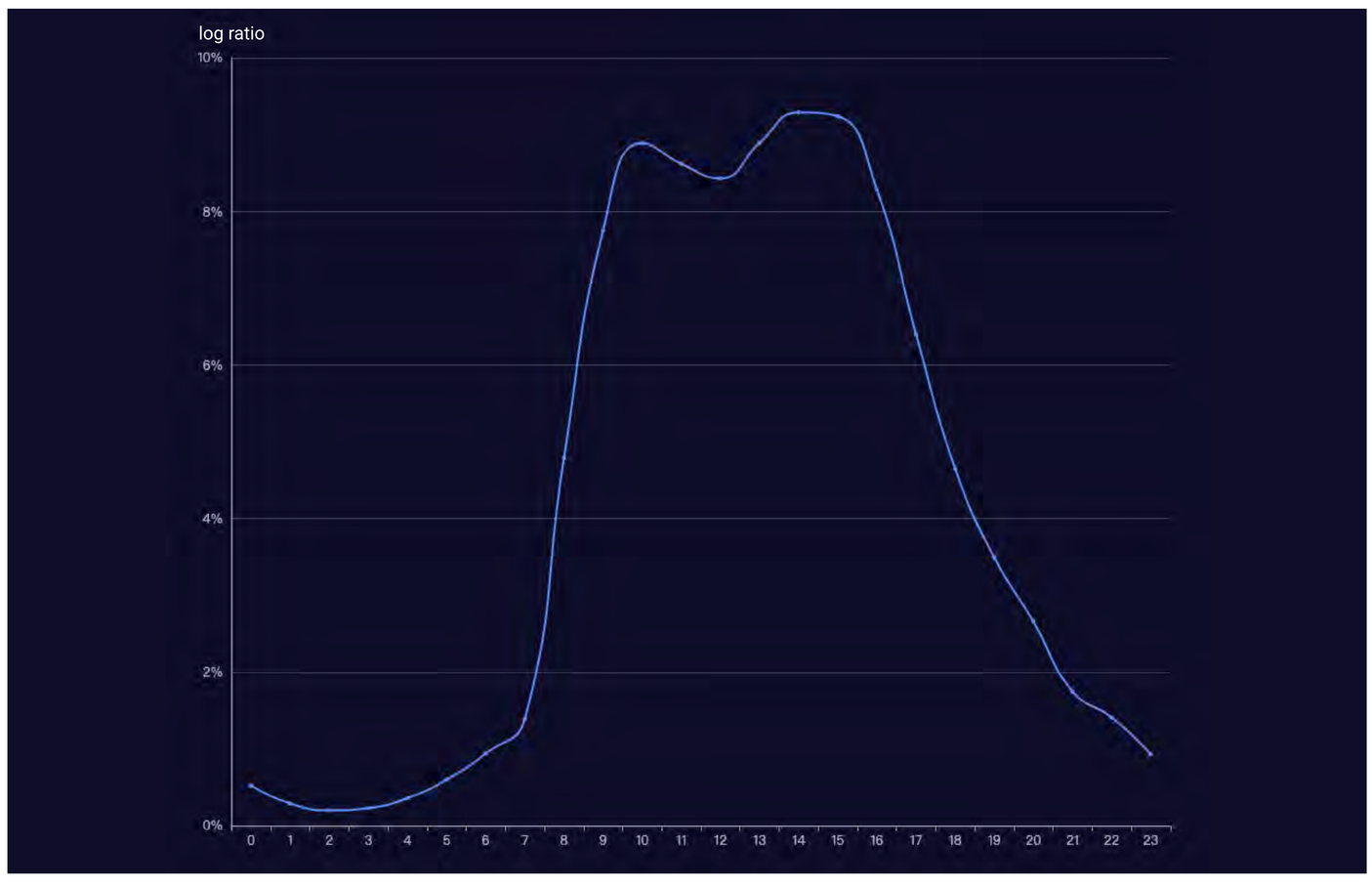

Figure 2.5 Working hour distribution of open source developers

As can be clearly seen from the above figure, developers typically

Description 2.6: Understanding method of working time distribution diagram

The 24 tick marks on the horizontal axis in the figure correspond to 24 hours in a day, and the 7 tick marks on the vertical axis correspond to all days in a week. The size of the dot at the intersection of the horizontal and vertical coordinates represents the relative numbers of GitHub logs during the period. The scale of 0 on the horizontal axis represents UTC standard time. The color of dots changes accordingly with the size of dots. The larger the dot, the greater the number of logs generated during the period, reflecting that developers are more active during that time period; by comparing the size of the dots horizontally, you can see the time period during which global developers are active on a given day; comparing the size of the dots along the vertical axis, you can compare the daily activity during a given week.

start working at 8 am local time, with a short lunch break from 11am - 1pm, and then reach a productivity peak at 3pm - 4pm and continue to output until the evening. It is more in line with the general working hours, but the proportion of developers' work output in the evening is still relatively high; they can even work until 1am in the morning. The level of activity should be significantly higher than that of other occupations.

2.5.Time Zone Distribution of All GitHub Developers

The geographic distribution of developers has always been an important aspect of the globalization indicators of open source projects. However, there has never been a proper way to estimate the developers' time zone on GitHub. While using the global GitHub log data, the personal behavior logs of the developer provide an efficient method of achieving an accurate estimate. It is possible to estimate the proportion of developers in different time zones in all GitHub projects or within a specific developer group, so as to effectively determine the global coverage of all GitHub projects or within a specific project. Based on the statistics of the top 50,000 developers in GitHub developer Activity ranking, we can estimate the distribution of all GitHub developers in various time zones as shown in Figure 2.6.

As can be seen from Figure 2.6, the Americas (the United States, Canada, and South America) have the largest distribution of developers among highly active developers, from UTC-8 to UTC-3.

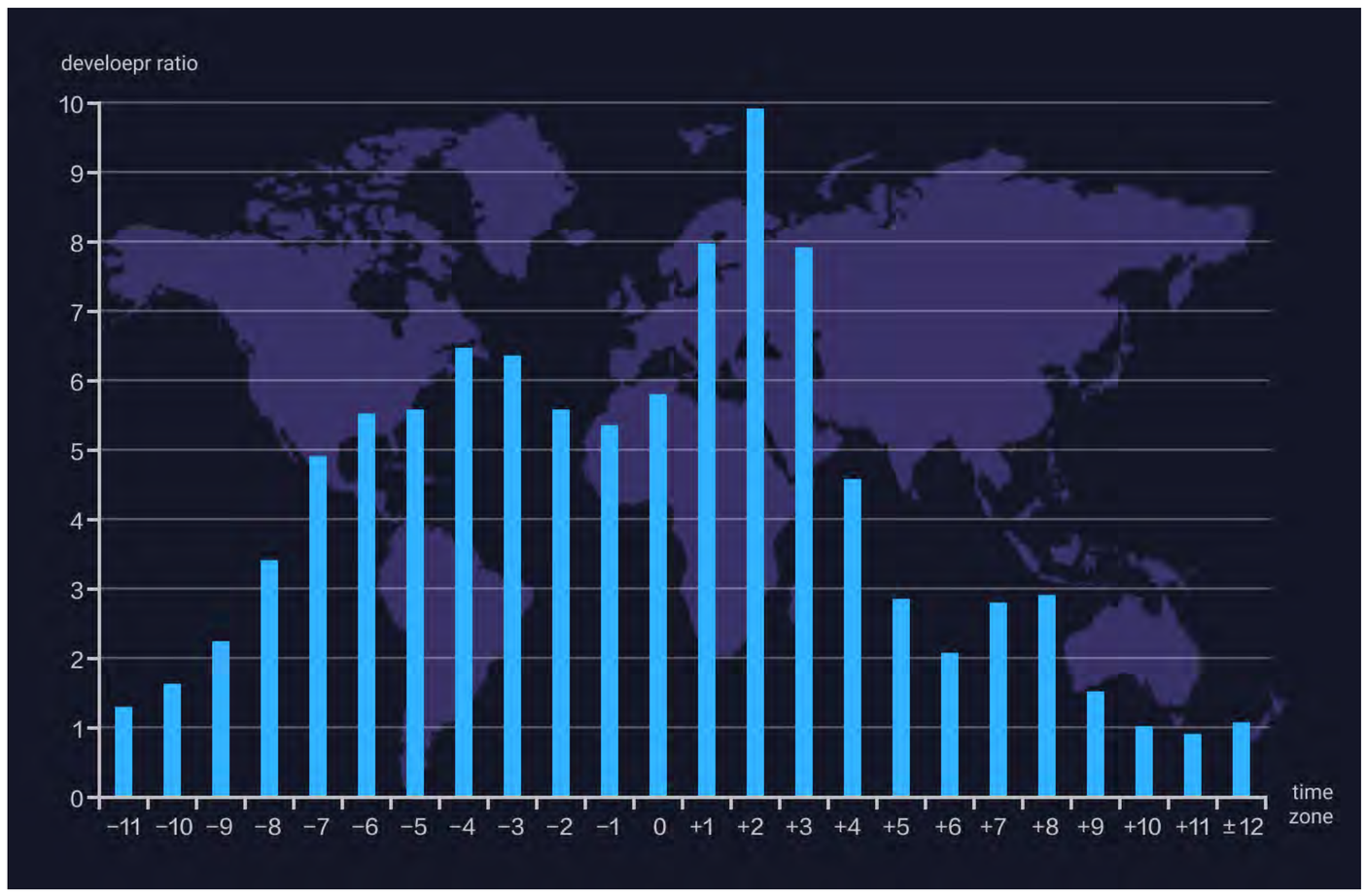

Figure 2.6 Distribution of the number of all GitHub developers in the time zone on GitHub in 2020

Description 2.7: Developer time zone estimation method

From the number of hourly logs generated by developers within a year, we calculate and obtain the maximum consecutive 12 hours of the event, and set the time from 9 am to 9 pm of the developer's local time. Then the developer's time zone can be roughly determined.
Because the time recorded in the log is the UTC standard time, we consider the last hour when the developer generates the most logs for 12 consecutive hours a day; the UTC time corresponding to the last hour (i.e., 9 pm local time) is stated as a mathematical formula: k , with the time zone calculation formula expressed as follows:

$$UTC\ offset = \begin{cases} 20 - k^*, & k^* > 8 \\ -4 - k^*, & k^* \le 8 \end{cases}$$

where, $k^* = argmax\{\sum_{i=k-11}^{k} c_i\},\ k = 0,1,...,23$

c_i represents the number of logs generated in the current hour. When i has a negative value, it means that it is the time of the previous day and needs to be converted using $i = i + 24$

Note: With this method, it is difficult to accurately estimate the time zone that a developer is in. Because the behavior of a single developer is sporadic and specific, it cannot be used to accurately estimate the time zone of a single developer, but it is meaningful in statistical dimensions; on the other hand, developers with low Activity levels also have sporadic behaviors. Therefore, for the purposes of this report, when judging the developer's time zone, we require the developer's Activity ranking to be in the top 50,000.

Although the proportion of developers in a single time zone is not the highest, the overall proportion of developers in this region is as high as about 33%. From UTC+1 to UTC+3, Europe has the highest proportion of developers in a single time zone. The UTC+1 time zone is as high as nearly 10%, with the total proportion of the three time zones being about 26%. In general, the number of developers in Asia is still relatively small, but a small peak is noted in the UTC+7 and UTC+8 region, indicating that developers in China and Russia are still more active in open source than other countries. The Pacific region (from UTC+9 to UTC-9) has the lowest proportion of developers due to population distribution.

2.6. Language Distribution for Developers

Table 2.3 shows the distribution of languages used by active developers across all GitHub projects in 2020 and the distribution of languages used by the top 100,000 active developers. Comparing the distribution of project languages, fluctuation of the ranking is observed.

