Companies' Participation in OSS Development – An Empirical Study of OpenStack

Yuxia Zhang, Minghui Zhou, Audris Mockus, and Zhi Jin

Abstract—Commercial participation continues to grow in open source software (OSS) projects and novel arrangements appear to emerge in company-dominated projects and ecosystems. What is the nature of these novel arrangements? Does volunteers' participation remain critical for these ecosystems? Despite extensive research on commercial participation in OSS, the exact nature and extent of company contributions to OSS development, and the impact of this engagement may have on the volunteer community have not been clarified. To bridge the gap, we perform an exploratory study of OpenStack: a large OSS ecosystem with intense commercial participation. We quantify companies' contributions via the developers that they provide and the commits made by those developers. We find that companies made far more contributions than volunteers and the distribution of the contributions made by different companies is also highly unbalanced. We observe eight unique contribution models based on companies' commercial objectives and characterize each model according to three dimensions: contribution intensity, extent, and focus. Companies providing full cloud solutions tend to make both intensive (more than other companies) and extensive (involving a wider variety of projects) contributions. Usage-oriented companies make extensive but less intense contributions. Companies driven by particular business needs focus their contributions on the specific projects addressing these needs. Minor contributors include community players (e.g., the Linux Foundation) and research groups. A model relating the number of volunteers to the diversity of contribution, shows a strong positive association between them.

Index Terms—Open source ecosystem, software development, commercial participation, contribution extent, contribution intensity, contribution focus

1 Introduction

Open source software (OSS) ecosystems,[1] particularly large ones such as the Linux kernel, have had a tremendous impact on computing and society [1]. Numerous companies have participated in and built business models around OSS ecosystems to achieve user innovations [2], reduce R&D costs [3], or generate profits on complementary services [4].

Commercial involvement in OSS ecosystems has attracted extensive attention from industry and research communities. Initial efforts focused on the motivation, business models, strategies and actions of companies getting involved in OSS ecosystems. For example, a study of four firms involved in OSS discovered three ways that firms used to connect with OSS communities [5]: accessing development in the community in order to extend their resource base; aligning their strategy with the work in the community; and assimilating communities in order to integrate and share results. Wagstrom et al. [6] identified two types of commercial involvement from analyzing GNOME and Eclipse: community-focused company building a vibrant GNOME community and monetizing services and product-focused company relying on product revenues. While the commercial involvement brings additional resources to OSS, it does alter the motivation and participation of developers (see, e.g., [6, 7, 8, 9]) and, thus, may challenge the sustainability of the open source approach to software development. For example, it has been found that a company's full control and high intensity of involvement in an OSS ecosystem may decrease the inflow of volunteers[2] [8, 9].

Despite extensive research on commercial involvement in OSS ecosystems, it is not clear what drives the intensity of code and developer contributions (two of the most important elements in OSS development [10, 11]) by various companies nor it is clear how it may affect the contributions from other companies. In particular, what are the specific properties of individual companies that may affect their contributions to open source? Iansiti and Levien [12] show how the diversity in the business ecosystems can increase its health. Would diversity of companies contributing to an OSS project contribute to its health through, for example, the participation of volunteers? How to measure such diversity? To answer such questions, we conduct an exploratory case study of company participation in OSS [13].

We need an ecosystem with active and extensive company participation in order to answer our basic research questions related to the degree, types, and diversity of commercial contributions and their impact on volunteer participation. This ecosystem needs to 1) have a large number of companies that actively participate; 2) contain a large number of individual projects; 3) ecosystem's projects vary

• *Y. Zhang, M. Zhou, and Z. Jin are with the School of Electronics Engineering and Computer Science, Peking University and Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing 100871, China.*
A. Mockus is with the Department of Electrical Engineering and Computer Science, University of Tennessee, Min H. Kao Building, Room 613, 1520 Middle Drive, Knoxville, TN 37996-2250.
E-mail: yuxiaz@pku.edu.cn.

1. We use the term "ecosystem" to represent a group of software users, developers, organizations, artifacts, and infrastructure interacting as a system. Operationally, an ecosystem may contain one or more software projects.

2. We use the term "volunteers" to refer to developers who make contributions to OSS projects on their own (even if they might be employed), instead of participating to fulfill their obligations to a company or other organization.

with respect to the composition of companies involved. The study of an ecosystem removes some undesirable variability that may arise if we study a large number of unrelated projects by ensuring that the observed variation in outcomes is not affected by factors that are shared among ecosystem projects [13]. While the Linux kernel ecosystem has, perhaps, the largest number of companies participating, which is not easy to be separated into autonomous projects that we need to observe the variation in both company and volunteer participation. Instead we conduct an empirical study of OpenStack, one of the fast growing OSS ecosystems that is increasingly attracting scholarly attention [14, 15, 16]. OpenStack has 485 individual repositories with a large number of volunteers and 268 companies participating in the development of a recent release (the 14th release), including hardware manufacturers, software vendors, system integrators, and consultancy corporations. OpenStack allows these companies to play a role in the rapidly evolving cloud computing technology. For comparison, the first release of OpenStack in 2010 had only 17 organizations involved.

Our research goal is divided into several research questions that will guide our exploratory study. We start from measuring the degree of commercial contributions: To what extent do companies contribute to the development of OpenStack in comparison to volunteers (RQ1)? Volunteer participation is highly uneven in OSS, i.e., a small proportion of developers complete most of the work [1, 17]. Are contributions distributed unevenly among companies as well (RQ2)? Can the unique circumstances of each contributing company be grouped into a small number of interpretable contribution patterns/models (RQ3)? If so, what are these models and do they vary in terms of code and developer contributions (RQ3.1)? High diversity is associated with healthy ecosystems [12, 18, 19, 20]. Would the diversity of companies (in terms of their contribution models) be associated with the health of the projects in terms of volunteer participation (RQ3.2)?

To answer these questions, we mined the code commit history of OpenStack and analyzed the abundant online records about OpenStack, and its participating companies and individual developers. We found that companies[3] contributed approximately 90% of the commits and 80% of the developers (median value over 14 releases), playing a critical role in the development of OpenStack. However, the distribution of the contributions among companies was highly uneven, with approximately 10% of companies accomplishing 80% of work and 20% providing 80% of the developers. Our analysis suggests eight contribution models (templates of strategies and actions) used by companies: full solution oriented, specific sub-solution oriented, self-business oriented, specific services oriented, usage oriented, community oriented, development infrastructure vendors, and research oriented. We further characterized the performance of each model along three dimensions: contribution intensity, extent, and focus. OpenStack consists of hundreds of projects (dozens of project types), therefore we used code and developers that companies contribute to different projects (and project types) to measure the three dimensions. We found that companies providing full cloud solutions make intensive and extensive contributions. Companies that contribute to specific projects are either motivated by their specific business goals or are infrastructure vendors developing software that support the development of OpenStack. Users of OpenStack make extensive – but not large by volume – contributions, with the main focus on deployment tools. Community players such as the Linux Foundation, and research groups are minor contributors. We found that the increase in diversity of companies as measured by the entropy of their contribution models is associated with an increase in participation by volunteers.

Our results may shed light on shaping and sustaining OSS ecosystems as commercial participation will continue to increase in the future [21]. In particular, the participating companies may choose to employ a relevant model (such as which projects to pay attention to and how much contribution to make) to maximize their interests while at the same time balancing such private goals with the sustainability of the entire ecosystem. OSS communities, on the other hand, may use our results to understand the evolution and the status of their ecosystem and take appropriate actions if problems emerge, e.g., a project becomes dominated by a single company.

The remainder of this article is organized as follows. We outline our multi-method research approach in Section 2 and present the results in Section 3. Section 4 discusses the implications for research and practice. We address the limitations in Section 5 and review the related work in Section 6. We make a conclusion in Section 7.

2 Study Design

In this study, we employed a case study approach as the basis of our overall research strategy because it is suitable for exploratory research [13, 22], and frequently used as a standard approach to conduct empirical studies [23]. More specifically, this study combined the analysis of code commit history with an examination of the published literature and online documents. We started from selecting OpenStack as the study case, and grouping the projects of OpenStack in its technology stack into types (for understanding OpenStack context and for later analysis of commercial involvement), as described in Section 2.1. We then collected, cleaned and prepared the code commit data of OpenStack, as described in Section 2.2. To assign contributions to companies, we identified the developers working on behalf of different companies and their commits, as described in Section 2.3. The processed data were then used to quantify the extent (RQ1) and the distribution (RQ2) of companies' contributions. We collected companies' commercial objectives to extract contribution models and derived three dimensions to characterize the performance of each model (RQ3.1), as described in Section 2.4. We designed a metric to describe the diversity of contribution models, which is used to explore the association between the diversity and the number of volunteers by fitting a regression model (RQ3.2), as described in Section 2.5.

3. There are also a few universities and research institutions participating in OpenStack. For convenience, we label all the organizations as "companies."

2.1 OpenStack and Its Projects

The ongoing OSS ecosystem we studied, OpenStack, is a major open source cloud computing platform, and is currently widely used around the world [24]. OpenStack was started in July 2010 by NASA and Rackspace (an established IT web hosting company) [16]. After two years, Rackspace moved the leadership to the OpenStack Foundation, which oversees the development and construction of surrounding ecosystem.

OpenStack is a collection of OSS projects for building and managing cloud computing platforms for public, hybrid, and private clouds. It follows a six-month, time-based release cycle[4] to produce products which control large pools of computing, storage, and networking resources throughout a data center, managed through a dashboard or via API.[5] At the time of this study, OpenStack consisted of 538 git repositories with more than 560 companies and 9,500 developers involved, as shown in Table 1. We chose to investigate OpenStack among all OSS ecosystems because it: (i) implements the popular cloud technology; (ii) is a large ecosystem with hundreds of projects; (iii) has extensive commercial participation; (iv) has highly interconnected nature among many companies and volunteers [14].

TABLE 1
An Overview of OpenStack

#projects	538
#companies	564
#developers	9,532
#lines of code	79,352,235
#commits	455,825

*The statistical cut-off date is January 18, 2017.

The technology stack of OpenStack consists of various projects. Some collaborate with each other to offer a complete service, and some may have similar functions but differ somewhat in terms of the details for meeting various usage scenarios. For example, Swift and Cinder both are providing storage services but their units of storage are different.[6] Therefore, we classified the 538 projects of OpenStack into different types to achieve the following goals: 1) to have an intuitive understanding about how an OpenStack release is produced by different types of projects; 2) to use project type as an instrument to explore contribution performance of companies.

We manually looked for the documents[5] and the "readme" file of each repository and merged projects with similar functions. The first two authors performed this process independently, and conflicts were resolved by a sequence of meetings. Eventually we obtained 14 types ranging from infrastructure services, including computing, storage and networking, to development, deployment, monitoring tools as well as documents and localization, as shown in Table 2. To validate the results, we shared the 14 types with two experts who have more than five years of experience in OpenStack. We adjusted the types of six projects based on their advice.

