No.89: 开源社区对比研究

任务介绍

- 任务标签：GitHub 社区数据分析，GitHub 社区社会学研究
- 任务描述：选取 GitHub上包含 PaddlePaddle/Paddle 社区在内的 10 个国内外优秀开源社区（不需要局限在人工智能领域，star 数 10k 以上），依据 GitHub 公开数据、社区研究论文、第三方分析报告等资料，对 10 个开源社区做横向对比研究。开放性任务。

开源社区选取原则

参照《中国开源发展研究分析2022》，我们选取了包含 PaddlePaddle/Paddle 社区在内的 10 个国内外优秀开源社区，它们分别是：

- PaddlePaddle/Paddle
- ant-design/ant-design
- pingcap/tidb
- apache/echarts
- taosdata/TDengine
- apache/flink
- microsoft/vscode
- flutter/flutter
- kubernetes/kubernetes
- pytorch/pytorch

社区日志数据来源

日志数据取自于 OpenDigger 项目的 ClickHouse sample data，在确定好待分析的 10 个社区项目之后，参考 Create sample data 教程自行导出了十个社区项目的数据

repo_id	repo_name	repo_id	repo_name
65711522	PaddlePaddle/Paddle	20587599	apache/flink
34526884	ant-design/ant-design	41881900	microsoft/vscode
41986369	pingcap/tidb	31792824	flutter/flutter
9185792	apache/echarts	20580498	kubernetes/kubernetes
196353673	taosdata/TDengine	65600975	pytorch/pytorch

SQL 脚本内容如下：

```
SELECT * FROM github_log.events WHERE repo_id in (65711522, 34526884, 41986369, 9185792, 196353673,
20587599, 41881900, 31792824, 20580498, 65600975)
```

最终导出的数据集为 data.tar.gz，总共包含 `7477602` 条日志数据，数据集中最早的一条日志数据的时间 `2015-01-01 01:50:31`，最新的一条日志数据的时间为 `2022-09-10 03:58:02`，

注：本报告中的「开源社区数据分析」采用的数据集为 data.tar.gz

社区基本信息介绍

PaddlePaddle/Paddle

项目简介

飞桨 (PaddlePaddle) 以百度多年的深度学习技术研究和业务应用为基础，是中国首个自主研发、功能完备、 开源开放的产业级深度学习平台，集深度学习核心训练和推理框架、基础模型库、端到端开发套件和丰富的工具组件于一体。目前，飞桨累计开发者 477 万，服务企业 18 万家，基于飞桨开源深度学习平台产生了 56 万个模型。飞桨助力开发者快速实现 AI 想法，快速上线 AI 业务。帮助越来越多的行业完成 AI 赋能，实现产业智能化升级。

贡献者的构成

PaddlePaddle/Paddle 的 Github 项目截止 2022 年 9 月 10 日共 619 个贡献者，其中包括 2 名提交 1000+ commits（reyoung, jacquesqiao）和 10 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，PaddlePaddle/Paddle 共计发布了 56 个 releases，其中：

- 第一个 releases（v0.8.0beta.0 ）的发布时间是 2016 年 8 月 31 日
- 最新一个 releases（v2.3.2 ）的发布时间是 2022 年 8 月 16 日

ant-design/ant-design

项目简介

Ant Design 是一套企业级 UI 设计语言和 React 组件库，主要用于研发企业级中后台产品。Ant Design 的特性包括提炼自企业级中后台产品的交互语言和视觉风格、开箱即用的高质量 React 组件、使用 TypeScript 开发，提供完整的类型定义文件、全链路开发和设计工具体系、数十个国际化语言支持、深入每个细节的主题定制能力。

贡献者的构成

ant-design/ant-design 的 Github 项目截止 2022 年 9 月 10 日共 1712 个贡献者，其中包括 3 名提交 1000+ commits（afc163, zombieJ, benjycui, yesmeck）和 4 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，ant-design/ant-design 共计发布了 507 个 releases，其中：

- 第一个 releases（v0.7.0 ）的发布时间是 2015 年 7 月 21 日
- 最新一个 releases（v4.23.1 ）的发布时间是 2022 年 9 月 9 日

pingcap/tidb

项目简介

TiDB 是 PingCAP 公司自主设计、研发的开源分布式关系型数据库，是一款同时支持在线事务处理与在线分析处理 (Hybrid Transactional and Analytical Processing, HTAP) 的融合型分布式数据库产品，具备水平扩容或者缩容、金融级高可用、实时 HTAP、云原生的分布式数据库、兼容 MySQL 5.7 协议和 MySQL 生态等重要特性。目标是为用户提供一站式 OLTP (Online Transactional Processing)、OLAP (Online Analytical Processing)、HTAP 解决方案。TiDB 适合高可用、强一致要求较高、数据规模较大等各种应用场景。

贡献者的构成

pingcap/tidb 的 Github 项目截止 2022 年 9 月 10 日共 779 个贡献者，其中包括 2 名提交 900+ commits（tiancaiamao, coocood）和 6 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，pingcap/tidb 共计发布了 153 个 releases，其中：

- 第一个 releases（v1.0.0 GA ）的发布时间是 2017 年 10 月 16 日
- 最新一个 releases（v6.1.1 ）的发布时间是 2022 年 9 月 1 日

apache/echarts

项目简介

Echarts 是一个使用 JavaScript 实现的开源可视化库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE9/10/11，Chrome，Firefox，Safari等），底层依赖矢量图形库 ZRender，提供直观，交互丰富，可高度个性化定制的数据可视化图表。

贡献者的构成

apache/echarts 的 Github 项目截止 2022 年 9 月 10 日共 174 个贡献者，其中包括 3 名提交 1000+ commits（pissang, 100pah, kener）和 4 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，apache/echarts 共计发布了 104 个 releases，其中：