Table 2.3 Comparison of the language distribution of all GitHub projects' active developers and the top 100,000 active developers in 2020

#	language(all GitHub)	accounts	#	language(top 100k developers)	accounts
1	JavaScript	305,814	1	JavaScript	15,858
2	Python	175,610	2	Python	10,866
3	HTML	159,303	3	TypeScript	7,419
4	Java	139,673	4	Java	6,665
5	Ruby	87,780	5	Go	5,094
6	TypeScript	85,116	6	C++	4,204
7	C#	54,343	7	Ruby	3,802
8	PHP	52,915	8	HTML	3,490
9	C++	47,799	9	PHP	3,000
10	CSS	46,528	10	C#	2,892

JavaScript and Python are consistent winners and runners-up in the rankings, while HTML and CSS are clearly more popular in the context of all GitHub developers. This is due to the large number of blog sites and other similar repositories on GitHub. These huge numbers of projects are generally small, independent, and maintained by individual developers.

Description 2.8 Measurement of the language used by developers

The language used by developers is defined as the language most frequently used by a developer account in 2020; it is specifically measured as the main programming language used by a developer account in 2020 for projects with the most PRs.

3. Project Analysis

3.1. Overall Analysis of All Projects on GitHub

Based on the definition of developer activity metric, we also propose a method to calculate the activity metric of open source projects. Under the given activity metric calculation method, after excluding the collaborative behavior of GitHub Apps, the total number of active projects in 2020 is approximately **11.67 million**. The distribution of the activity metric of these projects and the number of participating developers in the projects are shown in the figure below.

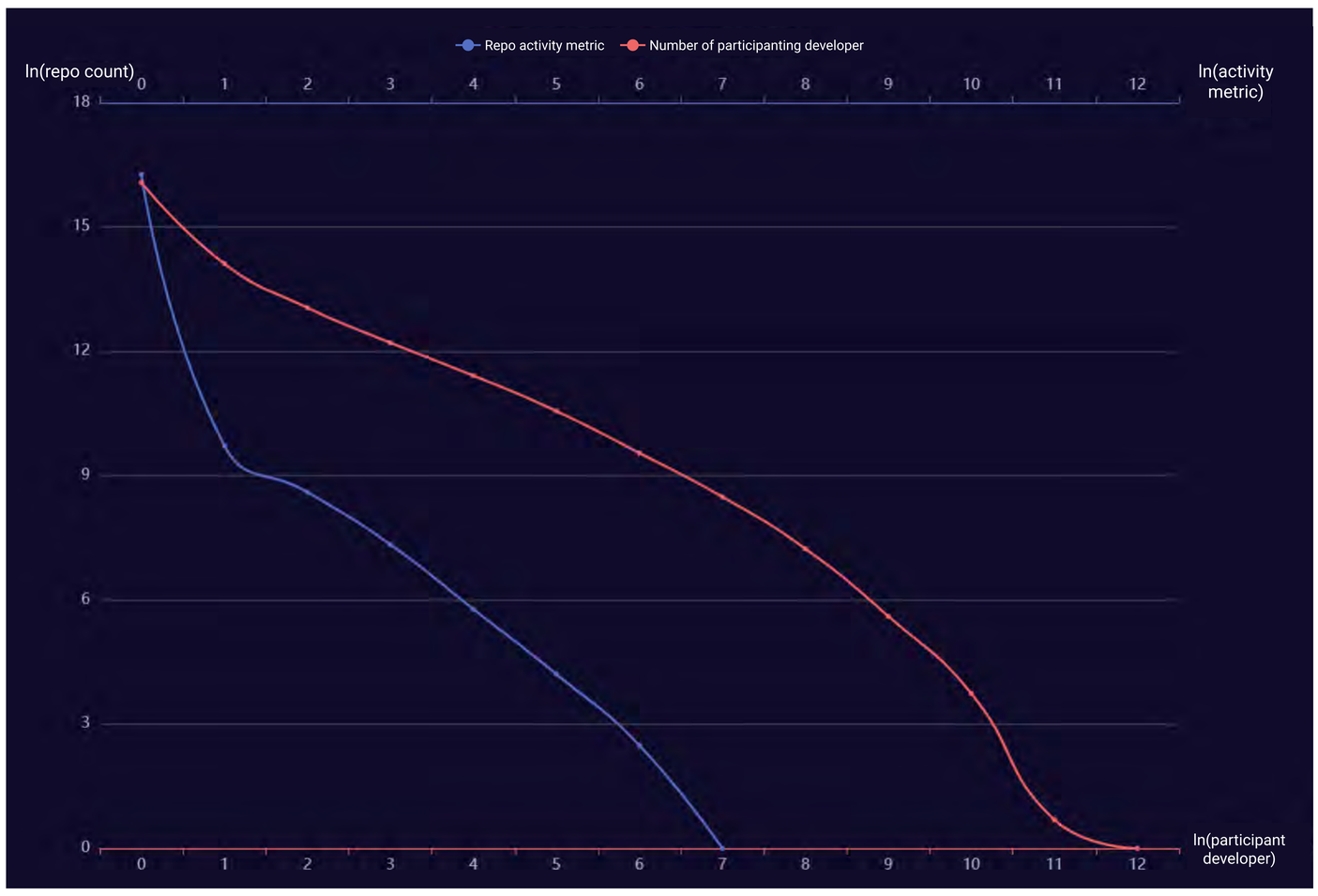

Figure 3.1 Distribution of GitHub Project Activity Metric and the Number of Participating Developers in 2020

The above figure is plotted in a double logarithmic coordinate system. The abscissa on top of the figure indicates the activity metric of the project, the abscissa on the bottom indicates the number of participating developers in the project, and the ordinate indicates the number of corresponding projects. According to statistics, the highest value of project activity metric in 2020 is 971.2, while the number of highly active projects is extremely low. More than 99.95% of projects have activity metric values below 10, which means that projects are in a low-active state. In addition, the number of total participating developers in one single project is as high as 85,532, which means that there are maximally 85,532 developers participating in the same project in 2020. However, 71.21% of projects have fewer than 10 participating developers, which means that most projects on GitHub were having participants fewer than 10 during the entire year of 2020.

Description 3.1: Project Activity Metric

Project activity metric is defined as the metric of the activity of a specific project within a period of time. The project activity metric can be calculated by summing up the activity of a single developer in one day (the calculation of activity metric at a given point or period in time is based on the developer's personal activity):

$$A_r = \sum (A_{u\_d} / day\_count)$$

3.2. Top 20 Most Active Projects

According to the definition of project activity metric, we have collected activity statistics and the ranking of active projects in 2020. Below is a list of the world’s top 20 most active projects. Since this is the report for 2020, the data used in this table are all newly generated during 2020; historically reported data was not included.

Table 3.1 Top 20 Most Active Projects on GitHub in 2020

#	repo name	activity metric	developer	issue comment	open issue	open pull	pull review comment	merge pull	star	fork
1	pddemo/demo	971.2	2	0	251,354	0	0	0.0	1	0
2	google-test/signcla-probe-repo	871.5	6	138,621	0	104,135	0	0.0	1	0
3	test-organization-kkjeer/app-test	703.1	6	109,172	40,226	47,731	7,529	18,169.3	1	0
4	elastic/kibana	698.0	5326	145,309	9,838	22,891	45,224	17,094.3	2,202	1,293
5	test-organization-kkjeer/bot-validation	691.3	8	104,083	40,242	477,29	7,425	17,724.2	2	0
6	ouyanxia2/hgmgmg	627.3	52	4	162,355	0	0	0.0	0	0
7	imamandrews/imamandrews.github.io	621.2	13	79,182	104,764	13	0	0.0	3	2
8	kubernetes/kubernetes	601.6	23279	23,5006	3,897	6,868	30,721	3,984.3	13,327	6,181
9	flutter/flutter	583.8	52414	126,337	17,538	7,222	18,438	4,124.3	33,130	6,714
10	NixOS/nixpkgs	538.1	5467	84,113	4,489	26,407	30,328	18,371.8	1,798	1,956
11	microsoft/vscode	432.2	46612	99,180	23,349	1,977	1,780	1,220.8	23,347	6,470
12	rust-lang/rust	422.8	14532	103,937	4,799	7,809	25,194	5,039.9	10,912	1,820
13	Ramos-dev/jniwebshell	417.7	3	0	108,092	0	0	0.0	2	0
14	dotnet/runtime	415.2	8914	81,740	7,886	7,942	40,000	5,662.5	4,007	1,616
15	pytorch/pytorch	413.4	19643	67,923	5,805	12,165	38,319	283.2	11,781	4,313
16	home-assistant/core	388.6	21407	76,135	6,101	7,987	30,681	5,621.0	9,832	5,526
17	MicrosoftDocs/azure-docs	339.0	18766	85,504	17,367	5,165	1,015	2,791.0	1,709	3,828
18	apache/spark	324.7	7465	138,874	0	3,877	32,252	1.8	4,581	3,263
19	elastic/elasticsearch	319.5	12865	40,330	3,610	12,515	22,599	9,791.7	7,848	4,202
20	odoo/odoo	312.8	8304	54,666	1,706	19,392	19,509	26.8	4,521	2,879

As can be derived from the above table, in the top 5 projects, all the projects have robots participating in the collaboration except *elastic/kibana*. The project listed first in the table is */demo*, which has a large number of issues. The introduction of this project is characterized by "a new issue created in this repo every minute," which is consistent with the performance characteristics.

google-test/signcla-probe-repo is in second place. This project is used to verify whether SignCLA is running correctly. *Test-organization-kkjeer/app-test* and *test-organization-kkjeer/bot-validation* are in the third and fourth places, two of which are used for bot validation.

Due to the great impact of automated actions, under the premise that all indicators are greater than 0, we have collected activity metrics and rankings of active projects in 2020. Table 3.2 presents a list of the world’s top 20 most active projects among the projects whose indicators are all greater than 0.

Table 3.2 Top 20 Most Active Projects on GitHub in 2020 (all indicators of the projects are greater than 0)

#	repo name	activity metric	developer	issue comment	open issue	open pull	pull review comment	merge pull	star	fork
1	elastic/kibana	698.0	5,326	145,309	9,838	22,891	45,224	17,094.3	2,202	1,293
2	kubernetes/kubernetes	601.6	23,279	235006	3897	6868	30721	3984.3	13,327	6181
3	flutter/flutter	583.8	52,414	126,337	17,538	7,222	18,438	4,124.3	33,130	6,714
4	NixOS/nixpkgs	538.1	5,467	84,113	4,489	26,407	30,328	18,371.8	1,798	1,956
5	microsoft/vscode	432.2	46,612	99,180	23,349	1,977	1,780	1,220.8	23,347	6,470
6	rust-lang/rust	422.8	14,532	103,937	4,799	7,809	25,194	5,039.9	10,912	1,820
7	dotnet/runtime	415.2	8,914	81,740	7,886	7,942	40,000	5,662.5	4,007	1,616
8	pytorch/pytorch	413.4	19,643	67,923	5,805	12,165	38,319	283.2	11,781	4,313
9	home-assistant/core	388.6	21,407	76,135	6,101	7,987	30,681	5,621.0	9,832	5,526
10	MicrosoftDocs/azure-docs	339.0	18,766	85,504	17,367	5,165	1,015	2,791.0	1,709	3,828
11	elastic/elasticsearch	319.5	12,865	40,330	3,610	12,515	22,599	9,791.8	7,848	4,202
12	odoo/odoo	312.8	8,304	54,666	1,706	19,392	19,509	26.8	4,521	2,879
13	tensorflow/tensorflow	308.8	33,089	62,952	7,277	3,126	7,960	1,909.5	17,321	8,392
14	labuladong/fucking-algorithm	285.7	85,532	679	261	360	37	123.1	81,970	15,273
15	istio/istio	257.2	8,512	82,976	4,338	5,519	14,572	3,826.9	5,361	1,465
16	SmartThingsCommunity/SmartThingsPublic	251.7	16,391	878	14	37,512	1,790	484.7	463	16,247
17	kubernetes/website	242.6	4,911	54,180	2,222	5,139	21,281	3,345.1	706	3,304
18	cockroachdb/cockroach	236.0	3,218	55,887	7,682	6,819	4,096	4,957.7	2,424	564
19	pandas-dev/pandas	235.3	11,008	32,697	2,906	5,282	21,512	3,870.2	5,835	3,563
20	jwasham/coding-interview-university	234.2	70,867	246	92	138	22	37.8	63,162	16,011

As indicated in Table 3.2, the most active project is Kibana, an open source data visualization dashboard from the Elastic Community. Kibana is an open source analysis and visualization tool that can easily search, visualize, and explore large amounts of data through a browser-based interface.