2.2 Collecting and Filtering Data

OpenStack uses Git for version control. When a version control system (VCS) is employed in the development process of software, it tracks all changes made by the developers on the platform. Each time a new commit is made, the VCS records its information. A commit (which is most relevant to our study) consists of the author's login name, email address, the commit message and the time when it is committed. Also, by comparing the state of software between commit and its parent commit it is possible to determine the list of files modified by a commit.

We obtained commits from OpenStack Development Dashboard,[7] where Bitergia[8] collected these data from OpenStack's Git server. The time span of the dataset is from OpenStack's creation date (July 21st, 2010) until January 18, 2017,[9] covering 14 complete releases.

We cleaned the raw data for further analysis. In particular, existing literature [8, 25] indicates that there might be commits submitted by some accounts that are not used by individual humans, such as automated bots. We collected the non-human accounts identified in prior studies [25, 26, 27] and removed all commits submitted by these accounts in our dataset. Table 3 shows some typical examples of the non-human accounts, i.e., names and emails, and full list can be found in the public dataset of this work[10]. This step removed 92,829 commits leaving us with 362,996 commits remaining.

2.3 Identifying Affiliations of Developers and Commits

It has been found that companies task their employees to contribute to OSS projects with the idea of influencing decisions made by these projects [21]. To gauge such influence we quantify each company's contributions by counting developers it employed and the commits submitted by those developers. Accurately identifying developers' affiliations presents several challenges. First, developers' affiliations are not recorded in Git commits directly. Second, many of the OpenStack developers have been frequently changing their jobs [28]. We employed a multitude of techniques to obtain and validate developer affiliation at the time of each commit they made to OpenStack. These techniques were developed through search for related research literature, online documents, and communicating with a core developer of OpenStack. We ended up with a four-step process that we validated to achieve high accuracy, as described below.

Step 1. Merging developer IDs.

Each time a commit is made, author information is recorded by Git based on the credentials (full name and email) of the local Git repository where the commit is made. It is common for developers to have several alternative

4. https://docs.openstack.org/project-team-guide/release-management.html
5. https://docs.openstack.org/
6. https://www.openstack.org/software/project-navigator/
7. http://activity.openstack.org
8. http://vizgrimoire.bitergia.org/
9. After Jan 2017, Bitergia stopped updating these data for OpenStack. To keep the consistency of the data processing style, we do not add the data produced after that time.
10. https://github.com/noname2018/Commercial-Participation-in-OSS

TABLE 2
Fourteen Types of the 538 Projects in OpenStack

Type	Description	# Projects
Computing	To implement services and associated libraries regarding computing	11
Storage	To implement services, associated libraries, and protocols regarding storage functionality	8
Networking	To provide capabilities for managing dynamic host configuration protocol (DHCP), static Internet protocols, or virtual area networks	42
Deployment	To deploy OpenStack in production	200
Infrastructure of development	To provide infrastructure for the development of OpenStack	75
Orchestration	To provide interface and tools for the management of OpenStack services	57
Application services	To develop, publish, and manage various cloud-ready applications in OpenStack	17
Data analytic	To implement services and libraries about database, data processing and searching	10
Monitoring and metering	To efficiently collect, normalize and transform data produced by OpenStack services	16
Security and compliance	To deal with security and compliance problems for many purposes, such as legal requirements, customer needs, privacy considerations	16
Community build	To record, manage, and monitor the participation activities of the members of community, e.g., the number of commits contributed by each company	5
Documents	To document guides which can help users to install and use, and help contributors to participate; to document the requirements collected from all the OpenStack users.	17
Localization	To localize projects to allow OpenStack to be used in different countries/areas	2
Architecture optimization	To optimize the development architecture and share common libraries	28

*Note: There are 34 projects that were not classified owing to the lack of information.

TABLE 3
Non-human Accounts in Commits Dataset.

Name	Email
Gerrit Code Review	review@openstack.org
OpenStack Jenkins	jenkins@openstack.org
Ubuntu	ubuntu@dev-stack-alone
OpenStack Project Creator	openstack-infra@lists.openstack.org

spellings of their name and email (we refer to this combination as author ID) recorded in Git commits [29, 30, 31, 32, 33].

The information sources we use to determine developer affiliations are based on author ID. We, therefore, need to merge potentially multiple spellings of author IDs in the commits. Author identity merging is a well-recognized problem in the literature (e.g., [31, 32, 33]). Most studies use string similarity-based techniques of identifiers (typically login credentials) i.e. name, user-name and email similarity, to solve the identity problem. However, such methods can not help if the string similarity for distinct IDs used by the same developer is low or when the same ID is used by multiple developers, e.g., some using "nobody" for anonymity, which are common. This problem is very difficult and lacks comprehensive solutions [30]. To address it, we employ a novel highly-accurate machine-learning method to disambiguate developer identities [26]. This method enhances the string similarity-based techniques with developers' behavioral features that tend to be more similar if the different IDs are used by the same developer and less similar for IDs of distinct developer. More specifically, this method defines three additional measures to encode the behavioral features of developers: similarity based on files touched, similarity based on time zone of the commits, and similarity based on commit message text. The last feature is computed by using Doc2Vec algorithm [34]) that creates a vector embedding for each developer ID. These three similarity measures are computed for all pairs of developers in conjunction to string similarity for the following components of developer IDs: full name, first name, last name, email, and username (the first component of the email). We then incrementally selected a sample of 2K pairs and manually checked if the pair is likely or unlikely to represent the same developer. Two raters were used to determine whether or not the pair represents the same developer. A random sample of the manually matched set was selected for further verification by contacting the developers using email in the provided IDs and asking if the two (or more) commits that were manually determined to belong to the same developer were indeed theirs (see Step 4 below). The iterative part involved an active learning approach [35], which was used to minimize the effort to generate such large (2K) manually validated (golden) set. Initially, a smaller set of pairs was manually classified (as a match or not a match) and used to train a preliminary classifier. The discrepancy in predictions between the preliminary classifiers fit on different subsets of the data was used to extract a small set of author pairs for subsequent manual classification. Iteration was repeated until the full set of 2K pairs was manually classified. The resulting labeled dataset was then used to train a random forest model to predict author IDs for the remaining developers (see [26] for full detail).

After applying the technique on 9,532 author IDs, 3,299 IDs were merged resulting in 6,233 distinct developers. We map multiple names and email addresses used by a single developer to a unique ID representing that developer. Fig. 1 shows an example of a developer's merged identities.[11] To check the accuracy of this identity merge we conducted a survey of developers as described in Step 4 below.

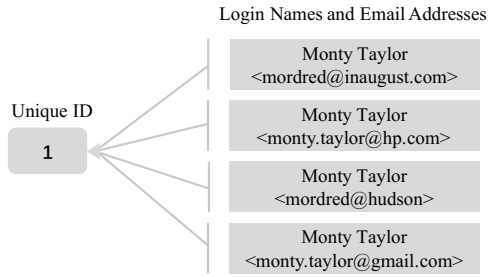

Fig. 1. Example of a merged identity

Step 2. Identifying developers' affiliations. It is important to note that each developer may have multiple affiliations during the time they were contributing to OpenStack. To determine these affiliations and time periods associated with each affiliation, we used the OpenStack community member list that can be found on its official website,[12] which provides the individual profiles of its community members. Each profile has an "Affiliations" field, containing all the companies that supported the developer to work on OpenStack and the corresponding time periods for those affiliations. It is entered and updated by the developers themselves and officially maintained by OpenStack. Fig. 2 shows an example profile. If the members are unaffiliated volunteers, their affiliation might be "volunteer," "unaffiliated," "individuals," etc. We obtained all the profiles via a crawler script, and conducted the following preprocessing:

- We replaced the end date "Current" (as shown in Fig. 2) of the affiliations with the date of data crawling, i.e., "2018-08-18".
- Some of the profiles contain "OpenStack" (e.g., "OpenStack Infrastructure", see Fig. 2) as one of a developer's affiliations which we determined to be provided by developers who wanted to demonstrate their expertise related to OpenStack. This peculiar affiliation also often overlapped with proper affiliations. We, therefore, ended up not considering "OpenStack" as an affiliation.

The 6,233 developers identified in Step 1 were matched with their corresponding affiliation history. We looked for each developer (who may have multiple names and emails in their commit author IDs) and considered it a match if at least one of her names matches the profile name and at least one of her email domains could be inferred from her profile. Note that the developers who exclusively use non-enterprise emails are considered as volunteers if there is no affiliation provided in their profile. We also identify as

11. We have obtained the permission to display this developer's personal information in this paper.

12. https://www.openstack.org/community/members/

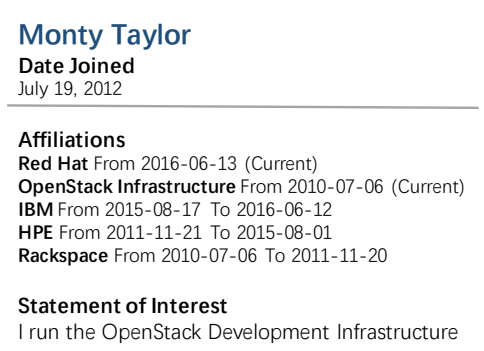

Fig. 2. A developer's profile listed in the OpenStack website

volunteer developers who have their affiliations identified as "volunteer,""unaffiliated," and so on.

The profile information covers 90% of developers. Table 4 shows an example of a developer and his affiliations that we determined. Note that each affiliation for a developer covers only the time period corresponding to the period she was supported by the specific company.

TABLE 4
Example of a Developer's Affiliations

Unique ID	Name	Affiliation	Start Date	End Date
1	Monty Taylor	Red Hat	2016-06-13	2018-08-18
		IBM	2015-08-17	2016-06-12
		HPE	2011-11-21	2015-08-01
		Rackspace	2010-07-06	2011-11-20

For the remaining 10% of developers whose affiliated companies had not been confirmed (i.e., they can not be found in the member profiles), we followed the following procedures. First, we considered their email domains. For example, if the email domain of developers was "redhat.com", they would be considered to be affiliated with Red Hat. Developers from consumer domains: "gmail.com", "outlook.com", "hotmail.com", etc., were classified as "Volunteer." In the next step handling this 10% of developers, we determined developers' tenure in each affiliation by considering the range of dates of the commits associated with each affiliation (email domain). Some developers were submitting code using their enterprise email and personal email over the same period. In such cases, we give priority to their enterprise email address.

Step 3. Identifying commits' affiliations.

After we identified which companies the developers work for and the exact time periods they work for each company, we determine affiliation for each commit made by the developer using this work history table. Specifically, if the author-submit time of a commit is within the interval representing author's tenure in a company, the commit would be assigned to that company. Thus, even developers who have worked for more than one company, have their commits properly attributed to each company they worked for.