- 第一个 releases（v1.0.0 ）的发布时间是 2013 年 6 月 3 日
- 最新一个 releases（v5.3.3 ）的发布时间是 2022 年 6 月 14 日

taosdata/TDengine

项目简介

TDengine 是一款开源、云原生的时序数据库，专为物联网、工业互联网、金融、IT 运维监控等场景设计并优化。它能让大量设备、数据采集器每天产生的高达 TB 甚至 PB 级的数据得到高效实时的处理，对业务的运行状态进行实时的监测、预警，从大数据中挖掘出商业价值。而且除时序数据库功能外，它还提供缓存、数据订阅、流式计算等功能，最大程度减少研发和运维的复杂度，且核心代码，包括集群功能全部开源（开源协议，AGPL v3.0）。

贡献者的构成

taosdata/TDengine 的 Github 项目截止 2022 年 9 月 10 日共 132 个贡献者，其中包括 4 名提交 1000+ commits（guanshengliang, hjxilinx, hzcheng, dapan1121）和 16 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，taosdata/TDengine 共计发布了 67 个releases，其中：

- 第一个 releases（v1.6.1.7 ）的发布时间是 2019 年 8 月 26 日
- 最新一个 releases（v3.0.1.0 ）的发布时间是 2022 年 9 月 7 日

apache/flink

项目简介

Apache Flink是由 Apache 软件基金会开发的开源流处理框架，其核心是用 Java 和 Scala 编写的分布式流数据流引擎。Flink以数据并行和管道方式执行任意流数据程序，Flink的流水线运行时系统可以执行批处理和流处理程序。此外，Flink的运行时本身也支持迭代算法的执行。Flink 能在所有常见集群环境中运行，并能以内存速度和任意规模进行计算。

贡献者的构成

apache/flink 的 Github 项目截止 2022 年 9 月 10 日共 1065 个贡献者，其中包括 4 名提交 1000+ commits（zentol, tillrohrmann, StephanEwen, aljoscha）和 9 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，apache/flink 共计发布了 206 个releases，其中：

- 第一个 releases（v0.4 ）的发布时间是 2014 年 1 月 10 日
- 最新一个 releases（v1.14.6-rc2 ）的发布时间是 2022 年 9 月 9 日

microsoft/vscode

项目简介

Visual Studio Code（简称 VS Code）是一款由微软开发且跨平台的免费源代码编辑器。VS Code 使用 Monaco Editor 作为其底层的程式码编辑器。

微软在2015年4月29日举办的 Build 2015大会上公布了 Visual Studio Code 的开发计划；同日，其预览版本发布。2015年11月18日，Visual Studio Code 在 GitHub 上开源，同时宣布将支持扩展功能。2016年4月14日，Visual Studio Code 正式版发布。

在2019年的 Stack Overflow 组织的开发者调查中，Visual Studio Code 被认为是最受开发者欢迎的开发环境。据调查，87317名受访者中有50.7%的受访者声称正在使用 Visual Studio Code。

贡献者的构成

vscode 的 Github 项目截止 2022 年 9 月 10 日共 1693 个贡献者，其中包括 16 名提交 1000+ commits（liggitt, wojtek-t, deads2k, sttts, lavalamp, thockin, brendandburns）和 24 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，microsoft/vscode 共计发布了 78 个releases，其中：

- 第一个 releases（v0.10.1 ）的发布时间是 2015 年 11 月 17 日
- 最新一个 releases（v1.71.0 ）的发布时间是 2022 年 9 月 2 日

flutter/flutter

项目简介

Flutter 是 Google 开源的构建用户界面（UI）工具包，帮助开发者通过一套代码库高效构建多平台精美应用，支持移动、Web、桌面和嵌入式平台。Flutter 开源、免费，拥有宽松的开源协议，适合商业项目。Flutter已推出稳定的2.0版本。Flutter第一个版本支持Android操作系统，开发代号称作“Sky”。 它于2015年4月的Flutter开发者会议上被公布，宣称其目标为实现 120FPS 的渲染性能。在上海Google Developer Days的主题演讲中，Google宣布了Flutter Release Preview 2，这是Flutter 1.0之前的最后一个重要版本。2018年12月4日，Flutter 1.0在Flutter Live活动中发布，是该框架的第一个“稳定”版本。2019年12月11日，在Flutter Interactive活动上发布了Flutter 1.12，宣布Flutter是第一个为环境计算设计的UI平台。2022年5月12日，在 Google I/O 2022 发布了 Flutter 3，正式支援了 Windows、macOS、Linux 等操作系统。

贡献者的构成

Flutter 的 Github 项目截止 2022 年 9 月 10 日共 1062 个贡献者，其中包括 4 名提交 1000+ commits（engine-flutter-autoroll, jonahwilliams, abarth, Hixie）和 10 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，Flutter 共计发布了 41 个releases，其中：

- 第一个 releases（v0.0.6 Alpha）的发布时间是 2017 年 5 月 12 日
- 最新一个 releases（v3.3.1 stable）的发布时间是 2022 年 9 月 7 日

kubernetes/kubernets

项目简介

Kubernetes 是一个可移植、可扩展的开源平台，用于管理容器化的工作负载和服务，可促进声明式配置和自动化。 Kubernetes 拥有一个庞大且快速增长的生态，其服务、支持和工具的使用范围相当广泛。

Kubernetes（在希腊语意为“舵手”或“驾驶员”）由Joe Beda、Brendan Burns和Craig McLuckie创立，并由其他谷歌工程师，包括Brian Grant和Tim Hockin等进行加盟创作，并由谷歌在2014年首次对外宣布 。该系统的开发和设计都深受谷歌的Borg系统的影响，其许多顶级贡献者之前也是Borg系统的开发者。在谷歌内部，Kubernetes的原始代号曾经是Seven，即星际迷航中的Borg（博格人）。Kubernetes标识中舵轮有七个轮辐就是对该项目代号的致意。