The front-end cross-platform development framework Flutter initiated by Google and the container orchestration system Kubernetes are ranked 2nd and 5th, respectively, which shows that Google's efforts and influence on open source have been recognized by the industry.

microsoft/vscode, Microsoft’s cross-platform code editor, and *MicrosoftDocs/azure-docs*, Microsoft’s open source built project for the Azure cloud platform, are ranked 7th and 10th, respectively, showing that Microsoft’s efforts in open source have won programmers' recognition.

It should be noted that the ranking of rust-lang/rust has risen rapidly and is in 4th place. In 2020, Rust's GitHub Star count reached 51K, Reddit Fans number has reached 125K, and the combined PR reached 8,114 throughout the year. These data show that Rust is becoming an increasingly popular programming language. As Paul Jansen, CEO of Tiobe software, said, "All the verbose programming and sharp edges of other languages are solved by Rust while being statically strongly typed. Its type system prevents run-time null pointer exceptions and memory management is calculated compile-time.” This may be one of the reasons why Rust is becoming more and more popular.

3.3. Ranking of the Activity Metric of China's Projects

At the same time, we have also collected a list of Chinese-initiated projects through various channels and gave a ranking of the activity metric of Chinese projects, as shown in Table 3.3.

Table 3.3 Top 20 Most Active Chinese Projects on Github in 2020

#	repo name	activity metric	developer	issue comment	open issue	open pull	pull review comment	merge pull	star	fork
1	pingcap/tidb	210.1	5,831	53,022	2,801	4,969	10,928	3,459.2	4,862	1,052
2	ant-design/ant-design	193.3	23,620	32,026	4,836	3,131	3,320	2,130.7	12,709	8,052
3	PaddlePaddle/Paddle	127.4	4,842	15,329	2,256	5,656	9,625	3,478.2	3,574	786
4	tikv/tikv	81.7	2,593	17,817	997	2,019	5,547	1,279.9	2,129	434
5	apache/shardingsphere	75.3	5,267	9,055	1,713	3,235	1,858	2,539.5	3,834	1,443
6	apache/incubator-tvm	70.4	2,148	7,961	437	2,112	8,506	1,540.1	1,454	662
7	pingcap/docs-cn	65.1	532	8,202	96	2,965	6,959	2,315.9	140	320
8	apache/incubator-echarts	64.2	11,638	7,650	1,620	324	346	194.5	6,664	4,463
9	pingcap/pd	60.9	437	13,325	667	1,667	4,972	1,297.7	214	224
10	alibaba/nacos	59.9	9,956	7,042	1,640	706	827	410.0	6,347	3,450
11	NervJS/taro	54.7	7,469	9,339	2,231	917	135	551.5	5,250	1,012
12	youzan/vant	54.2	9,806	4,897	1,661	715	201	554.4	4,672	4,502
13	pingcap/docs	53.9	314	7,014	64	2,736	5,226	2,257.8	90	164
14	ElemeFE/element	52.7	11,749	4,993	1,762	297	10	33.3	6,853	3,411
15	apache/skywalking	51.9	5,556	6,783	1,084	860	3,455	583.4	4,201	1,471
16	PaddlePaddle/PaddleOCR	47.9	9,394	4,039	1,033	573	622	420.0	8,430	1,664
17	apache/incubator-dolphinscheduler	47.1	2,588	9,364	1,269	1,407	730	902.7	1,835	909
18	apache/apisix	45.4	2,923	5,855	1,109	1,029	3,383	715.0	2,496	579
19	seata/seata	45.1	7,339	3,754	785	517	1,805	313.5	5,261	2,296
20	pingcap/tidb-operator	45.1	425	8,627	703	1,498	3,683	1,172.1	240	140

From the above list, we found that PingCAP has stellar performance in open source. They represent 6 projects on the list of the Top 20 projects. These projects include the top ranked *pingcap/tidb*, an open source distributed relational database, which is independently designed and developed by PingCAP. Among these projects are also the distributed transaction key-value database *tikv/tikv*, and document projects *pingcap/ docs-cn* and *pingcap/docs*, which show that PingCAP attaches great importance to project documentation.

Alibaba's contribution in open source is nothing short of impressive. They have 2 projects on the Top 10 list; namely, a set of component libraries *ant-design/ant-design*, which placed second, encapsulated in React by Ant Financial, and a feature set dedicated to configuring and managing microservices *alibaba/nacos*.

Baidu has demonstrated an outstanding performance in the field of artificial intelligence. Two projects of its deep learning platform *PaddlePaddle*, namely the core framework *Paddle* and related tool libraries, made the list.

The list of China’s Top 20 most active projects includes Alibaba’s Ant Design component library, JD’s development framework *taro*, which is based on the React front-end framework, and the Vue UI

Description 3.2: Chinese-initiated project and corporate project ownership

We collected projects that are open-sourced by China-based developers and local technology companies headquartered in China through multiple data sources. For details, see the open source project configuration list. In the event any omissions or errors are discovered, please submit an Issue or PR for supplements and corrections.

component library *Element*, open-sourced by the front-end team of Ele.me (acquired by Alibaba), etc. This shows that in China, front-end groups are more active in the community; in addition, front-end codes are generally less sensitive, which is the reason that companies are more open to them. However, it should be noted that the list has many fewer core projects than front-end ones.

It is worth noting that compared with the China Top 20 most active projects presented in the GitHub 2019 report, the number of projects incubated by Apache has increased from 2 to 6, indicating that the attention and participation of the Apache project has been increasing.

Most of the major open source projects are supported by technology companies. We calculated the activity status of open source projects maintained by technology companies in 2020. The results are shown in Table 3.4:

Table 3.4 GitHub Chinese Enterprise Activity Ranking List of Open Source Project in 2020

#	company	activity metric	repo	issue comment	open issue	open pull	pull review comment	merge pull	star	fork
1	Alibaba	1,571.1	1,496	130,558	33,947	29,097	22,615	17,471.6	216,980	68,864
2	PingCAP	778.4	151	139,255	8,138	25,401	61,538	18,880.4	18,008	5,058
3	Baidu	671.2	540	55,265	12,592	20,720	23,380	13,475.9	70,960	22,148
4	Tencent	432.3	388	21,446	8,599	10,264	2,870	7,088.8	69,198	19,348
5	JD	153.0	74	20,126	4,504	4,483	2,043	3,214.2	13,119	3,316
6	Huawei	101.8	200	10,322	1,709	2,930	3,867	2,005.2	8,758	3,168
7	DiDi	89.4	63	3,114	1,290	827	207	508.7	20,489	3,907
8	Youzan	88.6	58	7,259	2,760	1,409	634	1,068.1	9,509	5,820
9	Bytedance	59.2	85	1,973	645	785	659	514.9	14,034	1,671
10	WeBank	57.9	59	2,197	718	3,501	596	2,411.9	5,225	1,902
11	Xiaomi	50.4	98	1,767	1,604	1,007	3,001	691.6	5,823	1,760
12	Meituan	46.9	68	1,356	564	305	17	147.0	10,879	2,573
13	Bilibili	42.7	51	1,306	446	132	52	66.4	10,278	2,295
14	360	39.8	147	1,769	810	441	40	231.7	8,105	1,914
15	Juejin	39.5	26	3,866	578	661	3,624	546.9	4,208	810
16	CTrip	36.9	25	2,346	537	216	276	130.5	6,196	2,562
17	Linux China	34.1	16	226	10	3,862	11	3,123.3	482	302
18	Netease	25.0	119	1,603	777	313	32	149.6	3,880	1,445
19	Deepin	18.6	267	2,555	931	326	21	132.7	1,339	821
20	Qunar	7.1	43	113	54	56	10	9.9	1,653	478
21	Vipshop	7.1	14	112	127	66	0	14.0	1,604	421
22	Douban	3.7	41	98	43	158	58	128.6	508	99

As can be seen from the above table, Alibaba stands out from Chinese companies in many aspects, some of which are on par with the sum of the rest of the Top 10 companies. They are also doing well in terms of communitization and openness.

It can be seen that in recent years, major technology companies have continuously increased investment in open source communities and ecosystems. PingCAP announced the completion of a $270 million Series D financing in H2 2020, creating a new milestone in global database history. Similarly, PingCAP's performance in the current open source industry is also quite

impressive, exceeding Baidu to rank second in the list. The number of their pull review comments exceeded the ones from Alibaba, which shows that PingCAP's open source community is highly active. AI is the most distinctive competitive edge of Baidu's open source, such as the deep learning platform *PaddlePaddle* and the autonomous driving platform *Apollo*. Youzan's ranking has risen very fast. It may be attributable to the excellent performance of its open source project *youzan/vant*, which is a lightweight mobile UI component.

3.4. Language Distribution for Projects

Based on the definition of project activity metric, we calculated the total activity of projects aggregated by programming languages across all projects, as shown in Figure 3.2. We found that projects using JavaScript have the highest overall activity, followed by Python and TypeScript.