Step 4. Manual verification. To validate the accuracy of the identity matching and affiliation assignment, we designed a survey. Instead of asking developers directly to

indicate whether we identified their multiple identities and historical affiliations correctly, we adopted a less intrusive approach. Specifically, for each unique pair of a developer's identity and affiliation, we randomly selected one commit and recorded the affiliation. We asked developers to confirm if these commits with pairs of identity and associated affiliations were done by them on behalf of the affiliated company. In the pilot stage, we randomly selected 50 developers to review the survey to ensure that the questions were clear and complete. We received 6 responses that suggested minor edits, including changing the term "volunteers" to "independent contributors" and clarifying the wording of some questions. We randomly selected 400 developers (with an error margin of 5% and confidence level of 95%) and sent them questions. Sixty-nine emails were returned due to the delivery problems. After 20 days, we obtained 45 responses, resulting in a response rate of 13% ($\frac{45}{400-69}$). Of the 45 answers, no respondent indicated that the commits were not submitted by them, and only three respondents corrected their affiliations. This suggests that the accuracy of developers' affiliations was approximately 93% and the accuracy of developers' identities has 95% confidence interval of $[0.99; 1]$, i.e., if the probability of that developer's identity being correct is 0.99, then 95% of the time we can observe zero mistakes in randomly chosen 45 matches.

2.4 Discovering and Characterizing Contribution Models

The existing literature provides a number of theories regarding why and how companies engage in OSS ecosystems [8, 36, 37, 38, 39]. As discovered repeatedly, commercial objectives always drive companies' strategic actions when they participate in OSS ecosystems [8, 37, 39]. Different companies tend to employ similar actions and policies if their goals are consistent. For example, both SUSE and Canonical maintain OpenStack to promote their Linux distributions [40]. Thus, we decided to use commercial objectives to classify the contribution models of companies' participation.

To characterize the participation performance of companies that belong to different models, we need to introduce suitable dimensions. Our preliminary investigation of OpenStack from online documents and commit data combined with the existing literature resulted in a proposal to use three dimensions: contribution intensity, contribution extent, and contribution focus. The contribution intensity and extent have been used by earlier studies to characterize commercial participation in OSS ecosystems [8]. We found that some companies focus on a particular type of projects. For example, more than half of the contributions of SUSE are devoted to Project-config, OpenStack-manuals, and OpenStack-doc-tools, all of which belong to *Documents* type.

2.4.1 Extracting contribution models

Commercial objectives refer to the motivation of a company to join an OSS ecosystem [37, 41]. To obtain the commercial objectives of the involved companies, we first sorted companies by their contributions (number of commits) to OpenStack. We investigated companies in that order. Specifically, for each company, we entered its name plus OpenStack in Google search engine, visited the top 20 links to explore the company's commercial objectives, and collected the related online records. We also collected documents from the marketplace page in the OpenStack official website[13] regarding the products, services, or solutions produced by the companies. We stopped at the top 124th company because the remaining companies had insufficient online data and we were unable to obtain their commercial objectives. Although the 124 companies only account for 22% of companies overall, they contributed approximately 90% of the commits to OpenStack.

We analyzed these records to obtain the categories of commercial objectives by using thematic analysis, a widely used technique for identifying and recording "themes" in textual documents [42, 43, 44]. The process mainly involves the following steps: (1) initial reading of the records, (2) generating initial codes for each record, (3) searching for themes among the proposed codes, (4) reviewing the themes to find opportunities for merging, and (5) defining and naming the final themes. We used MAXQDA[14] to support these steps. To reduce bias by individual researcher, steps (1) to (4) were performed independently by the first two authors [13]. After this, a sequence of meetings was held to resolve conflicts and to assign the final themes (step 5). When the first two authors fail to reach an agreement on a particular code or theme, we use a third author as an arbitrator. This process revealed eight themes of a wide variety of companies' commercial objectives. We considered companies with the same theme to be in the same contribution model, labeled by the theme.

2.4.2 Characterizing the performance of contribution models

Dimension 1: Contribution intensity (denoted as *CI*) is used to characterize the degree of a company's contributions to OpenStack compared to other companies. As mentioned earlier, some companies tend to show strong support for a specific project or a specific type of projects to make prominent contributions. Thus, we calculated contribution intensity at three levels: the overall OpenStack, the specific type of projects, and the specific project.

We define a company's *CI* as a ratio of the contributions contributed by the company to the total contributions at three levels, where the contributions are calculated both in developer (denoted as *dvpr*) and in commit (denoted as *cmt*) terms. Due to space constraints, we only show the *CI* formulas in developer terms:

$$CI^{O}(c;r) = \frac{\#dvpr_{c;r}}{\sum_{i} \#dvpr_{i;r}} \quad (1)$$

$$CI^{T}(c;r;t) = \frac{\#dvpr_{c;r;t}}{\sum_{i} \#dvpr_{i;r;t}} \quad (2)$$

$$CI^{P}(c;r;p) = \frac{\#dvpr_{c;r;p}}{\sum_{i} \#dvpr_{i;r;p}} \quad (3)$$

where the numerators $\#dvpr_{c;r}$, $\#dvpr_{c;r;t}$, and $\#dvpr_{c;r;p}$ represent the number of developers contributed by company c

13. https://www.openstack.org/marketplace/
14. https://www.maxqda.com

to OpenStack overall, to project type t and to project p in release r.

We denote $CI^{O}(c) = median\{CI^{O}(c;r); r \in Rs\}$ as the *CI* of company c at the level of the overall OpenStack, where Rs represents all the releases in which company c participated. We denote $CI^{T}(c) = median\{max\_CI^{T}(c;r); r \in Rs\}$ as the *CI* of company c at the level of project type, where $max\_CI^{T}(c;r) = max\{CI^{T}(c;r;t); t \in Ts\}$, Ts refers to the set of project types where company c has made contributions in release r. We denote the *CI* of company c at the project level as: $CI^{P}(c) = median\{max\_CI^{P}(c;r); r \in Rs\}$, where $max\_CI^{P}(c;r) = max\{CI^{P}(c;r;p); p \in Ps\}$, Ps refers to the set of projects where company c has made contributions in release r. Accordingly, we can obtain the *CI* in commit terms by simply replacing *#dvpr* with *#cmt*. The greater the value of *CI* is, the greater is the intensity of company c's contributions.

For each contribution model, we took the median[15] of its companies' *CIs* at three levels ($CI^{O}(c)$, $CI^{T}(c)$, and $CI^{P}(c)$) in developer and commit terms to represent the model's coordinates of contribution intensity.

Dimension 2: Contribution extent (denoted as *CE*) explains the scope of a company's contributions to OpenStack, focusing on the levels of project and project type. We define a company's *CE* as a ratio of the number of projects or project types contributed by the company to the total number of projects or project types in OpenStack:

$$CE^{P}(c;r) = \frac{\#prj_{c;r}}{\#prj_{r}} \quad (4)$$

$$CE^{T}(c;r) = \frac{\#prj\_tp_{c;r}}{\#prj\_tp_{r}} \quad (5)$$

where $\#prj_{c;r}$ and $\#prj\_tp_{c;r}$ represent the number of projects and project types contributed by company c in release r respectively. The denominators represent the total number of projects (in the fourth formula) and the total number of project types (in the fifth formula) in release r. Because OpenStack evolved over time, the number of projects varies across different releases, and this is true for different project types as well.

We define *CE* of company c at the project level as: $CE^{P}(c) = median\{CE^{P}(c;r); r \in Rs\}$ and define *CE* of company c at the project type level as: $CE^{T}(c) = median\{CE^{T}(c;r); r \in Rs\}$, where Rs represents all the releases in which company c participated. The greater the value of *CE* is, the greater is the proportion of projects or project types that company c has contributed to.

For each contribution model, we took the median of its companies' *CEs* at two levels ($CE^{P}(c)$ and $CE^{T}(c)$) to represent the model's coordinates of contribution extent.

Dimension 3: Contribution focus refers to the type of projects to which a company makes the most contributions. To obtain the contribution focus of a company, we sorted the project types by the number of commits made by the company, recorded the first type in each release, and counted the number of occurrences of each type over all releases. The most frequently occurring type was treated as the company's contribution focus.

15. We took the median value for a more reliable representation [45], while the mean value also presents the similar performance.

For each contribution model, we counted the number of occurrences of its companies' focus and considered the most frequent type as the model's contribution focus.

2.5 Diversity of different contribution models

"Shannon entropy" is proposed by Shannon to measure the unpredictability of the state, or equivalent of its average information content [46], and has been applied in a number of software engineering studies [47, 48, 49]. It is a well-established and frequently-used diversity measure for categorical variables [9, 50, 51]. To answer RQ3.2, we borrowed the classical idea of "Shannon entropy" to design a metric to describe the diversity of contribution models:

$$CMEntropy_{p;r} = \sum_{m} -P(cmt_{m;p;r}) \log_2 P(cmt_{m;p;r}) \quad (6)$$

where $P(cmt_{m;p;r}) = \frac{\#cmt_{m,p,r}}{\sum_{i} \#cmt_{i,p,r}}$ represents the ratio of the number of commits contributed by companies of model m to project p's commits in release r. The higher *CMEntropy* is, the more even the different models' contributions are, and the more diverse the commercial participation is.

3 Results

RQ1: To What Extent Do the Companies Contribute to the Development of OpenStack?

As shown in Fig. 3 and Fig. 4, the black bars represent the developers and commits from companies, and the gray bars represent volunteers', respectively. The horizontal axes represents the 14 releases. We can see that the number of developers from companies ranges from 71 to 2,332, while the number of volunteers changes from 22 to 390 among the 14 releases. At the same time, the number of commits made by companies ranges from 1,644 to 41,156, and volunteers' commits change from 250 to 3,544. On average, the number of developers assigned by the companies is approximately 4.3 times the number of volunteers, and the number of commits contributed by the companies is approximately 10.3 times that of volunteers.

Further, we calculated the proportions of the contributions made by companies and volunteers in every release. The results show that companies' proportions are greater than that of volunteers in all releases. On average, the proportion of developers invested by the companies is approximately 80%, and the proportion of commits made by these developers is 90%. The proportion of volunteers' contributions appear to decrease over releases, from 32% to 14% in terms of developer and from 18% to 7% in terms of commit.

In summary, companies made far more contributions as represented by the number of developers and commits than volunteers. Moreover, the percentage of contributions made by volunteers decreases over time.

RQ2: Are Contributions Distributed Unevenly among Companies?

For convenience of calculation, we consider volunteers as belonging to a single "unaffiliated" organization separate from other companies when investigating this research

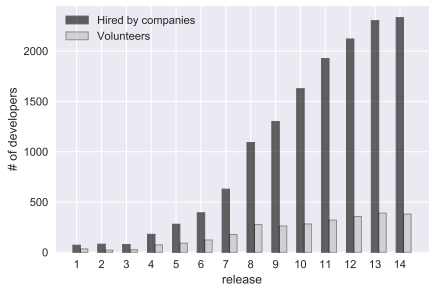

Fig. 3. The number of developers from companies and volunteers.