贡献者的构成

kubernetes 的 Github 项目截止 2022 年 9 月 10 日共 3230 个贡献者，其中包括 7 名提交 1000+ commits（liggitt, wojtek-t, deads2k, sttts, lavalamp, thockin, brendandburns）和 15 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，kubernetes 共计发布了 549 个releases，其中：

- 第一个 releases（v0.4 ）的发布时间是 2014 年 10 月 15 日
- 最新一个 releases（v1.25.0 ）的发布时间是 2022 年 8 月 24 日

pytorch/pytorch

项目简介

Pytorch 是一个开源的端到端的机器学习框架，由 Meta 的 AI 研究小组创建。其作为一个 Python 软件包，通过一个用户友好的前端、分布式训练以及一个 Python 库的生态系统，加速了从研究原型到生产部署的路径，实现了快速、灵活的实验和高效的生产。目前被广泛应用于学术界和工业界，而随着 Caffe2 项目并入Pytorch， Pytorch开始影响到TensorFlow在深度学习应用框架领域的地位。它提供了两个主要功能：

- 张量计算（如NumPy），具有强大的GPU加速能力
- 建立在基于磁带的自控系统（动态反向传播机制）上的深度神经网络

2022 年 9 月 12 日，全球顶级非营利开源组织 Linux 基金会宣布，正式成立 PyTorch 基金会。开源 Python 机器学习库——PyTorch，将从 Meta 转移到 Linux 基金会，并将在新成立的 PyTorch 基金会下运作。

贡献者

据 PYTORCH GOVERNANCE 团队所述，PyTorch 团队拥有近 100 名核心成员，来自 Meta 内部和外部，和众多开源贡献者及维护人员。Pytorch 的 Github 项目截止 2022 年 9 月 10 日共 2420 个贡献者，其中包括 5 名提交 1000+ commits（ezyang, zou3519,gchanan, malfet, jerryzh168）和 15 名提交 500+ commits 的贡献者。

版本发布

截止 2022 年 9 月 10 日，Pytorch 共计发布了 41 个releases，其中：

- 第一个 releases（v0.1.1 alpha-1）的发布时间是 2016 年 9 月 1 日
- 最新一个 releases（v1.21.1 Pytorch 1.12.1）的发布时间是 2022 年 8 月 6 日

开源社区数据分析

前期准备

```
from open_digger import openDigger
import sys
import os
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import numpy as np
baseDir = os.path.dirname(os.getcwd())
sys.path.append(os.path.join(baseDir, 'src'))
sys.path.append(os.path.join(baseDir, 'src', 'metrics'))
sys.path.append(os.path.join(baseDir, 'src', 'db'))
db_driver = openDigger().driver()
_clickhouse = db_driver.clickhouse
_neo4j = db_driver.neo4j

repo_id_name = dict()
repo_id_name['65711522'] = "PaddlePaddle/Paddle"
repo_id_name['34526884'] = "ant-design/ant-design"
repo_id_name['41986369'] = "pingcap/tidb"
repo_id_name['9185792'] = "apache/echarts"
repo_id_name['196353673'] = "taosdata/TDengine"
repo_id_name['20587599'] = "apache/flink"
repo_id_name['41881900'] = "microsoft/vscode"
repo_id_name['31792824'] = "flutter/flutter"
repo_id_name['20580498'] = "kubernetes/kubernetes"
repo_id_name['65600975'] = "pytorch/pytorch"
repo_id_list = list(map(lambda x : int(x), repo_id_name.keys()))
repo_name_list = list(map(lambda x : str(x), repo_id_name.values()))
```

```
startYear = 2015
endYear = 2022
```

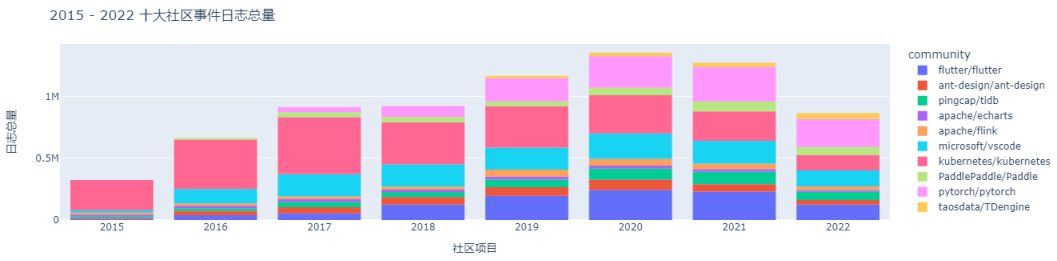
```
df = _clickhouse.queryDataframe(
    'DESC table github_log.events'
)
df
```

	name	type	default_type	default_expression	comment	codec_express
0	id	UInt64				
1	type	Enum8('CommitCommentEvent' = 1, 'CreateEvent' ...				
2	action	Enum8('added' = 1, 'closed' = 2, 'created' = 3...				
3	actor_id	UInt64				
4	actor_login	LowCardinality(String)				
...	...	...	...	...	...	...
127	commit_comment_path	String				
128	commit_comment_position	String				
129	commit_comment_line	String				
130	commit_comment_created_at	Nullable(DateTime)				
131	commit_comment_updated_at	Nullable(DateTime)				

132 rows × 7 columns

社区日志总量比较

```
df = _clickhouse.queryDataframe(
    '''
    SELECT repo_id, COUNT() AS count, formatDateTime(created_at, '%Y') AS year
    FROM github_log.events
    GROUP BY year,repo_id
    ORDER BY year,count
    '''
)
repo_names = list(map(lambda x: repo_id_name.get(str(x)),df.get('repo_id')))
df['community'] = repo_names
fig = px.histogram(
    df,
    x="year",
    y="count",
    color="community"
)
fig.update_layout(
    title="{} - {} 十大社区事件日志总量 ".format(df['year'][0], df['year'][len(df['year']) - 1]),
    xaxis_title='社区项目',
    yaxis_title='日志总量',
)
fig.show()
```