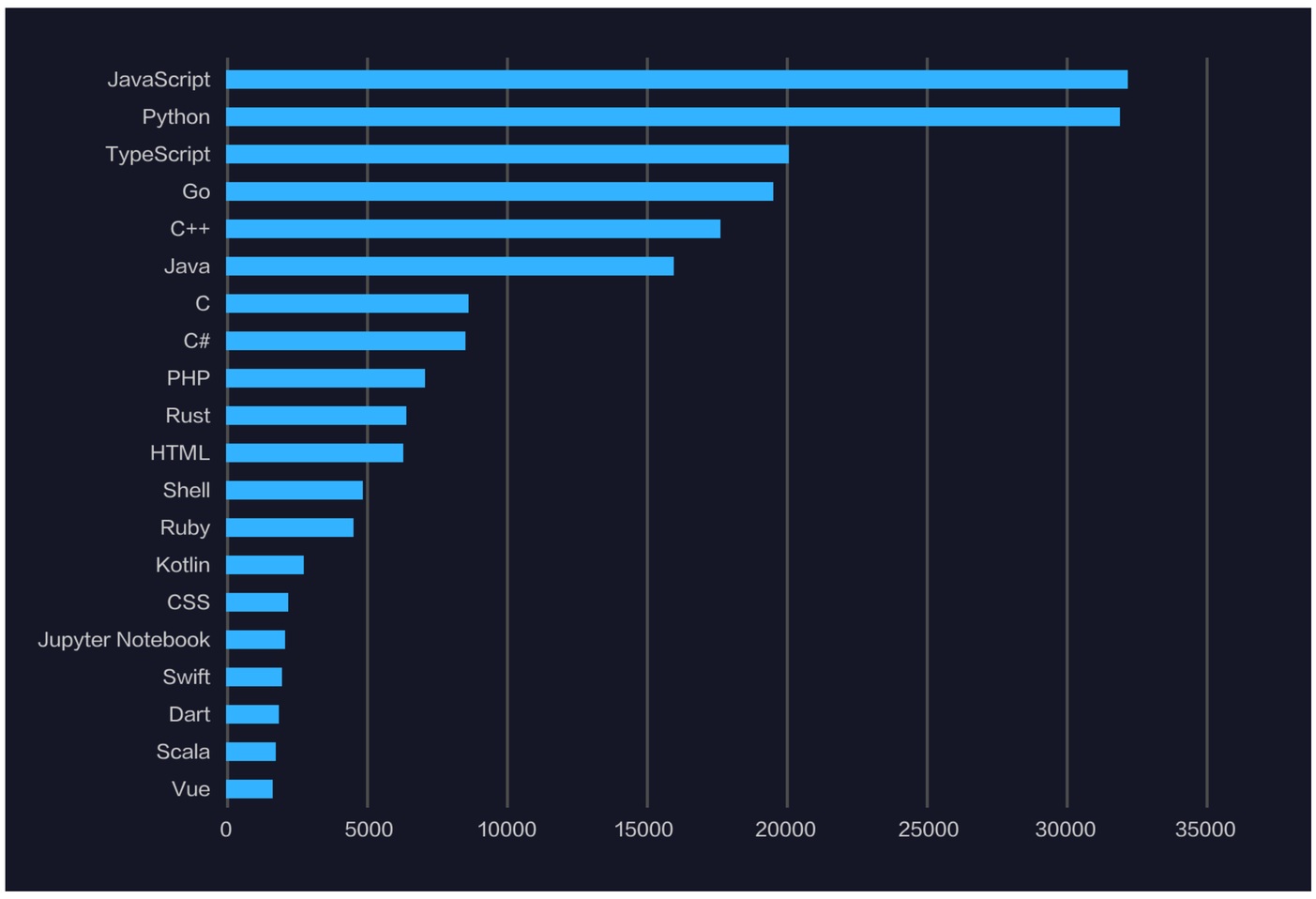

Figure 3.2 Language Distribution for Projects in 2020

The reasons for the high ranking of JavaScript are as follows:

- JavaScript is a scripting language directly embedded in HTML, which can be understood by web browsers. JavaScript wins the popularity contest because it does not need to be compiled and runs directly in the browser environment.
- Mainstream front-end frameworks like Angular developed by Google, React developed by Facebook, and Vue, are all part of the JavaScript ecosystem.
- JavaScript has become virtually omnipotent, as almost all mainstream Internet applications have a large amount of

JavaScript code on their front ends. For example, email and social media tools we use every day.

Python's popularity has not decreased, ranking second. Possible reasons are:

- Compared with other mainstream programming languages, Python is more readable, easy to learn, and easy to maintain.
- Python has a wide range of applications. It comes with various modules plus a wealth of third-party modules, which eliminate a lot of "re-engineering the wheel" work and can implement multiple functions faster.
- The wave of artificial intelligence has further promoted the development of Python. Many artificial intelligence tasks and big data analysis will be implemented in Python first.

It is noteworthy that the popularity of TypeScript has risen sharply. Possible reasons are:

- Although TypeScript is a relatively young language, which has only been released for eight years, it has inherited part of the popularity of JavaScript.
- TypeScript only needs to be compiled once, and users can run it on the server, browser, or anywhere they like.
- In some cases, TypeScript is irreplaceable, and the code is more readable and maintainable.

3.5. OpenGalaxy

As shown in section 2.2, the results obtained through activity metric analysis will be affected by automated collaboration behavior, and the activity metric of projects in different stages of life cycles may not be comparable. So, in this report, we have introduced OpenGalaxy, a collaborative relationship network for global projects.

OpenGalaxy builds a collaborative relationship between projects through all the developers' collaboration behaviors on GitHub. It provides a calculation method about influence and clustering through graph algorithms. When using this calculation method, projects are linked by the actions of developers, and the influence of projects will no longer be simple statistics. Excellent developers and excellent projects attract each other to improve the project's performance. Better results can be obtained even without filtering out automated behavior.

Description 3.3: Description of how to build OpenGalaxy

OpenGalaxy, the collaborative network across all GitHub projects, is based on the definition of developer activity metric in Description 2.1, and is constructed through the collaboration of developers in multiple projects. The specific calculation method is:

Mathematical formula:

$$R_{ab} = \sum_i \frac{A_{ia} A_{ib}}{A_{ia} + A_{ib}}$$

The mathematical formula: $A_{ia} A_{ib}$ is the activity metric of developer i on project a and project b, respectively, following the activity metric calculation method in Note 2.1 (since the blind obedience of Star behavior impacts the clustering result, we do not count Star action). The mathematical formula: R_{ab} is the degree of collaboration between project a and project b. That is, the degree of collaboration between the two projects is the sum of the harmonic average of the activity of all the jointly active developers.

Based on the above method, the number of global active projects is **1.057 million**, and the total number of collaborative relationships is **26.07 million**. We come to a result that the collaborative network across all GitHub Projects is an undirected graph with **1.057 million** nodes and **26.07 million** edges. We apply the weighted PageRank algorithm to the graph to converge (128 iterations in total) and get the PageRank value of each node as the node's influence metric. The Louvain method for community detection is applied to this graph, and the obtained first-level community division is recorded as the project clustering result, which is the field of the project under the collaborative network.

All analysis is based on this graph. Considering the visualization effect, the OpenGalaxy rendering uses about **221,000** items with a PageRank greater than 1, and the collaborative relationship uses a total of about **2.922 million** edges with a collaborative correlation greater than 2.

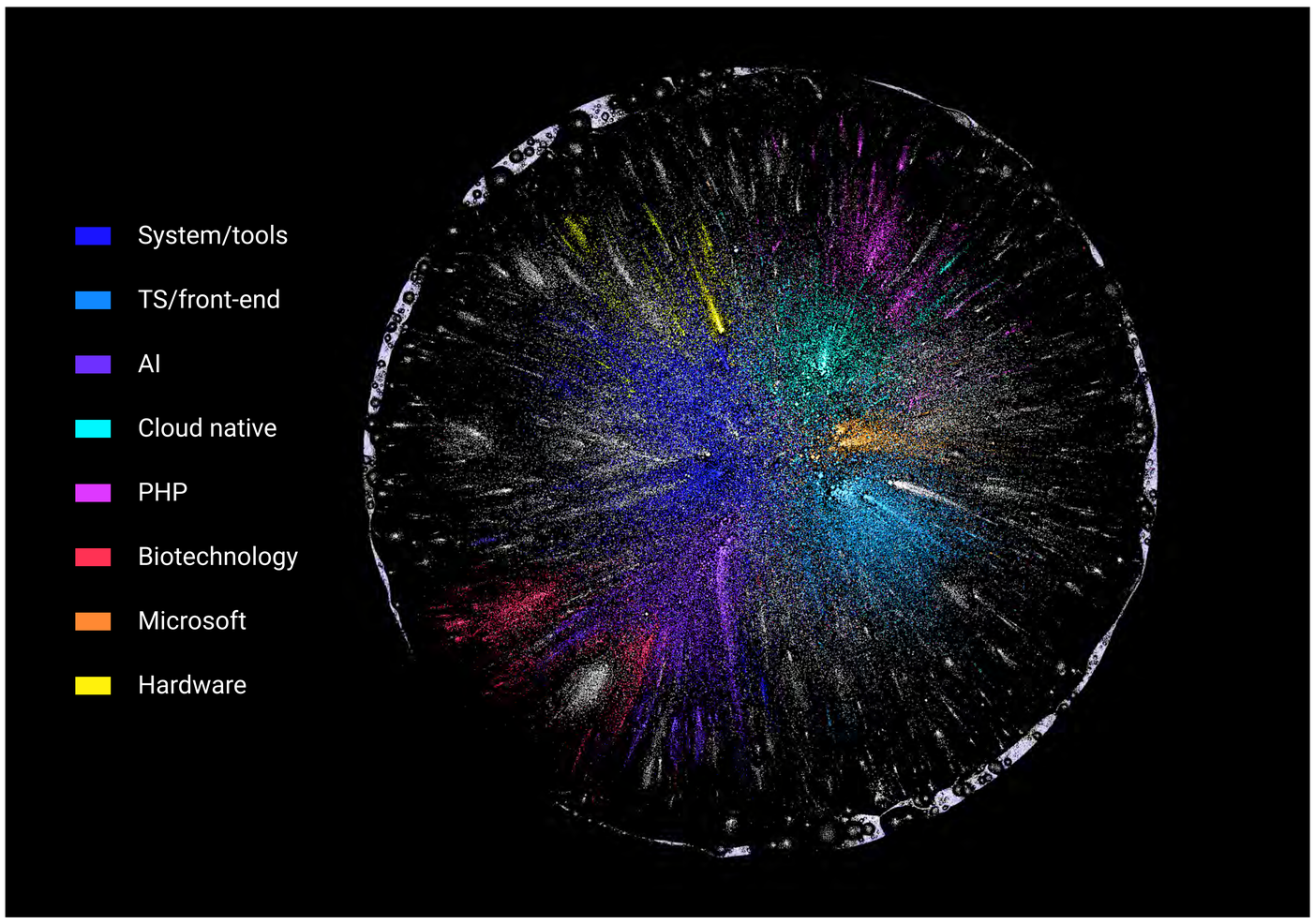

Figure 3.3 GitHub global project collaboration network in 2020 - OpenGalaxy 2020

3.5.1 Ranking of Project Influence Across all of Github

Figure 3.3 is a collaboration network diagram composed of the most active 221,000 open source projects on GitHub in 2020. The size of the node in the figure indicates the influence of the project, and the color of the node indicates the collaborative clustering result to which the node belongs. Unlike the activity metric, the project influence has better algorithm stability and can better reflect the influence of the developer's overall activity on the project. Please refer to Description 3.3 for the calculation method of project influence.

Evaluated on the influence of the collaborative network, the top 20 projects with the greatest influence among all GitHub projects in 2020 are shown in the following table.

Table 3.5 OpenGalaxy 2020, Top 20 Project Influence Across All of GitHub

#	repo name	influence
1	microsoft/vscode	977.3
2	flutter/flutter	593.3
3	tensorflow/tensorflow	484.8
4	microsoft/TypeScript	448.5
5	DefiitelyTyped/DefinitelyTyped	446.7
6	microsoft/WSL	431.7
7	golang/go	404.6
8	gatsbyjs/gatsby	381.3
9	kubernetes/kubernetes	370.1
10	vercel/next.js	354.5
11	pytorch/pytorch	340.2
12	NixOS/nixpkgs	335.4
13	rust-lang/rust	328.9
14	home-assistant/core	317.9
15	Homebrew/homebrew-core	311.5
16	microsoft/terminal	301.1
17	nodejs/node	292.9
18	MicrosoftDocs/azure-docs	285.5
19	electron/electron	276.5
20	facebook/create-react-app	276.3

As can be seen in Table 3.5, VS Code has jumped to #1 when measured by influence, from #5 when measured by activity. With its influence about 64.7% higher than Flutter, which ranks 2nd, VS Code has become the world's most influential project.

While VS Code has become globally the most popular IDE, it has also produced a vast amount of collaborative associations with top projects in other fields. As shown in Table 3.6, in addition to VS Code's own coding language and technology-related projects such as TypeScript and Electron, VS Code has a close relationship with other language support plug-ins. For example, *microsoft/vscode-go*, a Go language extension for VS Code, has the highest degree of collaborative relationship with *golang/go*, which ranks 7th in terms of influence. The Python support extension *microsoft/vscode-python* has in its top 10 collaborative related projects the Tensorflow, which is ranked 3rd in influence, and the famous AI library Pandas (influence ranking 201st). *Dart-Code/Dart-Code*, the Dart language supporter and debugger for VS Code, is a bridge between VS Code and Flutter, as Dart is mainly used for Flutter development.