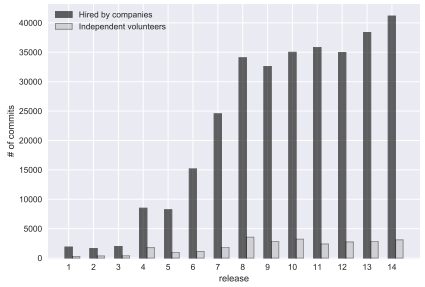

Fig. 4. The number of commits from companies and volunteers.

question. We investigated the proportion of contributions provided by various companies in the lifespan of OpenStack we studied. We found that approximately 20% (18) of the companies devoted 80% of the developers and 10% (9) of the companies contributed 80% of the commits, where the companies are ranked by the number of developers or commits they invested. In particular, the top 10 companies (approximately 1.8%) contributed approximately 69% of commits. Of the remaining companies (about 555), approximately 90% contributed less than 0.1% (362) of commits, and approximately 70% contributed less than 0.01% (36) of commits. Similar to the commit term, the top 10 companies contributed 63% of the developers. Out of the remaining companies, approximately 87% contributed less than 0.1% (10) of developers. It is important to note that the group accounting for the most developers is "volunteers", which means that volunteers are still the main source of developers in OpenStack compared to any single company.

We further calculated the fraction of companies that are together responsible for 80% of the developers and commits for every release, where companies are ranked by their developers and commits, as shown in Fig. 5. The fraction of companies that is responsible for 80% of the developers or commits appears to be decreasing over releases. This is largely due to the increasing number of companies; e.g., in the first release, there were 19 companies and 5 companies responsible for 80% of commits ($\frac{5}{19}$ = 26%), while in the 14th release, there were 269 companies and 16 companies responsible for 80% of commits ($\frac{16}{269}$ = 6%). In other words, the number of participating companies grows much faster than the size of the core team who is responsible for the majority of contributions to OpenStack. It may suggest an increase in the concentration of the commercial participation in OpenStack.

We also borrowed a well-known and frequently-used metric, Gini coefficient [52], to observe the uneven distribution of contributions among companies. This coefficient, introduced by Conrado Gini to measure income inequality in economics, shows how unequal something is distributed among a group. To calculate the Gini coefficient, we obtain the Lorenz curve firstly, a graphical representation of the cumulative distribution function of a probability distribution [53]. Perfect distribution of contributions among companies is given by a 45 degree line. The Gini coefficient is hence given by the area between the two curves, providing how far the actual distribution is from the perfect equality. Consequently, values of the Gini coefficient close to 0 correspond to equal or almost equal distributions, while values close to 1 are good indicators of high inequalities. Fig. 6 presents the Gini coefficient for the commits contributed by different companies of OpenStack in the time span we studied. As we can see, approximately 90% of the companies is responsible altogether for less than 10% of the total number of commits, with a Gini coefficient of 0.951. Approximately 20% of the companies devote 80% of the developers, with a Gini coefficient of 0.856.

Because the companies may change their contribution over time, we have also calculated the Gini coefficient on a release basis, as shown in Fig. 7. We can observe an upward trend in both commit and developer terms. The smallest Gini coefficient is 0.632 in commit term and 0.576 in developer term. This indicates that the contribution distribution among companies in OpenStack has always been uneven, and the degree of inequality is getting more unequal over time. It also suggests an increase in the concentration of the commercial participation.

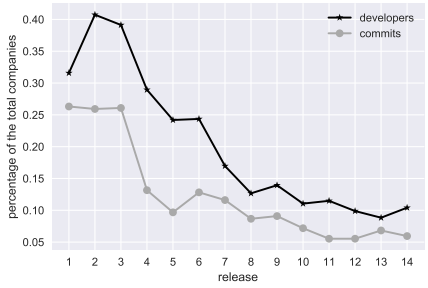

Fig. 5. Fraction of the companies who are responsible for 80% of the developers and commits.

A small core group does most of the work and coordinates a much larger group of peripheral participants in OSS ecosystems, meeting the Pareto distribution [1, 17, 54]. It appears that the Pareto-like phenomena, having been frequently encountered in software engineering, also applies to the companies that make contributions to OpenStack. This rather extreme distribution of (participating) companies may indicate that OpenStack is dominated by a few companies. This, to some extent, supports the same findings that have been found on four main projects of OpenStack [9]. OSS projects sometimes fail when their dominant companies withdraw [8]. So this unbalanced distribution may threaten the sustainable development of OpenStack.

Fig. 6. Gini coefficient and Lorenz curve for cumulative commit contributions from companies participated in OpenStack during the 14 releases.

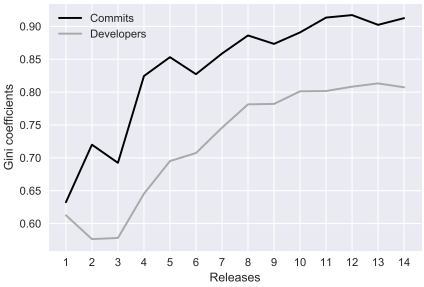

Fig. 7. Gini coefficient in both commit and developer terms per release.

In summary, the distribution of companies' contributions in OpenStack is highly uneven. The concentration of contributions increases over time.

RQ3.1: What Are the Contribution Models That Companies Employ and Do They Vary in Terms of Code and Developer Contributions?

Using the method described in Section 2.4.1, we obtained eight contribution models: **f**ull **s**olution **o**riented (denoted as FSO), **s**pecific **s**ub-**s**olution **o**riented (denoted as SsSO), **s**elf-**b**usiness **o**riented (denoted as SBO), **s**pecific **s**ervices **o**riented (denoted as SSO), **u**sage **o**riented (denoted as UO), **c**ommunity **o**riented (denoted as CO), **d**evelopment **i**nfrastructure **v**endor (denoted as DIV), and **r**esearch **o**riented (denoted as RO). Fig. 8 shows the occurrences of different models in the 124 studied companies. We can see that the models "Full solution oriented" (25%) and "Usage oriented" (23%) account for the majority of these companies, which may suggest OpenStack is a popular platform supported by sufficient vendors and users.

Companies in the FSO model tend to make profit directly by providing full cloud solutions to users, including private/ public/ hybrid cloud services, deployment, and maintenance services, etc. Most companies that have made significant contributions belong to this model, such as Mirantis, IBM, and Red Hat. The most common industry under FSO is cloud computing, accounting for 61% (19). The companies from other industries, often have set up cloud computing as a new business since they directly benefit from OpenStack.

Companies in SsSO make profits by providing solutions to users only on the basis of one or two project(s) in OpenStack. This contribution model directly relates to the original industry of the companies. For example, SwiftStack,[16] powering hybrid cloud storage for enterprises, mainly focuses on Swift (providing object storage service in OpenStack[17]), and its commits contributed to Swift reach 75% of its total commits to OpenStack.

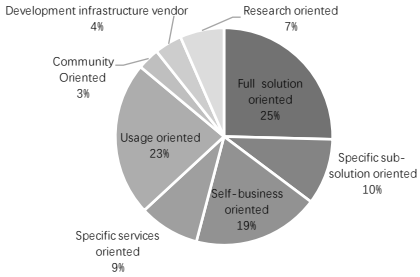

Fig. 8. Distribution of the 124 companies in the eight contribution models.

Companies in SBO integrate OpenStack with their original facilities and profit indirectly from OpenStack. For example, Citrix contributes most to "fuel-plugin-xenserver" in order to make its commercial product ("XenServer") compatible with OpenStack.[18] Other typical examples are Intel, Dell, and Fujitsu.

While OSS attracts more and more enterprises, some risks are perceived that often impede its adoption on a wider scale in marketplace. The biggest obstacles include the lack of centralized support due to "open" ownership, compatibility issues with skills and tasks, and poor documentation [55]. As OSS is not "owned" by a software provider who can provide support and training, users frequently have to seek support from other channels. Therefore, providing complementary services becomes a business opportunity [55, 56]. The same applies to OpenStack, and companies in the SSO model grasp this opportunity. For example, Objectif Libre[19] profits by providing consulting and training services around OpenStack. All the companies in the SsSO, SBO, and SSO models with a strategic dependence on a few projects closely related to their own business.

UO is a relatively simple model where companies join OpenStack because they use it in their production environment. Typical examples are AT&T, eBay, and Walmart. The most common industry under the UO model is telecommunication, accounting for 47%. This might be because OpenStack brings opportunities to some novel businesses without vendors in this industry, e.g., implementing network function virtualization on the basis of OpenStack.

Without any specific commercial objectives, companies in the CO model want to contribute to OpenStack because they are "living symbiotically off an open source ecosystems" [56]. Typical examples are the Linux Foundation, Debian community, and Cybera.[20]

16. https://www.swiftstack.com/
17. https://www.openstack.org/software/releases/stein/components/swift
18. https://xenserver.org/
19. https://www.objectif-libre.com/
20. https://www.cybera.ca/

As with all software projects, OpenStack requires an underlying infrastructure to support distributed-multiplayer-collaborative development, such as review management. Companies in DIV provide this type of infrastructure for OpenStack, e.g., Google's Gerrit and Puppet lab's Puppet. Similar to the CO model, this model's companies appear to share no commercial objectives toward OpenStack. Companies in RO are interested in the new technology brought up in OSS or the survival mechanism of OSS ecosystems [37]. Most universities and scientific research institutions take this model.

To conclude, companies are motivated by a variety of reasons to participate in OpenStack. By analyzing the themes of these reasons, we obtained eight contribution models. Companies in different models may profit directly by providing full/partial solutions or services on the basis of OpenStack, may indirectly profit by combining their products with OpenStack, or may not profit at all, for example, companies motivated by usage needs and research interests.

We calculated the performance of the eight contribution models, based on the metrics defined in Section 2.4.2. Table 5 shows the three levels contribution intensity (i.e., CI^O, CI^T, and CI^P in commit and developer terms) and the two levels contribution extent (i.e., CE^P and CE^T) of each contribution model. Based on the definition of contribution focus, we obtained the type of projects that each model's companies most prefer to participate in, as shown in Table 6. Next, we introduce the performance of each contribution model along the three dimensions.

Dimension 1: Contribution intensity. As shown in Table 5, the CI^O of FSO is the strongest among all models regarding both commits and developers, the same as the CI^T. The CI^P of FSO is only second to that of DIV while much smaller, which may be explained that, as discussed later – the companies in DIV have a much narrower contributing scope than the companies in FSO and much more intense contributions on the particular projects. In general, the contribution intensity of the companies in FSO is relatively large, possibly because they are benefiting from OpenStack directly.

The CI^T in SsSO is strong (only second to that of FSO), since the companies in this model tend to focus on a type of projects that is directly related to the specific sub-solution they provide. Companies in DIV tend to centralize their contributions on the same projects (suggested by the strongest CI^P in Table 5). As discussed earlier, these companies host the projects that are used to develop OpenStack. For example, approximately 95% of Google's commits are focused on Gerrit.[21]

The three levels of contribution intensity in the other models are relatively weak. As discussed earlier, the companies in the UO model are users of OpenStack and therefore make a few contributions. Similarly, it is easy to understand the weak contribution intensity of the CO model because the non-profit organizations do not have many developers. It is not surprising to observe that the research organizations in RO have the weakest CI^O and CI^T (and weak CI^P).