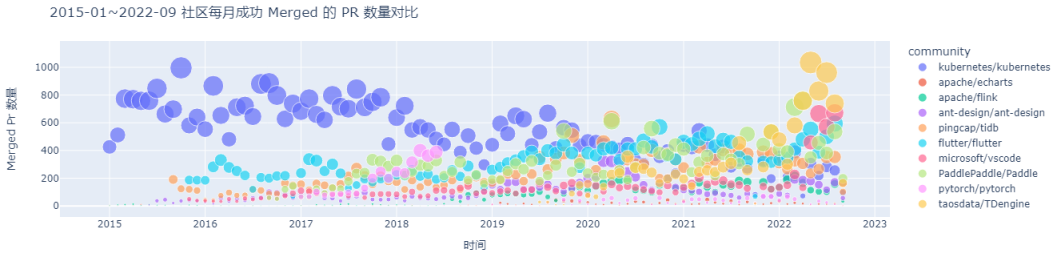


从宏观角度分析，对于各个社区的日志总数这个指标而言，日志总数越多，一定程度上可以反映社区的总体活跃情况以及受欢迎程度。

每月成功 Merge 的 PR 数量

```
df = _clickhouse.queryDataframe(
    '''
    SELECT repo_id,COUNT(id) AS record_num, formatDateTime(created_at, '%Y-%m') AS year_month
    FROM github_log.events
```

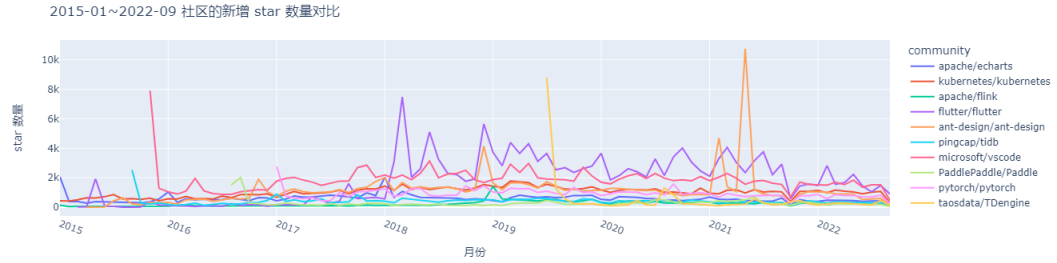
```
    WHERE (type='PullRequestEvent' AND action='closed' AND pull_merged=1)
    GROUP BY repo_id, year_month
    ORDER BY year_month, record_num
    DESC
    '''
)
repo_names = list(map(lambda x: repo_id_name.get(str(x)),df.get('repo_id')))
df["community"] = repo_names
fig = px.scatter(
    df,
    x="year_month",
    y="record_num",
    color="community",
    size = "record_num",
    size_max = 20
 )
fig.update_layout(
    title=" {}~{} 社区每月成功 Merged 的 PR 数量对比  ".format(df['year_month'][0], df['year_month']
[len(df['year_month']) - 1]),
    xaxis_title='时间',
    yaxis_title='Merged Pr 数量',
)
fig.show()
```



从每月社区中成功合并 PR 数量的变化来看，可以发现 kubernetes 在 2019 年 11 月之前遥遥领先其他社区，除此之外，flutter、Paddle、tidb 也同样名列前茅，值得注意的是新兴的物联网时序数据库 TDengine 也不容小觑

每月新增 Star 的数量

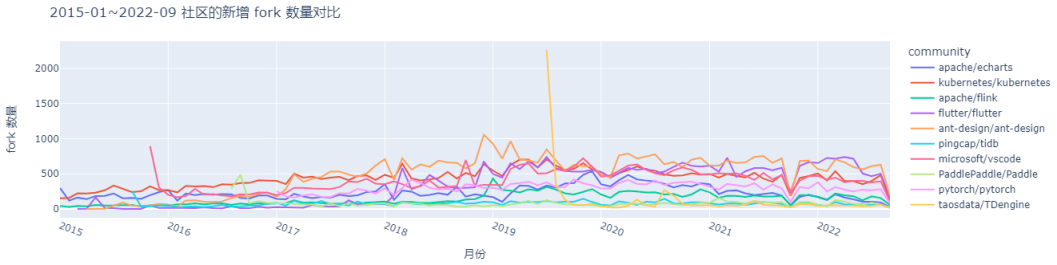
```
df = _clickhouse.queryDataframe(
    '''
    SELECT repo_id, countIf(type='WatchEvent') AS stars, formatDateTime(created_at, '%Y-%m') AS year_month
    FROM github_log.events
    GROUP BY repo_id, year_month
    ORDER BY year_month, stars
    DESC
    '''
)
repo_names = list(map(lambda x: repo_id_name.get(str(x)),df.get('repo_id')))
df['community'] = repo_names
fig = px.line(
    df,
    x="year_month",
    y="stars",
    color="community",
)
fig.update_layout(
    title=" {}~{} 社区的新增 star 数量对比  ".format(df['year_month'][0], df['year_month'][len(df['year_month'])
- 1]),
    xaxis_title='月份',
    yaxis_title='star 数量',
    xaxis_tickangle=20
)
fig.show()
```



从社区每月新增的 star 数量来看，大部分社区每月新增的 star 数量都是相差不大的，变化情况较为平稳，但是 flutter 项目每月新增 star 数量变化则相对波动较大，除此之外，值得注意的是大部分社区在 GitHub 平台上开源的前几个月里的 star 增长数量非常大，随后才降至稳定状态，这可能与项目初期的大力宣传有关。