Table 3.6 VS Code's Top 10 Collaborative Related Projects Across all of GitHub in 2020

#	repo name	relationship	influence
1	microsoft/TypeScript	803.3	448.5
2	microsoft/vscode-remote-release	576.7	214.2
3	microsoft/vscode-python	500.2	205.3
4	DefinitelyTyped/DefinitelyTyped	392.1	446.7
5	microsoft/terminal	374.6	301.1
6	microsoft/vscode-cpptools	316.8	131.9
7	microsoft/WSL	289.6	431.7
8	electron/electron	261.4	276.5
9	flutter/flutter	246.3	593.3
10	microsoft/vscode-docs	236.3	38.6

To sum up, OpenGalaxy has the feature that excellent projects will be linked together by excellent developers. As a result, the influence index will not be falsely high due to automated behavior, but has a better algorithm stability instead. Since the change in influence reveals the developer community activity change, influence becomes a better metric to reflect a project's influence level among all of GitHub projects.

3.5.2 Collaborative Network Clustering Across All GitHub Projects

We use a collaborative network to describe the collaborative relationship of all GitHub projects. Therefore, projects can be clustered through project collaboration correlation and community discovery algorithms to obtain clustering effects based on collaborative behavior. This method can be used to roughly classify projects. We use the Louvain method for community detection, a graph clustering algorithm to cluster the above network and give the following classification results.

Table 3.7 Top 8 clustering results of OpenGalaxy 2020 of all GitHub Projects

#	area	top repos	repo count
1	System/tools	rust-lang/rust,Homebrew/homebrew-core, ytdl-org/youtube-dl	26,601
2	TS/front-end	microsoft/vscode,microsoft/TypeScript,vercel/next.js	25,693
3	AI	tensorflow/tensorflow,pytorch/pytorch,conda-forge/staged-recipes	21,024
4	Cloud native	glang/go,kubernetes/kubernetes,moby/moby	12,637
5	PHP	laravel/framework,symfony/symfony,composer/composer	8,365
6	Biotechnology	rstudio/rstudio,bioconda/bioconda-recipes,apache/arrow	7,258
7	Microsoft	microsoft/WSL, microsoft/terminal, dotnet/runtime	6,626
8	Hardware	home-assistant/core, espressif/arduino-esp32, arduino/Arduino,	6,352

The graph clustering algorithm can group closely collaborated projects into one category based on the weight of the edges of graphs. This method helps us discover project types in all GitHub data and provides a novel perspective on open source projects' technical direction and strategy. Since the clustering results of low-active projects are not accurate, the number of projects in Table 3.7 is based on the 220,000 most influential projects in OpenGalaxy. Among these projects, front-end projects account for the biggest percentage. The number of Bioinformatics ranks 8th. The result shows that more low-active developers are more inclined toward front-end projects; the number of Bioinformatics projects is small, but the overall quality is higher.

The above clustering results obtained through the collaboration network are not entirely accurate, especially for system/tool projects, which do not have good collaboration stickiness or consistency and relevance on the technology stack. The nature of projects in other fields is more evident on the technical side. Interestingly, Microsoft’s open source projects form a collaborative category on their own. It also shows that Microsoft executes its open source strategy thoroughly. The direct use of GitHub as a platform for project development has led to a high degree of relevance among its project groups and also made Microsoft’s project more eye-catching in terms of influence.

3.5.3 Collaborative Connectivity and Isolated Islands in All GitHub Projects

Network connectivity is one of the fundamental analysis methods when using graph networks to build collaborative relationships. Analyzing the GitHub project collaborative network has brought us many new perspectives and insights.

Based on connectivity analysis, the 1.05 million open source projects across all of GitHub are divided into 44,990 connected subgraphs. Among them, 935,231 projects form a huge connected subgraph under a collaborative relationship. The largest subgraph of all the other 44,989 connected subgraphs contains 200 items; the smallest contains 1 item; there are only 9 connected subgraphs containing more than 100 items.

Specifically, 89% of the global active projects constitute a huge collaborative network, which is the core of the GitHub open source world. Besides, there are nearly 45,000 small isolated islands, which are not connected to the core network. This means that all developers on these projects only collaborate in their own groups but have never worked on other projects. On the other hand, no developer in the core network has had any collaborative behavior in these project groups. This is a very strong condition, so exploring in these numerous small closed collaborative worlds can offer numerous interesting insights. The data of 9 clusters which have more than 100 projects is shown in Table 3.8.

Table 3.8 Isolated Collaborative Island of all GitHub projects in 2020

#	repos	repo count	type
1	devfactory-dev/*	200	Automated test
2	labagithub2020-andrew3/*	187	Automated test
3	ICS3UI-2020-Q1/*	169	Teaching
4	natewachter/astr-119-hw-5, ThuraDwe/astr-119	167	Teaching
5	MIEE-ACS/*	158	Teaching(Russian)
6	bertikq/pibd21-NemovAL, SpokyOky/TimashevISEbd21	147	Russian
7	PowerShellForGitHubTeam/*	147	Automated test
8	team-grilled-cheese/back-end, Kayla-SA-W/NannyHelper-Client	132	Learning group
9	mednajjar/DevSpace, terbouchi/repo	120	French

Among the isolated collaborative islands in 2020, there are three project groups for automated testing. There are a large number of active projects under these organizations or accounts, most of which are automatically created, tested, and deleted. Many of them do not have much information on their homepages.

There are also some collaborative groups for teaching and learning purposes. For example, projects under ICS3UI-2020-Q1 and

MIEE-ACS were created through GitHub Classroom. MIEE-ACS is a group from Russia. The fourth-ranked project group seems to come from different individual developers. Nevertheless, we found a community of students from UC Santa Cruz (University of California, Santa Cruz). Assistant teacher Adriantsh uses GitHub to assign homework, and students learn through forking and developing over the teacher/assistant's repository.

Three other project groups are spontaneous collaborative network groups, and their primary purpose is to learn. Among them, the developers of the two project groups are from Russia and France. Language may be the reason that they are not linked to the core open source world. We also found isolated collaborative islands from Japan, Belarus, Ukraine, Vietnam, and other countries in even smaller project groups.

Moreover, a large number of automated tests have brought us another important insight. As GitHub becomes more open and integratable, its closed source nature brings many problems. One of the most critical points is the automated testing when applications are integrated with GitHub. When performing integration testing, you have to use the GitHub online environment instead of doing a complete testing in a sandbox environment. This not only makes it difficult to develop GitHub integration tools, but also leads to a large number of test repositories in the GitHub online environment. Meanwhile, behavior logs of these test repositories are also recorded and become noise in data analysis, while at the same time complicating it. Although we raised this issue with GitHub as early as 2019, it has thus far not been resolved.

4. Case Study

4.1. Case Study Analysis Method

4.1.1 Analysis of Developer Time Zone Distribution

Similar to the analysis for all GitHub projects, we selected the most active 50,000 developers on GitHub. Among them, we further screened out the developers of the case study projects. These developers' time zone distributions were used to estimate the number of developers in each time zone for the case study projects.

Note: Refer to the 3.4 above for the method of determination of the developer time zone

4.1.2 Open source Quadrant Analysis

This report proposed an OpenQuadrant method to describe the performance of an open source project in terms of influence, globalization, and community size. The open source quadrant analysis is represented by a scatter chart. The horizontal and vertical dimensions indicate the project influence and the project globalization, respectively. For the convenience of visualization, we use the logarithmic form of the two indicators and use the size of the dots to depict the number of active participants in the project as well as reflecting the scale of a community.

Thus, the open source quadrant includes four areas:

- **Foresighted**: Projects in this area have a strong influence and a high degree of globalization;
- **Leading**: Projects in this area have a strong influence, but the degree of project globalization is low;
- **Acting**: The influence of the projects in this area is weak, but the degree of globalization of the projects is high;
- **Incubating**: Projects that fall into this area have weaker influence and a lower degree of project globalization.

It should be noted that this classification is not the only answer. We are trying to let everyone have a more detailed and rich understanding of open source technology development through this visualization.

The specific calculation methods for influence, globalization, and the number of participants are as follows.

4.1.2.1 Influence Metric

The influence metric of case study projects is identical to the influence metric of all GitHub projects. For the calculation method, please refer to the description of 3.6.

4.1.2.2 Globalization Metric

Many factors affect globalization. In this study, we considered two factors, region and the number of developers, when calculating the project globalization indicator. For the region factor, we consider calculating the time zone distribution of developers participating in the project. When the developers' time zone distribution tends to be even across all time zones, the standard deviation of time distribution becomes smaller, such that the degree of globalization of the project is higher. Therefore, we first judge the time zone of all developers of the project and then count the number of developers of each 24 time zones, and on this basis, calculate the standard deviation of the number of people of each time zone. Regarding the number of developers, which refers to the total number of developers participating in the project, the greater the number of developers, the higher the degree of globalization of the project. Given that the behavior of developers has occasional characteristics, the calculation of the region factor is inaccurate, which leads to inaccurate results of globalization indicators. Therefore, we consider the following calculation formula for globalization indicators:

$$\sqrt{\frac{24}{\sum_{i=-12}^{11}(\bar{x_i}-x)^2}}\bar{x}^2, \quad x_i \text{ indicates the number of developers of the project in time zone } i$$

The above formula shows that the globalization index is directly proportional to the square of the average number of developers in the project time zone and inversely proportional to the standard deviation of the number of developers in the project time zone. This calculation method can effectively evaluate the degree of globalization of projects with a small number of developers. This calculation method also has a positive effect on the calculation of globalization indicators for projects with a large number of developers.

Note: There are many influencing factors of globalization, and the factors considered in the above calculation method are limited. Later, more factors need to be considered to evaluate project globalization indicators.

4.1.2.3 Number of Participants

The number of project participants represents the total number of participants who log behaviors on the project. When visualizing scatter plots, we use linear min-max normalization for the number of participants to map the value to the range of 1-10.

4.2. CNCF Foundation Projects

4.2.1 Introduction to CNCF Foundation

The full name of CNCF is Cloud Native Computing Foundation. It has the key components of the global infrastructure, and the foundation brings together the world's top developers, end-users, and suppliers. In addition, the CNCF Foundation operates the world's largest open source software developer conference. The "Cloud Native Foundation" is a foundation under the Linux Foundation, which is a non-profit organization. The slogan of "Cloud Native Foundation" is "Insist on and integrate open source technologies to orchestrate containers as part of the microservice architecture". It is committed to the popularization and promotion of cloud native applications and also focuses on the promotion of fast-growing open source technologies on GitHub. It has made outstanding contributions to the healthy development of the open source ecosystem.

Currently, projects that have graduated from CNCF include containerd, CoreDNS, etcd, Fluentd, Harbor, Helm, Kubernetes, Prometheus, Vitess, Envoy, Jaeger, Open Policy Agent, Rook, TiKV, TUF, etc.

The projects currently being incubated in the CNCF Foundation include Argo, CloudEvents, CNI, KubeEdge, Operator Framework, Thanos, SPIRE, SPIFFE, OpenTracing, Notary, NATS, Linkerd, gRPC, Falco, Dragonfly, CRI-O, Cortex, Contour , Buildpacks, etc.

4.2.2 Analysis of Developer Time Zone Distribution

Figure 4.1 shows the time zone distribution of CNCF developers. It shows that the time zone distribution of developers in this case is relatively similar to all github projects, indicating that projects in the CNCF have a high degree of globalization.