21. http://stackalytics.com/?release=all&metric=commits&project_type=all&company=google

Dimension 2: Contribution extent. From Table 5, we can see that the companies in FSO tend to have the most extensive contributing scope. These companies provide their customers with commercial cloud solutions toward OpenStack, so they are more willing to devote resources on a large scale. The UO model also makes extensive contributions (ranking #2). Companies in this model are motivated by requirements derived from their production environment, which seems to cover a range of projects owing to the diversity of usage scenarios.

As for the companies in SsSo, SBO, SSO, selecting which projects to contribute to is motivated by their specific business goals, which seem to cover a relatively small range of projects.

DIV has the weakest CE^T (and weak CE^P). Because companies in this model such as Google may be only interested in the projects that they dominate and are used by OpenStack. Similarly, the weak performance of RO could be explained by the nature of the research organizations – the projects that are interesting to researchers are not numerous.

Dimension 3: Contribution focus. As shown in Table 6, the results of each contribution model' focus well match its companies' commercial objectives. Despite intensive and extensive contributions, companies in the FSO model have a preference for *Computing* projects, which provide one of the main services of OpenStack together [9]. Most of the companies in the SsSO model provide storage services to customers, so their contribution focus is the *Storage* type, such as "Cinder" and "Swift". Consultancy companies, aiming to make a profit by providing complementary services based on OpenStack, contribute more to the *Documents* projects. A primary problem faced by the users of OpenStack, a large cloud computing operation system, is how to deploy various cloud services in their own production environment [57]. This may indicate that the deployment tools is the UO model's contribution focus. It is interesting to observe that the organizations in CO prefer to support development infrastructure.

To validate whether we characterized the eight contribution models appropriately, we clustered the 124 companies using KMeans (k is 8 in our case), a simple and widely used clustering algorithm [58]. Each company is an observation in the form of a 22-dimensional vector, containing contribution intensity (CI^O, CI^T, and CI^P in both developer and commit terms), contribution extent (CE^P and CE^T), and contribution focus (represented as a 14 dimensional binary vector). The clustering results had a BSS/TSS ratio of 93.5% indicating a good fit. Almost all companies from the five models (SsSO, SBO, SSO, UO, and RO) ended up clustered according to their categories. Only a few companies in the FSO model got mixed into the CO and DIV models; DIV also mixed with CO. At least five contribution models are, therefore, quantitatively detectable through measures of contribution intensity, extent, and focus.

We also conducted a survey to validate our models. We selected the 124 companies and, in order to find knowledgeable developers to represent their company, we asked the top five developers (ranked by their commits) from each company to nominate colleagues whom they deemed to have deep insights regarding their company's participation in OpenStack. We contacted the candidates from each com-

TABLE 5
Statistic Results of Contribution Performance of the Eight Contribution Models

Metrics / Contribution Models	Contribution Intensity			Contribution Extent	
	CI^O	CI^T	CI^P	CE^P	CE^T
Full solution oriented (FSO)	0.0035, 0.0058	0.019, 0.012	0.069, 0.038	0.073	0.46
Specific sub-solution oriented (SsSO)	0.0017, 0.0019	0.014, 0.011	0.042, 0.036	0.022	0.27
Self-business oriented (SBO)	0.00090, 0.0016	0.0057, 0.0080	0.015, 0.023	0.021	0.22
Specific services oriented (SSO)	0.0014, 0.0015	0.0065, 0.0052	0.033, 0.038	0.021	0.21
Usage oriented (UO)	0.0011, 0.0021	0.0066, 0.0064	0.037, 0.033	0.030	0.31
Community oriented (CO)	0.00065, 0.0011	0.0059, 0.0051	0.032, 0.051	0.017	0.16
Development infrastructure vendors (DIV)	0.00081, 0.0014	0.0025, 0.0048	0.24, 0.14	0.022	0.084
Research oriented (RO)	0.00017, 0.00091	0.0019, 0.0032	0.029, 0.032	0.0078	0.10

In each cell that has two numbers, the fist number is the metric in commit term, and the second is in developer term.

TABLE 6
Contribution Focus of the Eight Models

Contribution Model	Contribution Focus
Full solution oriented	Computing
Specific sub-solution oriented	Storage
Self-business oriented	Networking
Specific services oriented	Documents
Usage oriented	Deployment
Community oriented	Infrastructure of development
Development infrastructure vendors	Infrastructure of development
Research oriented	Orchestration

pany and asked if we have categorized their companies accurately. We received 16 responses from IBM, Huawei, Nokia, Time Warner Cable, etc., with at least one response from each of the eight models. None of the respondents disagreed with our categorization. For example, the developer representing Nokia (SBO) said "*I somewhat agree on this conclusion. We want to integrate our business with Networking therefore make contributions related to that*". The survey supports our eight models of company contributions.

In summary, there are eight contribution models derived from commercial objectives in OpenStack. Companies that make substantial and extensive contributions tend to be the providers of a full cloud solution. Usage–oriented companies tend to contribute to a large scope of projects and have a preference on deployment tools. Companies that select specific projects to contribute to either are driven by their particular business or are infrastructure vendors who develop infrastructure that supports the development of OpenStack. Minor contributors include organizations who are community players such as the Linux Foundation, and research groups driven by interest.

RQ3.2: Is the Diversity of Contribution Models Associated with Volunteer Participation?

As the answers for RQ2 suggest, the joint contribution of volunteers is relevant, ranking first in terms of overall developers and fifth in terms of overall commits. However, the growth of volunteers' contributions is slow, and their contribution percentage decreases over time when compared with all the companies. Meanwhile, the concentration of companies' contributions increases over time, which may suggest a decrease of diversity of companies. It has been found that a company's full control and intense involvement are associated with a decrease in volunteer inflow [8]. Therefore, it is of interest to investigate the relationship between the diversity of companies (represented by contribution models) and volunteer participation.

We measure the contribution diversity with *CMEntropy* defined in Section 2.5. We measure volunteer participation by the number of volunteers (*nVltr*) in each release of every project (1,553 observations). The smallest number was one for 591 of the project/release combinations, while the largest was 46 for Release 13 of *Openstack-manuals* project. The median number was two and a total of 291 project/release combinations had two volunteers.

Based on previous studies, e.g., [8, 28], we assume that the number of volunteers is likely to be affected by the project context, such as how large and active the project is, what *Type* of project it is (see Section 2.1), and the specifics of a *Release* (mentioned in 2.2). We therefore include these predictors in a regression model with the response being the number of volunteers (*nVltr*). To measure project size, we use the total number of developers who participated in the project (*nTotal_dvpr*), and to measure release activity, we use the number of commits (*nTotal_cmt*). We log-transform skewed variables to satisfy the modeling assumptions. The Variance Inflation Factor (VIF) was moderately high at 5.6 (due to moderately high collinearity between total developers and total commits). Because both variables positively affect the number of volunteers, it was not necessary to remove one of these predictors from the model for interpretability (if either predictor is removed the remaining predictor still has a positive effect on the response). Additional considerations regarding conditional independence and suitability of linear models are discussed in Section 5. The final regression equation is:

$$\log(nVltr) \sim CMEntropy + \log(nTotal\_dvpr) + \log(nTotal\_cmt) + Release + Type$$

The results of the fitted model are shown in Table 7. We also include two types of projects (*Documents* and *Community Build*) that had statistically significant coefficients.

Both are associated with increased volunteer participation. The adjusted R^2 of the model is 0:63, indicating that the model explains the observed data well. All continuous predictors are statistically significant (at < 0:005 level). We follow Johnson's [59] recommendation to use a p-value of 0:005 for statistical evidence instead of the commonly used value of 0.05 because using the latter value often leads to unreproducible results.

TABLE 7
Coefficients of the Model (1,553 Observations).
Adjusted R^2 = 63%

	Estimate	Std.Err	Pr(>|t|)
(Intercept)	-0.84	0.19	0
CMEntropy	0.24	0.040	0
log(*nTotal_dvpr*)	0.51	0.032	0
log(*nTotal_cmt*)	0.12	0.022	0
Documents	0.26	0.094	0.0061
Community Build	0.56	0.14	0.00012

The results show that, as expected, the number of volunteers increases in more populous and more active project/release combinations, and higher *CMEntropy* (more diverse contributions) has a strong positive effect on volunteer participation. Finally, the two types of projects with a significantly larger proportion of volunteers may be the result of the phenomena that volunteers tend to do simple tasks [17] thus providing some support for our proposed classification of projects in Section 2.1. The answer to RQ3.1 has conveyed that companies from different contribution models prefer to participate in some specific types of projects. In this regression model, some specific types also have an effect on the participation of volunteers. For example, the coefficient for the project type *Community Build* (0.56) means that the projects of this type tend to have higher levels of volunteer participants.

We have included *Release* as a nuisance parameter since each release may affect volunteer participation differently. While the obtained coefficients are not of primary concern, it is still worth noting that all releases beyond the sixth have statistically significant (at < :005 level) negative coefficients. This indicates that the volunteer participation in OpenStack has dropped significantly over time at the project level.

More importantly, the *CMEntropy* predictor has a positive coefficient of 0.24. This means that a project with a higher diversity of companies would increase the number of volunteers who participated in it. A possible explanation for the observed effect is that the projects with more equal participation by different companies can ensure a good self-governance of the ecosystem and therefore provides a favorable environment for volunteers [60]. Another possible interpretation would be that the existence of multiple companies motivated by different objectives increases the degree to which participants must be treated equally regardless of company background. A volunteer can expect to gain equal standing with developers from well-known companies, so they can be assured that their effort will not be disregarded because they are unaffiliated.

To validate our findings, we conducted a survey to elicit developers' perspectives on the importance of volunteers to OpenStack and the impact of the diversity of contribution models on volunteer participation. We selected 30 senior developers who have more than five years of contribution experience and have taken roles in the OpenStack Foundation. We deem this type of developers to have a deep understanding of the OpenStack ecosystem. We sent emails in July, 2018 and obtained four responses. All four respondents hold a positive view of volunteers' importance to OpenStack, providing three reasons: 1. *bringing valuable feedback*; 2. *caring about the software for reasons other than a paycheck* (two respondents mentioned this); 3. *representing a mitigation against companies following tech fashion or hype curves*. All the respondents agreed that the diversity of companies has a positive impact on volunteer participation, and various explanations are provided, such as *a balanced and diverse commercial participation can ensure the effective performance and sustainability of OpenStack and encourage volunteers to participate*. The responses help understand the value of volunteers for the OSS community with extensive commercial participation, and interpret the association between the diversity of companies and volunteer participation.

In summary, the diversity of companies (in terms of contribution models) is positively associated with the number of volunteers in OpenStack.