每月新增 Fork 的数量

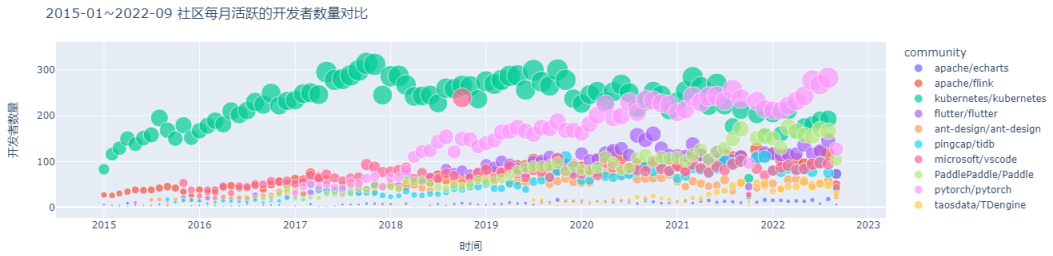
```
df = _clickhouse.queryDataframe(
    '''
    SELECT repo_id, countIf(type='ForkEvent') AS forks, formatDateTime(created_at, '%Y-%m') AS year_month
    FROM github_log.events
    GROUP BY repo_id, year_month
    ORDER BY year_month, forks
    DESC
    '''
)
repo_names = list(map(lambda x: repo_id_name.get(str(x)),df.get('repo_id')))
df['community'] = repo_names
fig = px.line(
    df,
    x="year_month",
    y="forks",
    color="community",
)

fig.update_layout(
    title=" {}~{} 社区的新增 fork 数量对比  ".format(df['year_month'][0], df['year_month'][len(df['year_month'])
- 1]),
    xaxis_title='月份',
    yaxis_title='fork 数量',
    xaxis_tickangle=20
)
fig.show()
```



fork 数量可以体现有潜在贡献意愿的开发者数量，相较于社区每月新增的 star 数量，大部分社区每月新增的 fork 数量是存在较大波动的，我们可以发现随着时间的变化，总体来说，各个社区的 fork 新增量都在提高，比较有意思的是在 2021 年 10 月份，上述社区的 fork 新增量都不约而同地有较大幅度的减少。在询问 OpenDigger 的 maintainer 之后得知，这是因为在 2021 年 10 月份的部分日志数据有丢失。最终导致了在 2021 年 10 月份，每个社区的 star 和 fork 新增量都有较大程度的减少。

每月在仓库中活跃的不同开发者总数

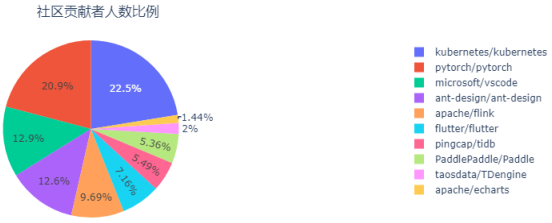
```
df = _clickhouse.queryDataframe(
    '''
    SELECT repo_id, COUNT(DISTINCT actor_id) AS actor_count, formatDateTime(created_at, '%Y-%m') AS year_month
    FROM github_log.events
    WHERE type = 'PullRequestEvent'
    GROUP BY repo_id, year_month
    ORDER BY year_month, actor_count
    '''
)
df
repo_names = list(map(lambda x: repo_id_name.get(str(x)),df.get('repo_id')))
df['community'] = repo_names
fig = px.scatter(
    df,
    x="year_month",
    y="actor_count",
    color="community",
    size="actor_count",
    size_max=20
)
fig.update_layout(
    title=" {}~{} 社区每月活跃的开发者数量对比  ".format(df['year_month'][0], df['year_month']
[len(df['year_month']) - 1]),
    xaxis_title='时间',
    yaxis_title='开发者数量',
)
fig.show()
```



开源社区以人为本，从社区每月活跃的开发者数量对比图中可以发现， kubernetes 在活跃开发者的数量上完全压制住了其他的社区，这也得益于云原生成为驱动业务发展的动力引擎，而 kubernetes 则开启了整个云原生时代。其次较高的就是 pytorch，flutter 和 paddle，因此我们可以发现 PaddlePaddle 社区每月活跃的开发者人数也是比较高的。

各个社区贡献者的数量占比

```
df = _clickhouse.queryDataframe(
    '''
    SELECT repo_id, COUNT(DISTINCT(arrayJoin(push_commits.name))) AS contributor_count
    FROM github_log.events
    GROUP BY repo_id
    ORDER BY contributor_count
    DESC
    '''
)
repo_names = list(map(lambda x: repo_id_name.get(str(x)), df.get('repo_id')))
df['community'] = repo_names
fig = px.pie(df, names= "community",values="contributor_count")
fig.update_layout(
    title={
        "text":"社区贡献者人数比例",
        "y":0.95,
        "x":0.49,
        "xanchor":"center",
        "yanchor":"top"
    }
)
fig.show()
```

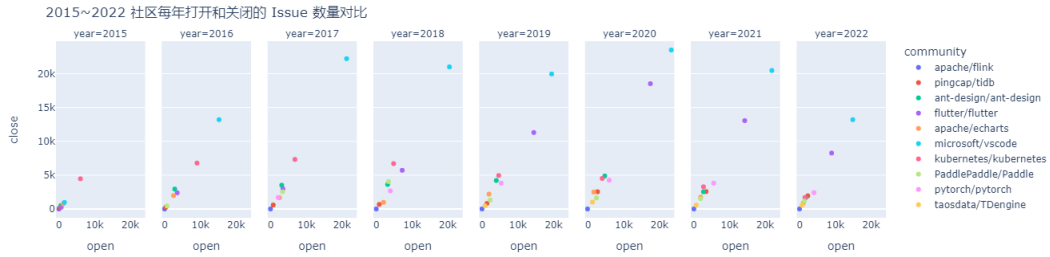


从社区贡献人数来看，kubernetes 和 pytorch 作为顶级项目，它们的贡献者人数最多，分别占据 22.5% 和 20.9%，在这一方面，PaddlePaddle 的贡献者人数相较而言就比较少，后续应该加强社区的推广