Besides, developers in the UTC+7 to UTC+8 time zone account for about 11%, which is an increase compared to the proportion of all github projects. This shows that more Chinese developers participated in CNCF projects compared to all GitHub projects.

Figure 4.1 Time zone distribution of developers in the CNCF

4.2.3 Analysis of the Open Source Quadrant

Figure 4.2 shows the visualization results of the open source quadrant analysis of the CNCF. The Kubernetes project deserves to be in the first place in terms of influence, globalization, or the size of community developers.

Figure 4.2 The open source quadrant of the CNCF

4.3. Linux Foundation AI & Data Foundation Projects

4.3.1 Introduction to LF AI & Data Foundation

LF AI & Data is an umbrella foundation of the Linux Foundation, supporting open source innovation in artificial intelligence, machine learning, deep learning and data. The purpose of creating LF AI & Data is to support open source artificial intelligence, machine learning, deep learning and data, and to create a sustainable open source artificial intelligence ecosystem that can easily create artificial intelligence and products and services through open source technologies. In addition to some supporting services, including membership and fund management, ecosystem development, legal support, public relations/marketing/communications, event support, and compliance scanning, it also supports diversification, development, and fosters open source projects in the community.
Currently, projects that have graduated from the LF AI & Data Foundation include Acumos, Angel-ML, Egeria, Horovod, and ONNX.

The projects being incubated are Adlik, Adversarial Robustness Toolkit, AI Explainability 360 Toolkit, AI Fairness 360 Toolkit, Amundsen, DataPractices, DELTA, Elastic Deep Learning (EDL), Feast, ForestFlow, JanusGraph, Ludwig, Marquez, Milvus, OpenDS4Streamer, Pyro, SOAJS, sparklyr.

4.3.2 Analysis of Developer Time Zone Distribution

Figure 4.3 shows the time zone distribution of developers in the field of data and artificial intelligence under the umbrella foundation LF AI & Data. We can see that the developer time zone distribution of projects in this field has changed a lot compared with all GitHub projects. About 16% of developers are from UTC+7 to UTC+8. In these time zones, Chinese developers are the most involved, which reflects that China has more data and artificial intelligence practitioners.

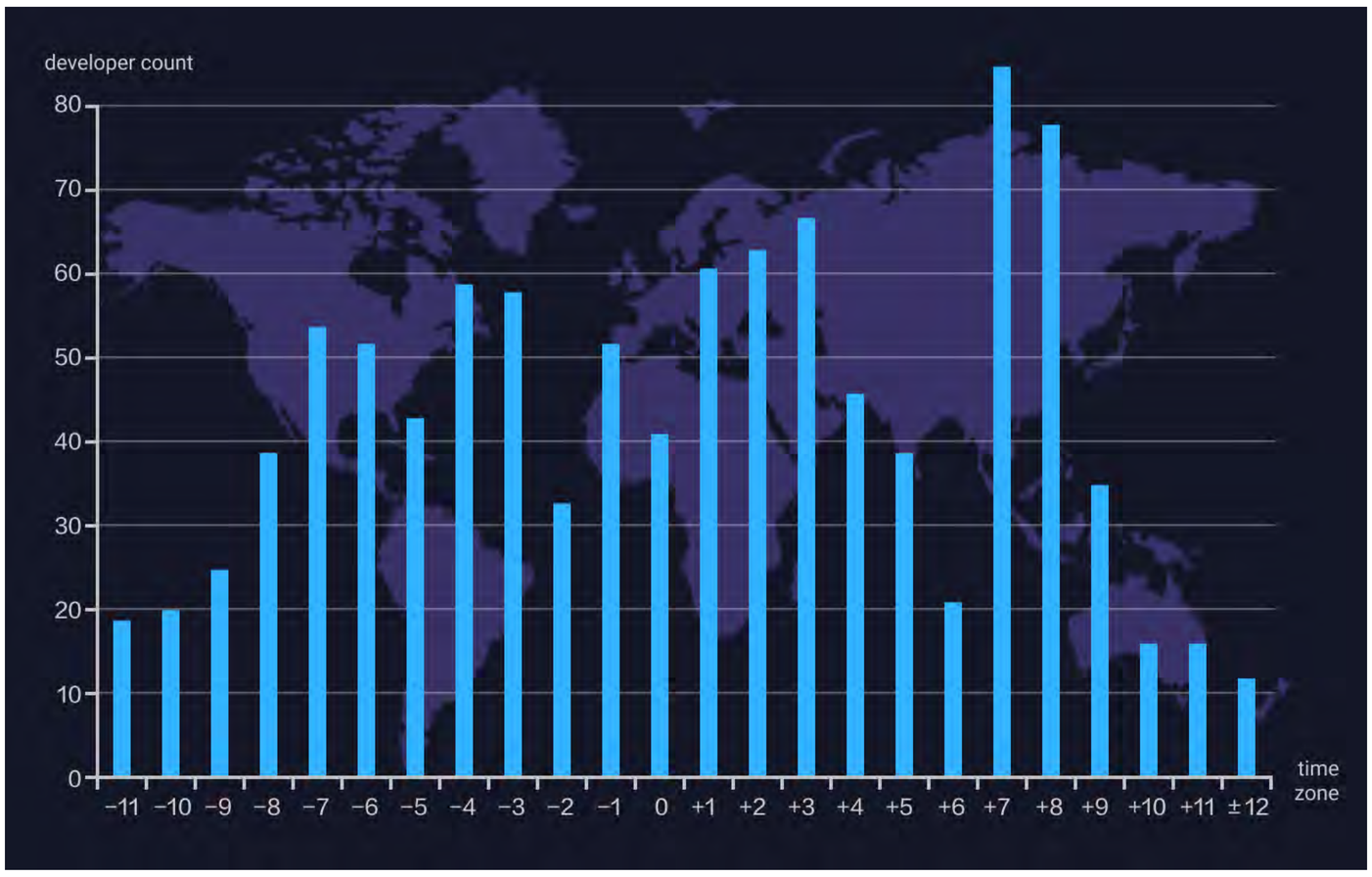

Figure 4.3 Time zone distribution of developers in the field of data and artificial intelligence under LF AI & Data

4.3.3 Open Source Quadrant Analysis

Figure 4.4 shows the visualization results of open source quadrant analysis in the field of data and artificial intelligence under the umbrella foundation LF AI & Data. It can be seen that there are many projects that have done well in globalization in this field, which is in line with the global trend of artificial intelligence.

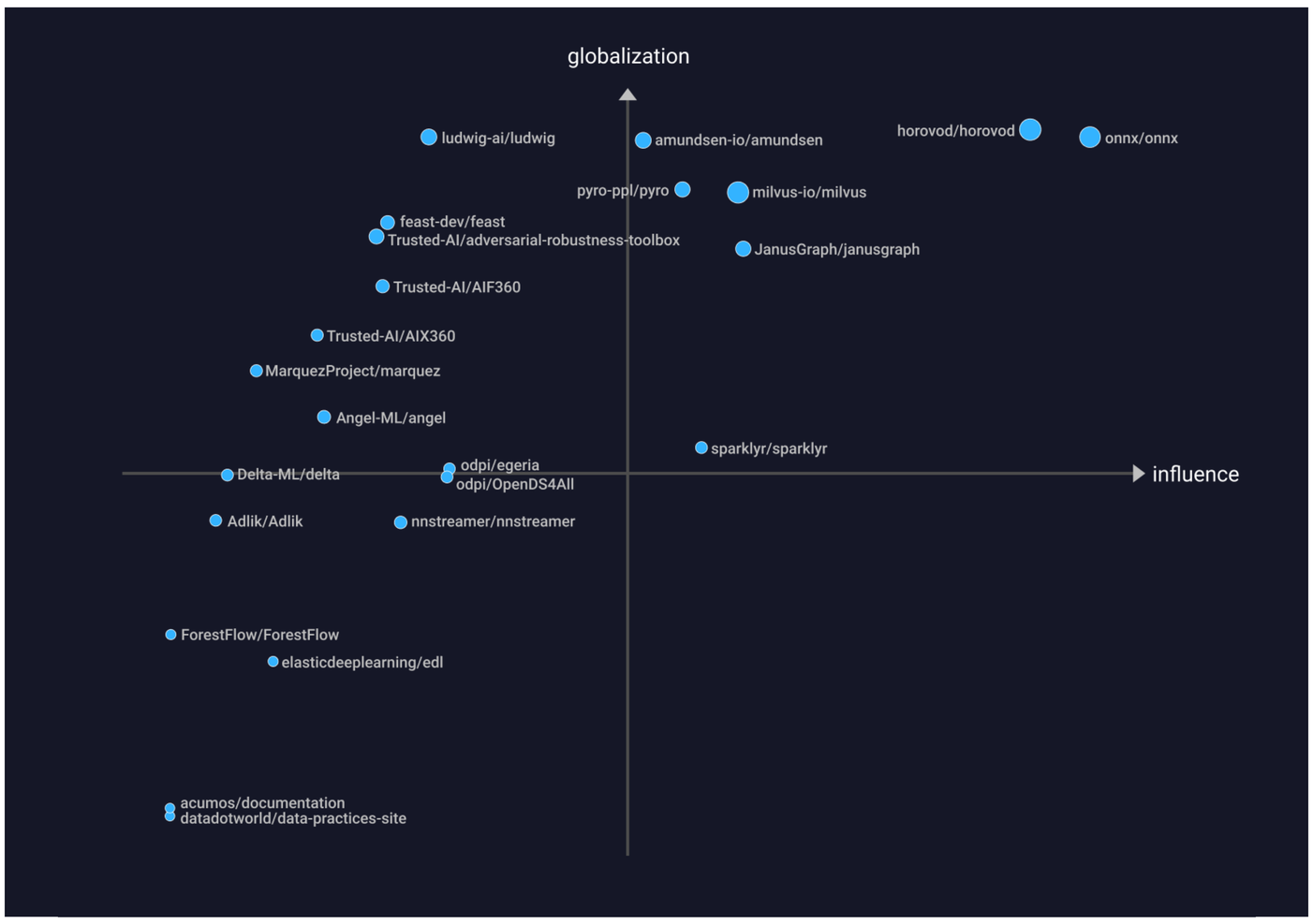

Figure 4.4 The open source quadrant of data and artificial intelligence under LF AI & Data

4.4. Apache Software Foundation Big Data Projects Analysis

4.4.1 Introduction to the Apache Software Foundation

The Apache Software Foundation (ASF) was established in 1999 and is a nonprofit public organization established in the United States under Section 501(c)(3). The mission of the foundation is to provide software for the public good. The foundation helps independent individuals and organizations to understand how open source can represent an advantage in a fiercely competitive market. The focus is not to produce software, but to guide the community that produces software. Through a meritocratic development process known as "The Apache Way," more than 800 individual Members and 8,000 Committers across six continents successfully collaborate to develop freely available enterprise-grade software, benefiting millions of users worldwide.

At present, the open source projects with the category label "big-data" in Apache include Accumulo, Lens (in the Attic), Airavata, Ambari, Oozie, Bigtop, Zeppelin, Flink, Flume, Spark, Sqoop, Storm, etc

4.4.2 Analysis of Developer Time Zone Distribution

The time zone distribution of project developers in the big data field under Apache is shown in Figure 4.7. As can be seen from the figure, there are two obvious peaks in the number of developers in the time zone distribution of big data projects under Apache, about 16% of developers are from UTC+2 to UTC+3 time zone, about 15% are from UTC+7 to UTC+8 Time zone, and about 26% are from the UTC-8 to UTC-3 time zone. Although Chinese developers in the field of Apache big data are more involved, projects in this field are still dominated by European and American developers.