4 Discussion

This section discusses implications of our findings, including the extensive commercial participation, the eight contribution models and the positive impact of company diversity on volunteers.

4.1 Extensive Commercial Participation

The early contributors to OSS were mostly volunteers [21], who reported bugs, proposed feature requests and made contributions to source code. Some companies, such as Netscape, saw advantages in making their products open source and over time the number of companies contributing to OSS increased. Examples of projects with massive commercial participation are the Linux kernel, the Android OS, and OpenStack. Despite a large body of research on commercial involvement in open source, the extent of developer and code contributions by different companies has not been extensively investigated. In our study of OpenStack we found that approximately 80% of developers and 90% of commits originated from companies. It is not entirely clear how commonly the open source development might be dominated by companies instead of volunteers and in what types of projects that domination may occur. Volunteer driven projects appear to be in a majority on GitHub [61], but the commercial participation in many OSS projects has been growing. The Linux kernel represents another case with heavy commercial participation. It has gone all the way from a volunteer-driven project to a consortium of companies [62]. Such high level of commercial participation might affect how open source development is conducted and enrich our understanding of open source research and practice. Some findings, drawn from OSS projects having primarily volunteer contributors, may need to be reconsidered in the context of OSS projects that have broad

commercial participation. For instance, the factors that affect the odds of newcomers becoming long-term contributors in OSS projects [28] may not apply to commercial participants. Furthermore, it is necessary to investigate other projects to understand whether such wide commercial participation is more common and what kind of OSS projects/ecosystems tend to attract primarily commercial participation.

4.2 Contribution Models

We extracted eight contribution models of companies from OpenStack that differ in their objectives, and contribution performance (i.e., intensity, extent, and focus). Some of the models are in line with the findings of earlier studies. For example, the product provider, infrastructure provider and service provider categorized by Linaker et al. [63] in the Apache Hadoop platform are similar to the cloud business types (e.g., FSO, SBO, and SSO) in OpenStack, and their platform user is similar to our usage oriented model. Our community oriented model is similar to the community-focused model in [14] and collaborating model in [8]. This suggests that commercial participation presents similarities across OSS ecosystems. However, the models of development infrastructure vendor and research oriented that were discovered in this study do not seem to have been discussed before, that expands our understanding towards commercial participation in open source. On the other hand, these eight models draw a picture of how hundreds of companies participating in hundreds of projects and forming an ecosystem that delivers a product (i.e., OpenStack release). This kind of picture helps to understand how a large-scale complex system is developed by various organizations driven by different commercial objectives, that has never been quantified before. However, the fine-relationships the companies may have such as competition, conflict and dependency that affects how an OSS ecosystem sustains requires further analysis.

From the practice perspective, these models can be used as a guidance to help companies intending to join open source develop participation strategies. For instance, if a company attempts to benefit directly from the software, i.e., selling packaged software-based solutions, it better makes intensive and extensive contributions, and pay close attention to the main functions around this software (such as computing projects in OpenStack). For those companies that have already involved in an OSS community, the dimensions to quantify the contribution performance can be used to characterize their effort and an ecosystem's status (in terms of companies' impact). Thus a company can be aware of the quantified contributions it devotes, where it puts its effort, and what impact it brings to the community. Companies can adjust their actions based on the understanding of the advantages and disadvantages (the same for a community). A company can also learn the performance of other companies, especially their competitors, then adjust their strategies to maximize resource utilization.

4.3 Volunteers and Diversity of Participating Companies

There is little doubt regarding the importance of volunteers in OSS projects [4, 28, 64] even if they are peripheral participants [65], because volunteers bring a number of significant benefits, for example, a high level of innovation potential, improved software quality, and new features [66, 67, 68]. The respondents of OpenStack in our study also pointed out that volunteers *represent a mitigation against companies following technology fashion* along with their contributions. However, as found in this study, the percentage of volunteers' contributions decreases over time. It is important to understand better why that is the case, to what extent volunteers add unique value to open source development in company-dominated ecosystems and what can be done to mitigate the lack of volunteers.

An OSS project often relies on contributions from many, not a single company, and that is likely to increase the diversity of contributions. Diversity may directly enhance the stability of an ecosystem by ensuring that the ecosystem has the capacity (in terms of the variation of capabilities and business objectives) to respond to environmental changes [12]. Diversity may also preserve the overall structure and productivity of a business ecosystem, therefore the individual members of the ecosystem may change, but the ecosystem as a whole persists [12]. This appears to apply to OSS ecosystems as well. For example, increased diversity among the contributing developers, such as fluency in different programming languages, improves OSS ecosystems' resilience [19]. Increased gender and tenure diversity were found to be associated with greater productivity of OSS teams [15, 20]. We found that the diversity of companies has a positive association with the number of volunteers. That may provide an avenue for attracting volunteers. But it is not clear what is the nature of this positive correlation and whether or not it is possible to increase the diversity of commercial participation in any particular software project. Both of these questions require additional investigations.

This study looked at the impact of commercial participation on volunteers, but its impact on many other relevant elements is unclear. For instance, does development efficiency and code quality improve after the involvement of companies? This question deserves attention because OSS is getting more important to our society and also because the commercial participation appears to be an irreversible trend.

5 Limitations

We now discuss the threats to the validity of our study, following common guidelines for case studies [13, 69].

5.1 Data

The accuracy of identifying the affiliations of developers (and corresponding commits) directly influences the validity of our results. To make developers' affiliations as accurate as possible, we first adopted a novel supervised learning based approach to merge duplicate developer identities, then explored both member profiles provided by the OpenStack website and developers' email domains recorded in the Git commits to identify developer affiliations, and, at last, verified the results of the identity merge and affiliation with a developer survey. The survey results indicate that our approach had an accuracy of 93%. Given the complexity of identity matching and the determination of affiliation, we believe that such results are sufficiently accurate for the purpose of our analysis.

Several reasons may lead to mis-affiliation of developers. On the one hand, although contributions from companies are welcomed and encouraged, it is possible that companies have less interest in asking their developers to be recognized as being a part of the corporation. Thus, these developers' affiliations may be mistakenly identified as volunteers if in cases when they use non-enterprise email accounts to submit commits. On the other hand, there might be a few developers who have contribute on their free time, although they are may be hired or supported by companies. This imperfect data may impact some of the results. For example, we have identified 564 companies in our dataset, while the actual number of companies may be larger or smaller. Correspondingly, it also means that the number of volunteers we have identified is only an approximation to the true number of volunteers. Considering the difficulties (such as the lack of valid information and limited time) of getting a 100% accuracy and the relatively high accuracy (93%) we have achieved, we leave the new approaches needed for the further increases in accuracy for this type of data for future research.

5.2 Internal Validity

The definition of volunteers we have used and the operationalization of it may be both debatable. We sought a practical definition that had a pototential to be easily measured. It is also similar to the often implicit definition of volunteers in the extant literature on open source.

In this work, we regarded each repository in OpenStack managed by Git as a project. Nevertheless, there may be projects split into multiple repositories for different purposes. For example, Fuel, a deployment tool, has 17 repositories, and every repository performs a part of functions of Fuel, e.g., the repository "fuel-agent" is for building operating system images.[22] We found that this phenomenon is not common, and approximately 3% of projects in our dataset have multiple repositories. We classify this kind of repositories into the same type but treat each as a separate project, because the participation of companies and volunteers varies among them.

It is not easy to agree on measures that reflect a company's contributions. We need to find a balance between what is desirable, e.g., implementing new features or fixing bugs, and what is feasible to calculate. To address this problem, we studied the related literature and the development process of OpenStack. Ultimately, we decided to use the number of commits and developers as estimates because they are useful for characterizing company participation, simple to calculate, and are widely used [8, 11]. More importantly, six experts from four companies, deeply involved in OpenStack, agreed with the two estimates in [11]. Future work may be needed to include other activities, e.g., bug fixes, email discussions, and code review changes, to investigate commercial participation in more detail.

In addition to results shown in Table 5, we can run statistical tests to indicate which objectives produce better contribution performance. Using the method described by Konietschke et al. [70] and applied by Vasilescu et al. [71], we can illustrate in greater detail which of the differences in contribution performance are statistically significant (the results are presented in the online appendix[23]). Specifically, the FSO model had the best performance, while CO and RO had the worst, with the remaining models being in the middle.

We have investigated the distribution of the variables to fit the regression models for RQ3.2 to detect outliers and multicollinearity. We also tried to adjust for a number of factors that may bias our results. Two serious concerns remain. The observations of volunteers may not be independent as each project may have the same or a similar set of developers (including volunteers) from release to release. To address this concern, we first checked for autocorrelations over releases within each project. We found that only seven out of 86 projects with more than five releases (a bare minimum needed to estimate autocorrelations) had partial autocorrelation significant at :05 and none at :01. Such partial autocorrelations suggest Auto Regressive (AR) time series, which exhibits errors that are conditionally not independent of predictors. Given that only a few projects had this issue and it was not severe, we feel that the presented analysis is appropriate. We further fit a random effects model that includes project id (to ensure conditional independence), and the described effects of entropy, participation, and activity still point in the same direction and are as significant (see the online appendix[23] for more detail). The second concern is related to the use of linear models when the response variable has a low number of counts (a median of two). To address this concern we also fit a negative binomial generalized linear model (the most appropriate for such low count data with variance that is much higher than, e.g., a Poisson distribution). The effects of entropy, participation, and activity are still pointing in the same direction and are as significant (see the online appendix[23] for more detail).

5.3 External Validity

Threats to external validity correspond to the generalizability of our work. OpenStack is an ecosystem for developing a cloud operating system, which may limit our findings to a single domain or even a specific OSS ecosystem. Yin [69] emphasized that case studies are generalizable to theoretical propositions and not to populations or universes. This work revealed companies having different goals with respect to their contributions in OpenStack through eight derived contribution models characterized by three dimensions. The dimensions and coordinates of different models in the hybrid space presented in this study appear to reveal the alignment between commercial objectives and contribution performance of companies. Finding the position of additional projects and companies in the hybrid space may reveal new flavors of OSS development within intense commercial participation and would likely highlight ways to both extend our findings and make them more precise and repeatable.

The findings drawn from OpenStack, a pioneer OSS ecosystem with intense commercial involvement, may well represent some ecosystems with companies engaged and can be used for reference. For instance, the dimensions to

22. https://github.com/openstack/fuel-agent

23. https://github.com/noname2018/Commercial-Participation-in-OSS/tree/master/Appendix.

quantify the contribution models can be used to characterize a company's effort and an ecosystem's status. So, a company can adjust its actions based on the understanding of where it places its efforts and how it impacts the community. A community can understand the evolution and the status of their ecosystem and take appropriate actions if problems emerge. The important role of volunteers and the positive association between their participation and company diversity conveyed in this work may be a wake-up signal for other OSS ecosystems with decreasing volunteer participation. To addressing this problem a potential solution is suggested this work: improving the diversity of participating companies and reducing the contributions of the dominating companies at the same time. However, whether or not this solution can be implemented in practice is not clear. As far as we know, the volume of contributions from volunteers has been in slow decline for many years in the Linux kernel [62], and it might be a good candidate for the proposed approach.