每年打开 Issue 和关闭 Issue 的个数

```
df = _clickhouse.queryDataframe(
    '''
    SELECT repo_id, countIf(type = 'IssuesEvent' AND action = 'opened') AS open, countIf(type = 'IssuesEvent'
AND action = 'closed') AS close,formatDateTime(created_at, '%Y') AS year
    FROM github_log.events
    GROUP BY repo_id, year
    ORDER BY year, open, close
    '''
)
df_open = df.loc[:,['repo_id','open','close','year']]
repo_names = list(map(lambda x: repo_id_name.get(str(x)), df.get('repo_id')))
df_open['community'] = repo_names
fig = px.strip(
    df_open,
    x="open",
    y="close",
    color="community",
    facet_col="year"
)
fig.update_layout(
    title=" {}~{} 社区每年打开和关闭的 Issue 数量对比  ".format(df['year'][0], df['year'][len(df['year']) - 1]),
    xaxis_title='open',
    yaxis_title='close',
)
```

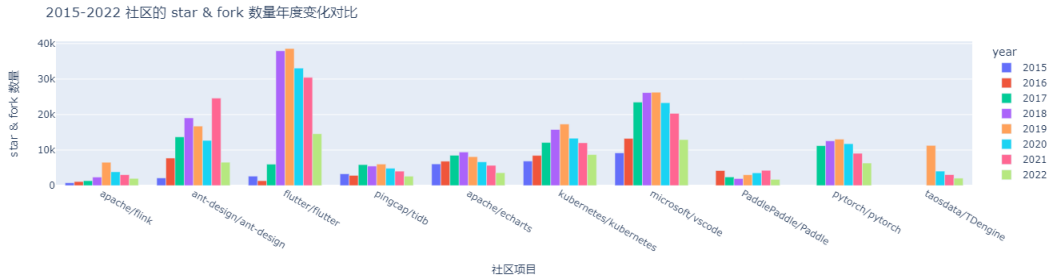
```
fig.show()
```



从社区每年打开和关闭 Issue 的数量情况来看，我们可以发现 flink 的数据都是 0，这其实是因为 flink 是不完全使用 GitHub 功能的项目，它虽然在 GitHub 上进行协作，但事实上并不会在 GitHub 上利用 Issue 进行需求的提交与追踪，而会使用例如 JIRA 一类的工具，尤其是 Apache 基金会下的众多项目，都是这种情况。vscode 的打开 issue 数量和关闭 issue 数量都遥遥领先于其他社区项目，从而可以得出 vscode 社区的功能迭代速度和问题解答速度都是非常快的

每年各个社区 Star & Fork 数量对比

```
df = _clickhouse.queryDataframe(
    '''
    SELECT repo_id,SUM(CASE WHEN type = 'WatchEvent' THEN 1 ELSE 0 END) AS stars, SUM(CASE WHEN type =
'ForkEvent' THEN 1 ELSE 0 END) AS forks, formatDateTime(created_at, '%Y') AS year
    FROM github_log.events
    WHERE type = 'WatchEvent' OR type = 'ForkEvent'
    GROUP BY repo_id, year
    ORDER BY  year,stars,forks
    DESC
    '''
)
repo_names = list(map(lambda x: repo_id_name.get(str(x)),df.get('repo_id')))
fig = px.bar(
    df,
    x=repo_names,
    y="stars",
    color="year",
    hover_data=['forks'],
    barmode="group"
)
fig.update_layout(
    title="{}-{} 社区的 star & fork 数量年度变化对比  ".format(df['year'][0], df['year'][len(df['year']) - 1]),
    xaxis_title='社区项目',
    yaxis_title='star & fork 数量',
)
fig.show()
```

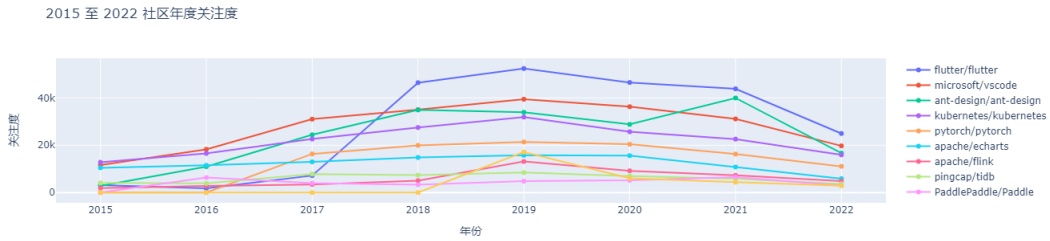


从社区的的 star 和 fork 数量年度变化来看，大部分社区的 star 和 fork 数量都先变高，随后再降低，而 PaddlePaddle 社区则是稳步上升（除去 2022 年，因为 2022 年的数据只计算到 9 月份）

项目关注度比较

```
startYear = 2015; startMonth = 1; endYear = 2022; endMonth = 12; x = []
for y in range(startYear,endYear+1): x.append(str(y))
data = openDigger().index().attention().getAttention({'repoIds:': repo_id_list, 'startYear':startYear,
'startMonth':startMonth, 'endYear':endYear, 'endMonth':endMonth,  'groupTimeRange': 'year', 'limit': 10})
fig = openDigger().render.Figure()
plotDatas = list(map(lambda row: {'x':x,'y':row.get('attention'),'name':row.get('name')}, data))
for plotData in plotDatas:
    fig.add_trace(
        openDigger().render.Scatter(
            x=plotData.get('x'),
            y=plotData.get('y'),
            mode="markers+lines",
            name=plotData.get('name')
    ))
fig.update_layout(
```