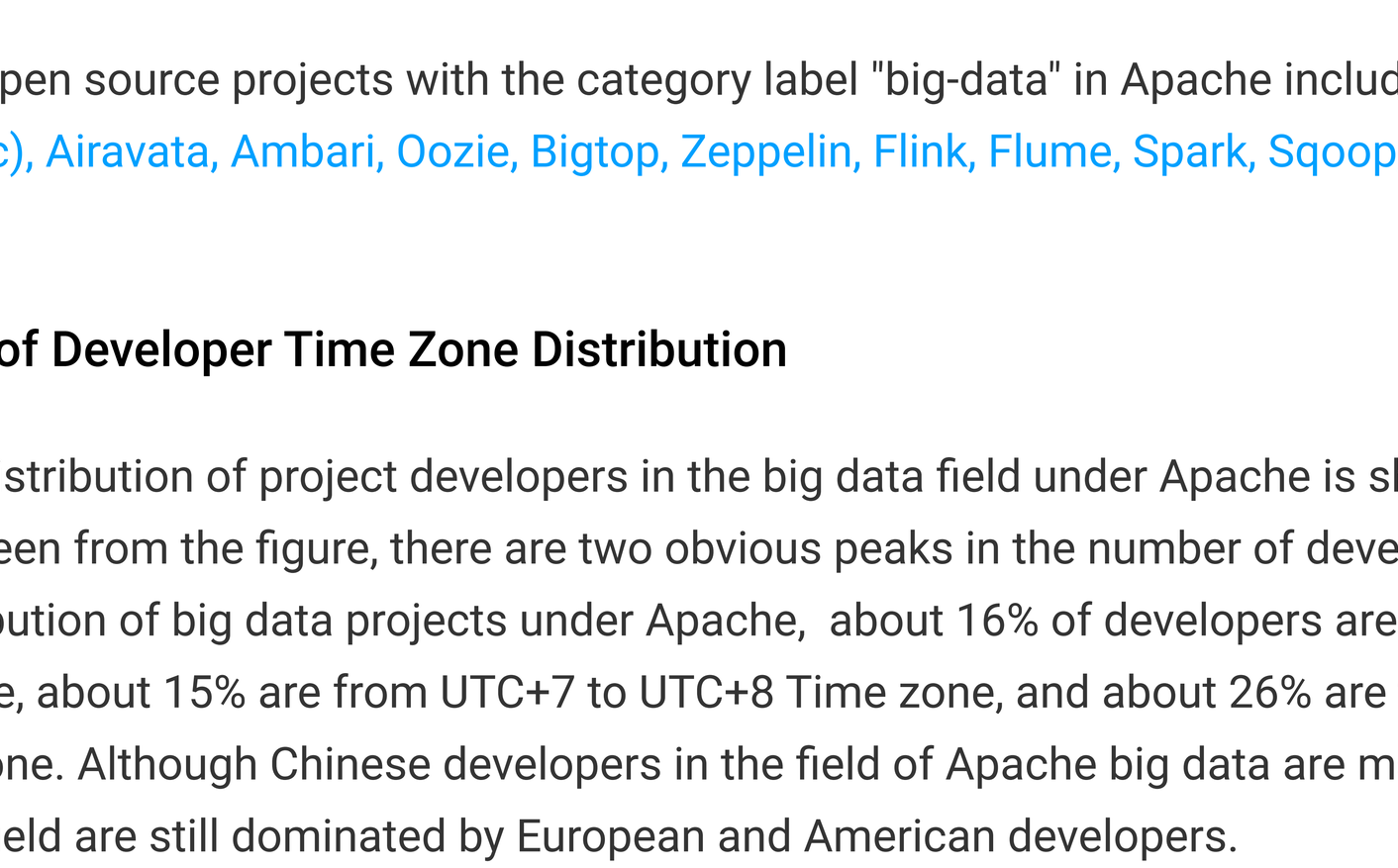

Figure 4.5 Time zone distribution of developers in the big data field under Apache

4.4.3 Open Source Quadrant Analysis

Figure 4.6 shows the visualization results of open source quadrant analysis in the big data field in Apache. It can be seen that the overall distribution of projects in this area is well dispersed. Star projects such as Spark, Flink, and Hadoop occupy the top position.

Figure 4.6 The open source quadrant of big data in Apache

4.5. VS Code Case Study

4.5.1 Overall situation

As a modern lightweight code editor, VS Code is Microsoft’s showpiece. It supports syntax highlighting, intelligent code completion, custom hotkeys, bracket matching, code snippets, code comparison, Git and other features of almost all mainstream development languages. The plug-in mechanism ensures its compatibility and scalability, making it particularly remarkable in the open source ecological chain.

As a Polaris in the open source ecosystem, the VS Code repository still maintained its vigor in 2020. This year, a total of 206,645 records were generated by VS Code, which has nearly doubled compared to 121,490 in 2019; this year, the VS Code repository has an average annual activity metric of 385, ranking 7th across all projects of GitHub; this year, 46,639 developers (including collaborative robot accounts) were active in the project.

4.5.2 Analysis of Developer Time Zone Distribution

The time zone distribution of developers in the VS Code repository is shown in Figure 4.9. As can be seen from the figure, the developers of this project have a relatively even time zone distribution,

which is characteristic of a typical global collaborative project. Active developers reached a peak from UTC+1 to UTC+3, indicating that the project is still dominated by Europe. The very next is North America, accounting for about 22%, in which area the eastern United States is more active. The Asian time zone is relatively less active, but still occupies a position that cannot be ignored with a proportion of about 15%.

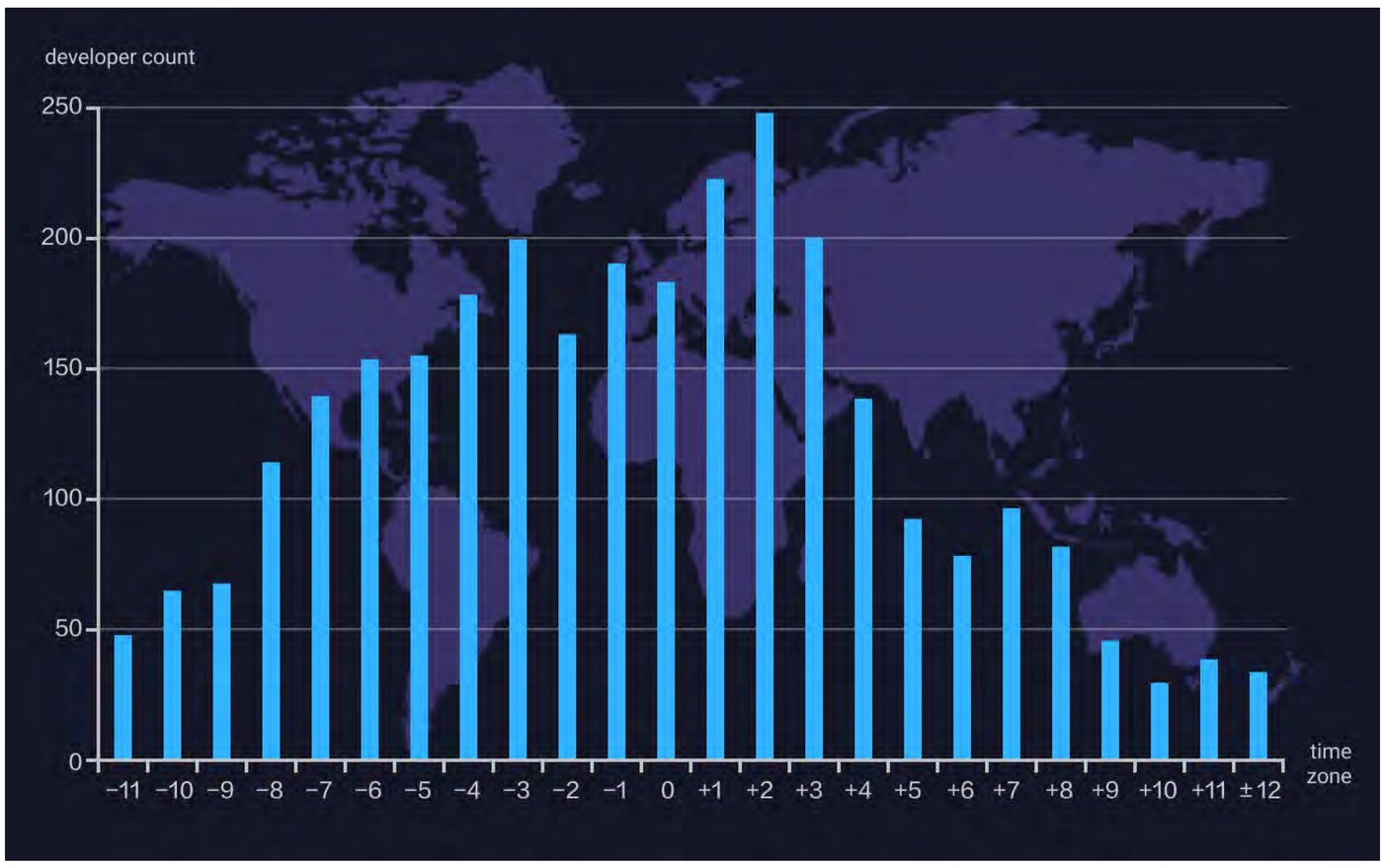

Figure 4.9 Developer Time Zone Distribution of VS Code

4.5.3 Analysis of Developer Collaboration Network

At the same time, we built a developer collaboration network for VS Code using the repository's event logs of 2020, as shown in Figure 4.10.