6 Related work

The first OSS communities were made up almost of volunteers [21] who, despite their affiliations, contributed to the projects on their own. The existing literature provides a number of theories and measures around volunteers in OSS projects, e.g., why they participate [28, 56, 64, 72], how they make contributions [4, 67, 68], how they coordinate their work in globally distributed environment [73, 74], and what influences their growth [75, 76]. Because of the advantages of OSS compared to traditional software, e.g., user innovation and low cost, more and more companies are attracted to join. Companies hired contributors from OSS communities or tasked their employees to contribute to, with the idea of gaining influence. Thus, great effort has been spent on investigating commercial participation in OSS [8, 21, 37, 77]. The investigation starts from exploring why companies adopt open source [41, 73, 78]. In comparison with individuals, companies are found to focus less on social motivations such as reputation and learning benefits, but emphasize much more economic and technological reasons [37, 39]. A study on the firm-developed innovations within the Linux kernel for embedded devices elaborates the importance of receiving outside technical support as a motivator for revealing code [41]. Some studies extract business strategies from the cases of commercial participation in OSS. For example, Daffara [79] analyzed 120 firms which derive their main revenue stream from OSS, and clustered these firms into six business strategies, such as twin licensing. Dahlander and Gallagher studied four firms involved in OSS and discovered three ways that firms use to connect with OSS communities [5].

As more and more companies get involved in open source, studies have focused on how and where companies devote their resources. Dahlander and Wallin found that the most common form of commercial participation is to deploy paid developers to OSS projects [80]. Other ways of investment include, employing core project developers, making donations, and joining project steering committees in order to advance strategic interests, as discovered by Butler et al. [81]. As the understanding of commercial objectives, strategies, and actions in OSS context improves, the research questions involving these three and their impact start being investigated. Wagstrom et al. [6] conducted an analysis of commercial involvement in GNOME and Eclipse and identified two types of commercial involvement: community-focused company building a vibrant GNOME community and monetizing services; product-focused company relying on product revenues. Zhou et al. [8] identified three commercial involvement models through analyzing three JavaEE application server projects, and found that full control mechanisms and high intensity of commercial involvement came with direct profit from OSS product, and were associated with a decrease of external inflow and with improved retention. Lee [82] investigated how the developers employed by companies influence OSS communities through conducting social network analysis on two OSS projects (SilverStripe and eZ), and found that the developers having central (degree) positions can influence the integrity and cohesion of the community network in various channels of communication. Ho and Rai [83] found that companies' quality control in OSS projects can influence volunteers' intentions of participation. One of our previous work [9] studied four main projects of OpenStack and found that the commercial domination is negatively associated with the participation of companies and contributors, while it is positively associated with the productivity of contributors and the quality of issue reports.

As the OSS projects no longer rely on a single company and that many companies are now investing significant efforts [21], the relationship and interaction among companies start to attract attention. Teixeira et al. investigated the cooperation among companies in OpenStack and found that companies tend to form alliances and that development transparency and weak intellectual property rights allow a focal company to transfer information and resources more easily between its multiple alliances [14].

Despite substantial studies on commercial participation, the degree and types of contributions made by various companies, and the impact of the diversity of companies on volunteers remain unclear. This paper bridges this gap by conducting an empirical study on OpenStack. More specifically, we followed Teixeira et al. [14] and Zhou et al. [8] to use commercial objectives to identify commercial participation models. While some of the models we discovered are similar to the models found in the earlier studies, e.g., the "community oriented" is similar to the "community-focused" [14] and "collaborating model" [8], we also extend the understanding of commercial participation to a wider scope. In contrast to prior studies that looked only at companies with the highest sponsorship, we considered all types of companies that participate in OpenStack. Moreover, we quantified companies' participation performance through multiple dimensions (i.e., contribution intensity, extent and focus) and fitted regression models to investigate the impact of the entropy (representing diversity) of commercial participation on volunteers, which have never been done before.

7 Conclusions

The spectrum and scale of commercial participation in OSS ecosystems have substantially expanded in recent years. Companies have started to take the lead, which may change

the nature of OSS development and affect the OSS community. It is necessary to have a clear understanding of the development of OSS ecosystems with intensive commercial participation.

The goal of this exploratory research was to investigate how companies contributed developers and commits to OpenStack. We found that companies made far more contributions than volunteers, playing a critical role in the development of OpenStack. The contributions made by different companies are highly uneven, with approximately 10% of the companies contributing 80% of commits and 20% providing 80% of the developers. We extracted eight contribution models from the companies involved in OpenStack by their commercial objectives and characterized the models by contribution intensity, extent and focus. We found that different companies contributed to different projects motivated by either similar or different objectives, but the diversity of these companies has a positive association with the numbers of volunteers. The framework we proposed to quantify the companies' contributions to OSS may help provide a multifaceted view and clarify the factors that allow rapidly growing ecosystems to be sustainable and the practices that reduce the risk of failure. Our study contributes to the understanding of increasing commercial participation in OSS development. To the best of our knowledge this is the first work that provides empirical evidence that volunteer participation is affected by the diversity of companies.

To facilitate replications of our work or other types of future work, we provide the data, scripts and retrieved materials used in this study online[10].

ACKNOWLEDGMENTS

We would like to thank the contributors in OpenStack for their valuable feedback on this study. We also thank the reviewers for their great suggestions. This work is supported by the National Basic Research Program of China Grant 2015CB352200, and the National Natural Science Foundation of China Grants 61432001, 61690200, and 61825201.

REFERENCES

[1] M. Zhou, Q. Chen, A. Mockus, and F. Wu, "On the scalability of linux kernel maintainers' work," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ACM, 2017, pp. 27–37.

[2] D. Harhoff, J. Henkel, and E. Von Hippel, "Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations," *Research policy*, vol. 32, no. 10, pp. 1753–1769, 2003.

[3] J. West and S. Gallagher, "Challenges of open innovation: the paradox of firm investment in open-source software," *R&d Management*, vol. 36, no. 3, pp. 319–331, 2006.

[4] J. Feller, B. Fitzgerald *et al.*, *Understanding open source software development*. Addison-Wesley London, 2002.

[5] L. Dahlander and M. Magnusson, "How do firms make use of open source communities?" *Long range planning*, vol. 41, no. 6, pp. 629–649, 2008.

[6] P. Wagstrom, J. Herbsleb, R. Kraut, and A. Mockus, "The impact of commercial organizations on volunteer participation in an online community," in *Academy of Management Annual Meeting*, 2010.

[7] X. Ma, M. Zhou, and D. Riehle, "How commercial involvement affects open source projects: three case studies on issue reporting," *Science China Information Sciences*, vol. 56, no. 8, pp. 1–13, 2013.

[8] M. Zhou, A. Mockus, X. Ma, L. Zhang, and H. Mei, "Inflow and retention in oss communities with commercial involvement: A case study of three hybrid projects," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 25, no. 2, p. 13, 2016.

[9] Y. Zhang, X. Tan, M. Zhou, and Z. Jin, "Companies' domination in floss development: an empirical study of openstack," in *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*. ACM, 2018, pp. 440–441.

[10] S. K. Shah, "Motivation, governance, and the viability of hybrid forms in open source software development," *Management science*, vol. 52, no. 7, pp. 1000–1014, 2006.

[11] J. M. Gonzalez-Barahona, D. Izquierdo-Cortazar, S. Maffulli, and G. Robles, "Understanding how companies interact with free software communities," *IEEE software*, vol. 30, no. 5, pp. 38–45, 2013.

[12] M. Iansiti and R. Levien, *The keystone advantage: what the new dynamics of business ecosystems mean for strategy, innovation, and sustainability*. Harvard Business Press, 2004.

[13] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, p. 131, 2009.

[14] J. Teixeira, S. Mian, and U. Hytti, "Cooperation among competitors in the open-source arena: The case of openstack," *Social Science Electronic Publishing*, 2016.

[15] D. Izquierdo, N. Huesman, A. Serebrenik, and G. Robles, "Openstack gender diversity report," *IEEE Software*, vol. 36, no. 1, pp. 28–33, 2018.

[16] J. A. Teixeira and H. Karsten, "Managing to release early, often and on time in the openstack software ecosystem," *Journal of Internet Services and Applications*, vol. 10, no. 1, p. 7, 2019.

[17] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and mozilla," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, no. 3, pp. 309–346, 2002.

[18] T. Elmqvist, C. Folke, M. Nyström, G. Peterson, J. Bengtsson, B. Walker, and J. Norberg, "Response diversity, ecosystem change, and resilience," *Frontiers in Ecology and the Environment*, vol. 1, no. 9, pp. 488–494, 2003.

[19] T. Mens and P. Grosjean, "The ecology of software ecosystems," *Computer*, vol. 48, no. 10, pp. 85–87, 2015.

[20] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov, "Gender and tenure diversity in github teams," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 3789–3798.

[21] J. M. Gonzalez-Barahona and G. Robles, "Trends in free, libre, open source software communities: From volunteers to companies," *it–Information Technology it–*

Information Technology, vol. 55, no. 5, pp. 173–180, 2013.

[22] D. E. Perry, S. E. Sim, and S. M. Easterbrook, "Case studies for software engineers," in *Proceedings. 26th International Conference on Software Engineering*. IEEE, 2004, pp. 736–738.

[23] R. K. Yin, *Case study research and applications: Design and methods*. Sage publications, 2017.

[24] J. A. Teixeira, H. Karsten, and G. Widén, "Investigating knowledge management practices at openstack," in *European Conference on Information Literacy*. Springer, 2018, pp. 201–210.

[25] G. Robles, J. M. González-Barahona, C. Cervigón, A. Capiluppi, and D. Izquierdo-Cortázar, "Estimating development effort in free/open source software projects by mining software repositories: a case study of openstack," in *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014, pp. 222–231.

[26] S. Amreen, A. Mockus, R. Zaretzki, C. Bogart, and Y. Zhang, "Alfaa: Active learning fingerprint based anti-aliasing for correcting developer identity errors in version control systems," *Empirical Software Engineering*, Jan 2020. [Online]. Available: https://doi.org/10.1007/s10664-019-09786-7

[27] B. Lin, G. Robles, and A. Serebrenik, "Developer turnover in global, industrial open source projects: Insights from applying survival analysis," in *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. IEEE, 2017, pp. 66–75.

[28] M. Zhou and A. Mockus, "Who will stay in the floss community? modeling participant's initial behavior," *IEEE Transactions on Software Engineering*, vol. 41, no. 1, pp. 82–99, 2015.

[29] A. Mockus, "Engineering big data solutions," in *Proceedings of the on Future of Software Engineering*. ACM, 2014, pp. 85–99.

[30] M. Goeminne and T. Mens, "A comparison of identity merge algorithms for software repositories," *Science of Computer Programming*, vol. 78, no. 8, pp. 971–986, 2013.