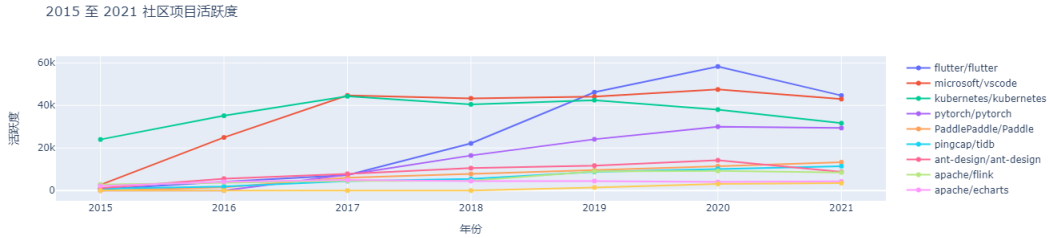
```
    title="{} 至 {} 社区年度关注度".format(startYear, endYear),
    xaxis=dict(type='category'),
    xaxis_title='年份',
    yaxis_title='关注度',
)
fig.show()
```



关注度指标是由 X-lab 团队开发的一个基本统计指标。该指标使用 GitHub 日志中的 `WatchEvent` 和 `ForkEvent`，此类事件可以表示仓库在一段时间内的被关注情况，但不会对仓库有实质性贡献。可以发现 TDengine 在 2018 年之前的关注度为 0，其实这是由于 TDengine 是在 2019 年才开始在 GitHub 上开源。查看源码中的关注度计算方式，可以发现一段时间内的关注度等于 `WatchEvent` 的数量加上 2 倍的 `ForkEvent` 数量。

项目活跃度比较

```
startYear = 2015; startMonth = 1; endYear = 2021; endMonth = 12; x = []
for y in range(startYear, endYear + 1): x.append(str(y))
data = openDigger().index().activity().getRepoActivity({'repoIds': repo_id_list, 'startYear': startYear,
'startMonth': startMonth, 'endYear': endYear, 'endMonth': endMonth,  'groupTimeRange': 'year', 'limit': 10})
fig = openDigger().render.Figure()
plotDatas = list(map(lambda row: {'x':x,'y':row.get('activity'),'name':row.get('name')}, data))
for plotData in plotDatas:
    fig.add_trace(
        openDigger().render.Scatter(
            x=plotData.get('x'),
            y=plotData.get('y'),
            mode="markers+lines",
            name=plotData.get('name')
    ))
fig.update_layout(
    title="{} 至 {} 社区项目活跃度".format(startYear, endYear),
    xaxis=dict(type='category'),
    xaxis_title='年份',
    yaxis_title='活跃度',
)
fig.show()
```



项目活跃度由 X-lab 团队设计的一个指标，它是基于 GitHub 行为数据的加权活跃度算法，具体的计算方式为：$A_d = \sum w_i c_i$ 其中的 A_d 为开发者活跃度，而 c_i 为上述五种行为事件由该开发者触发的发生次数，w_i 为该行为事件的加权比例。按照一个简单的价值评判，我们可以将这个值设置为 1 - 5，即 Issue 评论每个计 1 分、发起 Issue 每个计 2 分、发起 PR 每个计 3 分、PR 上的代码 review 评论每个计 4 分、PR 合入一个计 2 分。在计算出每个开发者的活跃度后，可以通过一种加权和的方式来计算项目的活跃度，之前给出的方式是：$A_r = \sum \sqrt{A_d}$ 即项目的活跃度为所有开发者活跃度的开方和，这里开方是为了降低核心开发者过高的活跃度带来的影响。从图中可以看出，活跃度前五的项目分别是 flutter、vscode、kubernetes、pytorch、ant-design，值得注意的是，PaddlePaddle 隐隐有超过 ant-design 的趋势。

Bus Factor

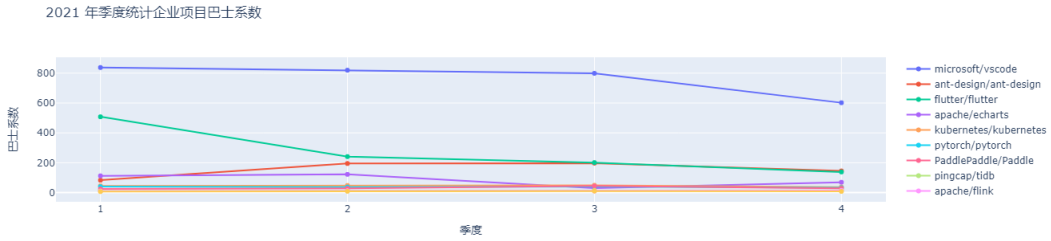
Bus factor（巴士系数） 是一个来自 CHAOSS 的指标，请参考 https://chaoss.community/metric-bus-factor/ 。

Bus factor 有如下一些额外选项

- withBot：`true` 或 `false`，是否包含 GitHub Apps 账号的贡献。默认：false。
- percentage：用于确定巴士系数开发者的贡献百分比。默认：0.5。
- by：使用何种方式进行计算，可选为 `commit`, `change request`, `activity`。默认：`activity`。

```
startYear = 2021; startMonth = 1; endYear = 2021; endMonth = 12;x = []
for y in range(1,5): x.append(str(y))
```

```
data = openDigger.metric().chaoss().busFactor({ 'repoIds': repo_id_list, 'startYear':startYear,
'startMonth':startMonth, 'endYear':endYear, 'endMonth':endMonth, 'groupTimeRange': 'quarter', 'limit': 10,
'options': { 'withBot': False, 'percentage': 0.2 } })
fig = openDigger().render.Figure()
plotDatas = list(map(lambda row: {'x':x,'y':row.get('bus_factor'),'name':row.get('name')},filter(lambda row:
row.get('name') != 'Others', data)))
for plotData in plotDatas:
    fig.add_trace(
        openDigger().render.Scatter(
            x=plotData.get('x'),
            y=plotData.get('y'),
            mode="markers+lines",
            name=plotData.get('name')
    ))
fig.update_layout(
    title="2021 年季度统计企业项目巴士系数",
    xaxis=dict(type='category'),
    xaxis_title='季度',
    yaxis_title='巴士系数',
)
fig.show()
```