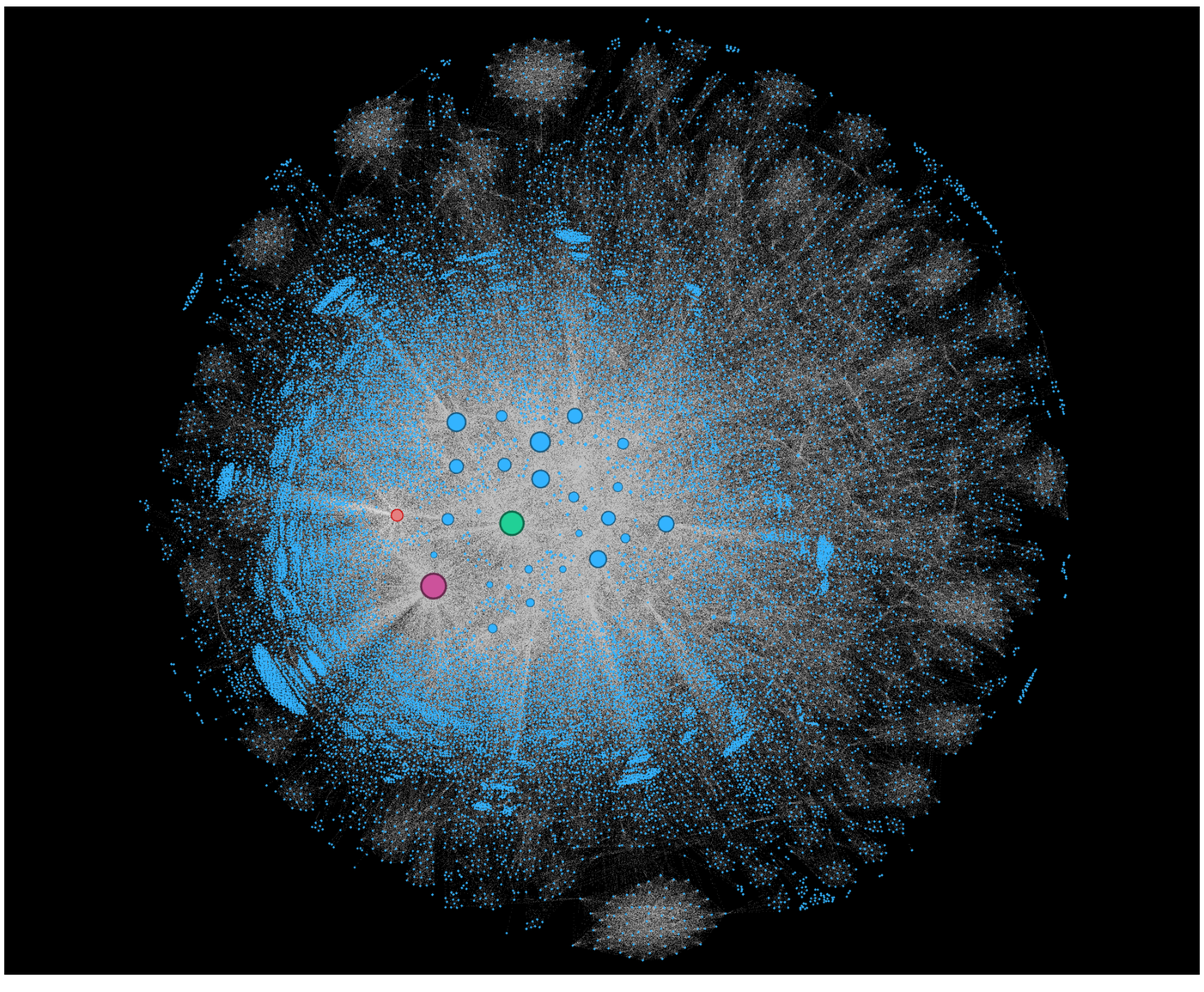

Figure 4.10 Developer collaboration network of VS Code

As shown in the figure above, the collaborative network is composed of more than 20,000 developers. The nodes are the developers' GitHub accounts, the edges are the collaboration relationships, and the size of the nodes reflects the activity level of the corresponding developer accounts. In this collaborative network, the larger nodes at the core of the network are the core team members of VS Code. They not only have a high activity metric, but also have a high collaborative relationship with other developers. There are about a hundred members in the team group. The closely followed outside nodes are active users or contributors of VS Code. They may submit Issues or PRs for discussion or contribution at any time. The members in this group number in the thousands. The outermost and largest number of developers are general users of VS Code and occasional contributors, most of whom only ask questions or discuss issues that they care about.

At the core of this collaborative network, we use different colors to mark three special developer accounts, which have extremely high activity metric values. They are *vscode-triage-bot, vscodebot[bot]* and *github-actions[bot]*. Among them, *vscode-triage-bot* is specifically for issue lifecycle management, which is used to prompt feature request voting status, development progress, etc., and to close expired issues. The *vscodebot[bot]* is a GitHub Apps independently developed by the VS Code team. This app has a large number of experimental features. For example, for newly opened issues, it automatically compares the content with historical issues, recommends similar issues, and automatically labels issues. The *github-actions[bot]* is the official GitHub Apps account of GitHub, dedicated to various automated collaboration tasks. It can be seen from the log that the VS Code team only introduced the bot at the end of March 2020. At present, the functional distinction of the three robots is not clear, but it is certain that VS Code has been working hard to use various automated means for project management to handle larger-scale collaboration.

The developers have formed some collaborative relationships, which are denoted in the graph as areas with varying density of nodes and edges that gather in large clusters in the collaborative network. These clusters are formed because of the issues generating the most concern in the VS Code repository.

Take the two largest clusters in the peripheral area for example. The huge cluster at the bottom center of the network diagram relates to issue #52855. This refers to the problem that VS Code in the Windows operating system will disappear if Windows is shutting down during a VS Code update. This issue has been raised since June 2018, and it has not yet been resolved. The latest comment appears on March 17, 2020. The cluster located at the top center relates to issue #108447. This issue, which was raised on October 10, 2020, is about the latest VS Code update causing code formatting problems. In a short period of time, a large number of developers entered the discussion. Even 45 days after the issue was closed by community maintainers, a large number of developers were still discussing the issue.

Through the collaborative relationship network, we can easily observe the core developers in the community and discover the hot spots in the community through algorithms and visualization. Even people who are not familiar with the repository can quickly mine in-depth information through this tool.

Description 4.1: Construction method of developer collaboration network

The developer collaboration network is based on the definition of developer activity metric in Description 2.1, and is constructed through the collaborative relationship of multiple developers in a repository. The specific construction process is to refine the calculation of the annual activities of each developer to the specific issue/PR, and at the same time, define the collaborative relationship between developers who are active on the same issue/PR, with the strength of the collaborative relationship being related to the two developers' activity metric on the issue/PR. The specific calculation method is as follows:

$$R_{ab} = \sum_{i} \frac{A_{ia} A_{ib}}{A_{ia} + A_{ib}}$$

where: $A_{ia} A_{ib}$ are the activity metric of developer a and developer b on issue/PR i respectively. The calculation method follows the activity metric calculation method in Description 2.1. R_{ab} is the degree of collaboration between developer a and developer b on the repository. That is to say, the degree of collaboration between two developers in the repository is the sum of the harmonic mean of their activity metric on all jointly active issues/PRs. The construction method is similar to that of OpenGalaxy's project collaboration network.

5. Star of the month

5.1. Evaluation method of the monthly star

In addition to top-level projects, some projects on GitHub have received a lot of attention from developers in a short period of time. These projects may be phenomenon projects, may become top-level projects in the future, or may relate to social hot spots, such as the COVID-19 pandemic. It is meaningful to discover and explain why they have received a lot of attention during a specific period. Therefore, the Star of the Month section lists the projects that have received a lot of attention from developers each month in 2020. Based on the log data, projects that received the most stars per month were screened first, of which one example for each month was manually selected.

5.2. 2020 Monthly Open Source Star

January: microsoft/playwright

Playwright is a new generation of automated testing tools released by Microsoft at the beginning of 2020. Compared with the most commonly used Selenium, it can automatically perform Chromium, Firefox, WebKit, and other mainstream browser automation operations with only one API.

February: wuhan2020/wuhan2020

wuhan2020 ranked high on the list from January to March. Given the severe situation of the COVID-19 pandemic, wuhan2020 was spontaneously organized by volunteers coming from various fields. Due to the rapid expansion of the anti-epidemic teams and the growing demands of medical supplies, it is urgently needed for a data service platform with real-time synchronization of information related to hospitals, factories, and procurement. Therefore, the project had grown at a very fast rate and was very active.

March: CSSEGISandData/COVID-19

COVID-19 is a Dashboard project created by the Johns Hopkins University Systems Science and Engineering Center (JHU CSSE) in February 2020 in the COVID-19 Environment. The number of stars reached a peak in March-April, which is consistent with the severity and duration of COVID-19, the time when the number of stars reached a peak (March 2020) was close to 2 months from the project's inception, which is thought to be related to the factors of continued attention to the global epidemic and the large scale of human and social resources involved in the project. This is a non-code project, which mainly records daily data files published by various countries and organizations, and gives a description and provides a link to the data source.

April: labuladong/fucking-algorithm

This is a leetcode-based mind development project. It was established in March. The number of stars reached three peaks respectively in April, August, and December. Compared with the three

peaks of development activity, they were exactly one month behind. Except for the months when the development activity reached its peak, the project became significantly less active. The main reason is that April, August and December were the months when students prepare for entrance exams and job interviews.

May: bradtraversy/design-resources-for-developers

Design-resources-for-developers is a design resource index. The resource (design URL) index link, which is also the readme.md file of the entire project, accounts for more than 90% of the work. 24K stars and 6K forks. It is a typical non-code project of spontaneous maintenance by design developers. Once initiated, it quickly reached top star count.

June: electronicarts/CnC_Remastered_Collection

Source code for Red Alert! In June 2020, the well-known game company EA open-sourced two games in the Command and Conquer series, one of which was Red Alert, which attracted the attention of many developers.

July: JaidedAI/EasyOCR

EasyOCR is an OCR third-party library written in python. It supports OCR recognition in more than 80 languages. Version 1.1 was released at the end of June 2020. In July, it attracted the attention of many developers.

August: geekxh/hello-algorithm

This project is a set of complete algorithm training processes for fresh learners, similar to the form of resource-sharing by network disk and website navigation directory. Because it is free, and face to a wide range of users (java language learners), with a wide coverage (algorithm basics, large-scale classics, title dictions, and algorithmic topic applications) the project earned quite a lot of stars, a number of 24.1K. It is more active during January to August, which is speculated to be related to employment.

September: cli/cli

GitHub official command line tool. At the beginning of 2020, GitHub open-sourced its own command line tool, officially releasing version 1.0 in September. The project currently has 21.6K stars and 2K forks, which has attracted many developers to use it and contribute.

October: kamranahmedse/developer-roadmap

A roadmap for web developers in 2019. These roadmaps give an outline of web development technology and guide the direction of learning; they can also be used to analyze why some tools are more suitable for use in specific situations than others. The readme.md has an excellent layout and is aimed at a large audience. Currently, the number of stars is 149K and the number of forks is 21.9k.

November: ytdl-org/youtube-dl

youtube-dl is a project to download video resources from YouTube or other video platforms (such as Youku, iQiyi, Bilibili, etc.). It has a history of more than 12 years of development since July 2008. With the modification of page component layouts or code changes of some of the video sites, the project is also being continuously updated.

December: beurtschipper/Depix

This is a GitHub open source project initiated in December 2020 that uses a matching method to eliminate mosaics. It earned nearly 7,000 stars in three days. What is interesting is that it has only 3 contributors, 3 branches, and the longest branch is the main branch. It has only been committed 15 times, but the number of forks has reached 1.8K, and the number of stars has reached 15K. There are easy-to-understand instructions and effect demonstrations with pictures and texts. Most users used WeChat, Meitu and other tools to do mosaic testing and share the feeling or give suggestions on social media. It is speculated that this project has received widespread attention because it involves the protection of privacy and security that most users are concerned about.

6. Summary and Outlook

As a data-driven visualization tool, *GitHub 2020 Digital Insight Report* mainly provides you with a new perspective on today's open source world as well as insights from intersecting with industrial experiences. Starting from this annual report, we will also maintain the work as an open source project, shorten the release cycle, and provide professional consulting services on demand.

If you find any data errors or omissions, please submit an Issue or PR to GitHub. The text of this report adopts the CC-BY-4.0 license agreement, and the project address is:
https://github.com/X-lab2017/github-analysis-report.

7. Acknowledgements

GitHub 2020 Digital Insights Report was initiated by X-lab, planned by the Allumos open source technology media, and jointly completed with East China Normal University, Kaiyuanshe, Shanghai Open Source Information Technology Association, along with many other scientific research institutions and open source community institutes.

Contributors to this report are:

Shengyu (Frank) Zhao, Wei (Will) Wang, Tianyi Zhou, Zhenjie Weng, Haoyue Wang, Xiaoya Xia, Xiangning Zhu, Ming Yang, Zexin Ning, Haiming Lin, Fuzheng Wang, Jingben Shi, Zehua Lou, Yeming Gu, Xue (Xander) Wu, Jia (Kate) Yang, Siying (Mabel) Li.

Contributors to the English version of the report are:

Xue (Xander) Wu, Puyu (Paul) Wang, Xiaoya Xia, Shengyu (Frank) Zhao, Xiaotian Dai, Siying (Mabel) Li, Yu (Atena) Chen. A special word of thanks to Aleksey Zaitsev (WeChat: vistal-media) for his invaluable help with editing.

We welcome more open source enthusiasts to join us and jointly promote the development of open source in the world.

GitHub 2020 Digital Insight Report

Online report

Allumos