[31] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan, "Mining email social networks," in *Proceedings of the 2006 international workshop on Mining software repositories*. ACM, 2006, pp. 137–143.

[32] G. Robles and J. M. Gonzalez-Barahona, "Developer identification methods for integrated data from various sources," *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4, pp. 1–5, 2005.

[33] E. Kouters, B. Vasilescu, A. Serebrenik, and M. G. van den Brand, "Who's who in gnome: Using lsa to merge software repository identities," in *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*. IEEE, 2012, pp. 592–595.

[34] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[35] S. Sarawagi and A. Bhamidipaty, "Interactive deduplication using active learning," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 269–278.

[36] F. Hecker, "Setting up shop: The business of open-source software," *IEEE software*, vol. 16, no. 1, pp. 45–51, 1999.

[37] A. Bonaccorsi and C. Rossi, "Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business," *Knowledge, Technology & Policy*, vol. 18, no. 4, pp. 40–64, 2006.

[38] L. Dahlander and M. G. Magnusson, "Relationships between open source software companies and communities: Observations from nordic firms," *Research policy*, vol. 34, no. 4, pp. 481–493, 2005.

[39] C. Rossi and A. Bonaccorsi, "Why profit-oriented companies enter the os field?: intrinsic vs. extrinsic incentives," in *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4. ACM, 2005, pp. 1–5.

[40] N. Munga, T. Fogwill, and Q. Williams, "The adoption of open source software in business models: a red hat and ibm case study," in *Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*. ACM, 2009, pp. 112–121.

[41] J. Henkel, "Selective revealing in open innovation processes: The case of embedded linux," *Research policy*, vol. 35, no. 7, pp. 953–969, 2006.

[42] D. S. Cruzes and T. Dyba, "Recommended steps for thematic synthesis in software engineering," in *2011 International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2011, pp. 275–284.

[43] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[44] V. Braun, V. Clarke, N. Hayfield, and G. Terry, "Thematic analysis," *Handbook of Research Methods in Health Social Sciences*, pp. 843–860, 2019.

[45] T. Yamane, "Statistics: An introductory analysis," 1973.

[46] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[47] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[48] W. Harrison, "An entropy-based measure of software complexity," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, pp. 1025–1029, 1992.

[49] V. Singh, M. Sharma, and H. Pham, "Entropy based software reliability analysis of multi-version open source software," *IEEE Transactions on Software Engineering*, 2017.

[50] O. Temizkan, S. Park, and C. Saydam, "Software diversity for improved network security: optimal distribution of software-based shared vulnerabilities," *Information Systems Research*, vol. 28, no. 4, pp. 828–849, 2017.

[51] A. Homescu, T. Jackson, S. Crane, S. Brunthaler, P. Larsen, and M. Franz, "Large-scale automated software diversity-program evolution redux," *IEEE Transactions on Dependable and Secure Computing*, no. 1, pp. 1–1, 2017.

[52] C. Gini, "On the measure of concentration with special reference to income and statistics," *Colorado College Publication, General Series*, vol. 208, pp. 73–79, 1936.

[53] J. L. Gastwirth, "A general definition of the lorenz curve," *Econometrica: Journal of the Econometric Society*,

pp. 1037–1039, 1971.

[54] M. Goeminne and T. Mens, "Evidence for the pareto principle in open source software activity," in *the Joint Porceedings of the 1st International workshop on Model Driven Software Maintenance and 5th International Workshop on Software Quality and Maintainability*. Citeseer, 2011, pp. 74–82.

[55] L. Morgan and P. Finnegan, "Beyond free software: An exploration of the business value of strategic open source," *The Journal of Strategic Information Systems*, vol. 23, no. 3, pp. 226–238, 2014.

[56] J. Lerner and J. Tirole, "Some simple economics of open source," *The journal of industrial economics*, vol. 50, no. 2, pp. 197–234, 2002.

[57] K. Pepple, *Deploying openstack*. " O'Reilly Media, Inc.", 2011.

[58] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *ICML*, vol. 1, 2001, pp. 577–584.

[59] V. E. Johnson, "Revised standards for statistical evidence," *Proceedings of the National Academy of Sciences*, vol. 110, no. 48, pp. 19 313–19 317, 2013.

[60] M. Sako, "Business ecosystems: how do they matter for innovation?" *Communications of the ACM*, vol. 61, no. 4, pp. 20–22, 2018.

[61] S. Onoue, H. Hata, A. Monden, and K. Matsumoto, "Investigating and projecting population structures in open source software projects: A case study of projects in github," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 5, pp. 1304–1315, 2016.

[62] C. Jonathan and K.-H. Greg, "2017 linux kernel development report," https://www.linuxfoundation.org/2017-linux-kernel-report-landing-page/, 2017.

[63] J. Linåker, P. Rempel, B. Regnell, and P. Mäder, "How firms adapt and interact in open source ecosystems: analyzing stakeholder influence and collaboration patterns," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2016, pp. 63–81.

[64] J. A. Roberts, I.-H. Hann, and S. A. Slaughter, "Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects," *Management science*, vol. 52, no. 7, pp. 984–999, 2006.

[65] A. Barcomb, A. Kaufmann, D. Riehle, K.-J. Stol, and B. Fitzgerald, "Uncovering the periphery: A qualitative survey of episodic volunteering in free/libre and open source software communities," *IEEE Transactions on Software Engineering*, 2018.

[66] A. Capiluppi and M. Michlmayr, "From the cathedral to the bazaar: An empirical study of the lifecycle of volunteer community projects," in *IFIP International Conference on Open Source Systems*. Springer, 2007, pp. 31–44.

[67] K. Crowston and B. Scozzi, "Bug fixing practices within free/libre open source software development teams," *Journal of Database Management (JDM)*, vol. 19, no. 2, pp. 1–30, 2008.

[68] C. Bird, A. Gourley, and P. Devanbu, "Detecting patch submission and acceptance in oss projects," in *Proceedings of the Fourth International Workshop on Mining Software Repositories*. IEEE Computer Society, 2007, p. 26.

[69] R. K. Yin, "Case study research: design and methods," *Journal of Advanced Nursing*, vol. 44, no. 1, pp. 108–108, 2010.

[70] F. Konietschke, L. A. Hothorn, and E. Brunner, "Rank-based multiple test procedures and simultaneous confidence intervals," *Electronic Journal of Statistics*, vol. 6, no. 2012, pp. 738–759, 2012.

[71] B. Vasilescu, A. Serebrenik, M. Goeminne, and T. Mens, "On the variation and specialisation of workload–a case study of the gnome ecosystem community," *Empirical Software Engineering*, vol. 19, no. 4, pp. 955–1008, 2014.

[72] I. Steinmacher, T. Conte, M. A. Gerosa, and D. Redmiles, "Social barriers faced by newcomers placing their first contribution in open source software projects," in *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*. ACM, 2015, pp. 1379–1392.

[73] K. Crowston, K. Wei, J. Howison, and A. Wiggins, "Free/libre open-source software development: What we know and what we do not know," *ACM Computing Surveys (CSUR)*, vol. 44, no. 2, p. 7, 2012.

[74] K. Crowston, H. Annabi, J. Howison, and C. Masango, "Effective work practices for floss development: A model and propositions," in *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*. IEEE, 2005, pp. 197a–197a.

[75] M. Zhou and A. Mockus, "Does the initial environment impact the future of developers?" in *Proceedings of the 33rd International Conference on Software Engineering*. ACM, 2011, pp. 271–280.

[76] ——, "What make long term contributors: Willingness and opportunity in oss community," in *International Conference on Software Engineering*, 2012, pp. 518–528.

[77] W. Oh and S. Jeon, "Membership herding and network stability in the open source community: The ising perspective," *Management science*, vol. 53, no. 7, pp. 1086–1101, 2007.

[78] P. G. Capek, S. P. Frank, S. Gerdt, and D. Shields, "A history of ibm's open-source involvement and strategy," *IBM systems journal*, vol. 44, no. 2, pp. 249–257, 2005.

[79] C. Daffara, "Business models in floss-based companies," in *Workshop presentatioon at the 3rd Conference on Open Source Systems (OSS 2007)*, 2007.

[80] L. Dahlander and M. W. Wallin, "A man on the inside: Unlocking communities as complementary assets," *Research Policy*, vol. 35, no. 8, pp. 1243–1259, 2006.

[81] S. Butler, J. Gamalielsson, B. Lundell, P. Jonsson, J. Sjöberg, A. Mattsson, N. Rickö, T. Gustavsson, J. Feist, S. Landemoo *et al.*, "An investigation of work practices used by companies making contributions to established oss projects," in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*. ACM, 2018, pp. 201–210.

[82] V. Lee, *How firms can strategically influence open source communities*. Springer, 2012.

[83] S. Y. Ho and A. Rai, "Continued voluntary participation intention in firm-participating open source software projects," *Information Systems Research*, vol. 28, no. 3,

pp. 603–625, 2017.

Audris Mockus received the BS degree in applied mathematics from the Moscow Institute of Physics and Technology in 1988, the MS degree in 1991, and the PhD degree in statistics from Carnegie Mellon University in 1994. He is a Harlan Mills Chair Professor of Digital Archeology in the Department of Electrical Engineering and Computer Science, the University of Tennessee. He also continues to work part-time at Avaya Labs Research. Previously, he was in the Software Production Research Department at Bell Labs. He studies software developers' culture and behavior through the recovery, documentation, and analysis of digital remains. These digital traces reflect projections of collective and individual activity. He reconstructs the reality from these projections by designing data mining methods to summarize and augment these digital traces, interactive visualization techniques to inspect, present, and control the behavior of teams and individuals, and statistical models and optimization techniques to understand the nature of individual and collective behavior. He is a member of IEEE and ACM. He can be reached at audris@utk.edu.

Zhi Jin received the BS degree in Computer Science from Zhejiang University, in 1984, a Master Degree Computer Science from Changsha Institute of Technology in 1987, and a PhD Computer Science from Changsha Institute of Technology in 1992. She is currently a professor of Computer Science at Peking University. She is deputy director of Key Lab of High Confidence Software Technologies (Ministry of Education) at Peking University. She is interested in software engineering, requirements engineering, knowledge engineering, and machine learning. She is a senior member of IEEE. Contact her at zhijin@pku.edu.cn.

Yuxia Zhang is a PhD candidate at the School of Electronics Engineering and Computer Science, Peking University. She received the BS degree in software engineering from Northwest University in 2015. Her research interests include Mining Software Repositories and Open Source Software Ecosystems, mainly focusing on the com-

Minghui Zhou received the BS, MS, and PhD degrees in computer science from National University of Defense Technology in 1995, 1999, and 2002, respectively. She is a professor of Computer Science at Peking University. She is interested in software digital sociology, i.e., understanding the relationships among people, project culture, and software product through mining the repositories of software projects. She is a member of the ACM. She can be reached at zhmh@pku.edu.cn.