巴士系数是软件开发中关于软件项目成员之间信息集中及共享度的一个衡量指标。一个项目至少失去若干关键成员的参与（“被巴士撞了”，指代职业和生活方式变动、婚育、意外伤亡等任意导致缺席的缘由）即导致项目陷入混乱、瘫痪而无法存续时，这些成员的数量即为巴士系数。理论上来说，一个社区项目巴士系数越高，则说明代码的贡献越分散在多个开发者身上，可以侧面说明项目的抗风险能力越强，但同样有另一种说法是，如果巴士系数很低，但如果这些少量的开发者是由公司完全支持的，其实也可以说明项目的抗风险能力越强。阅读源码后可以发现，巴士系数是根据 git commit的 数量或者 Pull Request 被合并的数量或者 activity 来计算的，默认是采用 activity，即活跃度，可以设置 thredhold，默认是计算活跃度排在前 50% 的贡献者数量作为巴士系数，而我在本次分析中是将阈值设置为 20%，即活跃度排在前 20% 的开发者数量作为巴士系数。

开源社区社会学研究

中国开源发展背景

- 2021 年开源第一次被写入国家《“十四五”软件和信息技术服务业发展规划》中，并且指定了未来五年中国开源发展的明确目标，其中规划中要求，一是要加强开源代码安全检测，保障开源代码组件供给安全，二是要培育和壮大市场主体，加快繁荣开源生态，提高产业集聚水平，形成多元、开放、共赢、可持续的产业生态。
- 中国信息技术飞速发展为开源发展奠定了基础。目前，中国开发者超过 1000 万，是 GitHub 上第二大开发者人群国籍
- 中国开源认知程度较好：开发者对开源模式不同程度的认知超过 86%，深度了解开源模式的开发者超过 20%
- 中国整体参与开源核心人群已经从认知期进入生产期，未来有望成为全球开源的领导者

中国开源发展四大价值

- 对开发者人群：个人成长和社交平台价值
- 对企业研发团队：研发加速工具价值
- 对商业创业公司：“研发加速、营销加速、商业化加速、招人加速工具”价值
- 对中国软件产业发展：创新的推动器价值

中国开源的主要参与者与关注方向

- 中国知名开源项目技术领域分布已经非常广泛、社区、企业、基金会的生态基本形成
- 技术领域正在从优势领域数据库向操作系统和人工智能以及云原生等方向进行升级聚焦
- 开发者（互联网+非互联网）关注的技术领域相对平均，排名前三的是数据库、云原生和工具

政府侧，将培育中国开源生态纳入国家级发展计划

- 四大抓手

1. 组织 & 文化

> 大力发展国内开源基金会等开源组织，完善开源软件治理规则，普及开源软件文化
>
> 2. 基础设施
>   加快建设开源代码托管平台等基础设施
> 3. 项目与生态
>   面向重点领域布局开源项目，建设开源社区，汇聚优秀开源人才，构建开源软件生态
> 4. 国际交流
>   加强与国家开源组织交流合作，提升国内企业在全球开源体系中的影响力

- 关键成果

1. 到 2025 年建成 2~3 个有国际影响力的开源社区
2. 培育超过 10 个优质开源项目

产业侧，人才规模壮大为开源发展奠定了坚实的基础

- 伴随着中国软件和信息服务行业的快速发展和企业数字化进程的不断推进；中国软件行业从业人员规模不断壮大，而这其中最主要的构成即为开发者人群
- 2021 年中国软件行业从业人员增速达 14.91%，达 809 万人。当然这个数据依然是被低估了的，根据 InfoQ 研究中心测算，开发者人群应该已经超过 1000 万人量级，中国开发者人群已经成为世界知名开源社区 GitHub 第二大人群，仅次于美国

分析结果小结

本部分介绍横向对比的结论

- 结论1：PaddlePaddle 与国际顶级如 vscode、flutter 等开源项目相比仍有较大提升空间，一方面可以继续大力推广 PaddlePaddle 社区以吸引贡献者，另一方面可以通过各项指标数据来监控社区的状态，以便于社区治理
- 结论2：PaddlePaddle 社区的活跃度在逐年上升，相比于国内的众多项目，Paddle 活跃度名列前茅，甚至近年来已经超过 ant-design，成为国内最活跃的项目
- 结论3：相比于国外顶级开源社区，PaddlePaddle 的关注度并不算高，PaddlePaddle 未来可以依托《“十四五”软件和信息技术服务业发展规划》，大力推广和建设社区，吸引更多的开发者参与项目和社区的建设
- 结论4：社区大于代码，稳定繁荣的开源社区是激发开源价值的关键，开源社区是开发者共同贡献的产物，也是完全不同于闭源软件研发中心协作关系的集中体现。开源社区对社区协作中的安全性、开放性和多样性最为重视

参考资料

- 《中国开源发展研究分析2022》
- 《2021 中国开源年度报告》
- 《2021 中国开源发展蓝皮书》
- 《Producing Open Source Software: How to Run a Successful Free Software Project》
- 如何评价一个开源项目（一）——活跃度
- 如何评价一个开源项目（二）——协作影响力
- 如何评价一个开源项目（三）——价值流网络