How Do Companies Collaborate in Open Source Ecosystems? An Empirical Study of OpenStack

Yuxia Zhang
Department of Computer Science and Technology, Peking University
Key Laboratory of High Confidence Software Technologies, Ministry of Education
Beijing, China
yuxiaz@pku.edu.cn

Minghui Zhou*
Department of Computer Science and Technology, Peking University
Key Laboratory of High Confidence Software Technologies, Ministry of Education
Beijing, China
zhmh@pku.edu.cn

Klaas-Jan Stol
School of Computer Science and Information Technology
University College Cork
Lero, the Irish Software Research Centre
Cork, Ireland
klaas-jan.stol@lero.ie

Jianyu Wu
Department of Computer Science and Technology, Peking University
Key Laboratory of High Confidence Software Technologies, Ministry of Education
Beijing, China
sswjy@pku.edu.cn

Zhi Jin
Department of Computer Science and Technology, Peking University
Key Laboratory of High Confidence Software Technologies, Ministry of Education
Beijing, China
zhijin@pku.edu.cn

ABSTRACT

Open Source Software (OSS) has come to play a critical role in the software industry. Some large ecosystems enjoy the participation of large numbers of companies, each of which has its own focus and goals. Indeed, companies that otherwise compete, may become collaborators within the OSS ecosystem they participate in. Prior research has largely focused on commercial involvement in OSS projects, but there is a scarcity of research focusing on company collaborations within OSS ecosystems. Some of these ecosystems have become critical building blocks for organizations worldwide; hence, a clear understanding of how companies collaborate within large ecosystems is essential. This paper presents the results of an empirical study of the OpenStack ecosystem, in which hundreds of companies collaborate on thousands of project repositories to deliver cloud distributions. Based on a detailed analysis, we identify clusters of collaborations, and identify four strategies that companies adopt to engage with the OpenStack ecosystem. We alsofind that companies may engage in intentional or passive collaborations, or may work in an isolated fashion. Further, wefi nd that a company's position in the collaboration network is positively associated with its productivity in OpenStack. Our study sheds light on how large OSS ecosystems work, and in particular on the patterns of collaboration within one such large ecosystem.

KEYWORDS

Open source software; OpenStack; company participation; OSS ecosystem; open collaboration; software development

ACM Reference Format:
Yuxia Zhang, Minghui Zhou, Klaas-Jan Stol, Jianyu Wu, and Zhi Jin. 2020. How Do Companies Collaborate in Open Source Ecosystems? An Empirical Study of OpenStack. In *42nd International Conference on Software Engineering (ICSE '20), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3377811.3380376

1 INTRODUCTION

It is now widely acknowledged that Open Source Software (OSS) has had a dramatic influence on the software industry. These virtual software projects have brought together developers spanning geographic, language, and time zone differences [78]. As open source adoption has grown significantly in the last decade or so, many companies have started participating in OSS projects. Numerous companies have built business models around OSS ecosystems[1] to achieve innovations [40], reduce costs [90], or generate revenue on complementary services [27]. Many companies achieve their goals of joining an OSS ecosystem by hiring developers to contribute to the projects within that ecosystem [9, 92, 98]. Many well-known OSS ecosystems, such as Linux, Android, and OpenStack, are developed mainly through collaborations of many different companies. For example, over 85 percent of code in the Linux kernel has been contributed by more than 500 companies in 2017 [45]. With such a significant level of contribution, companies participating in these

[*] Corresponding author

[1] Similar to Jansen et al. [44], 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.

ICSE '20, May 23–29, 2020, Seoul, Republic of Korea
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7121-6/20/05…$15.00
https://doi.org/10.1145/3377811.3380376

ecosystems have a very significant influence on not only their development roadmap and future, but also on their sustainability.

Collaboration among companies is of critical importance to the development of OSS ecosystems. First, many companies are highly specialized in one specific domain, and their highly specialized knowledge and expertise add great value to an ecosystem [53, 80]. Collaboration among different companies each with specific expertise therefore greatly benefits and facilitates the efficient development of OSS ecosystems. Second, companies may decide to allocate developers to projects—or stop contributing—and these decisions directly affect an OSS ecosystem' sustainability [47, 54, 98]. Third, one company's practices (and behavior) may have a significant impact on others' participation in OSS, whether they are volunteers or companies. For example, a dominating contributing company may dissuade other companies to participate [92]. Given that many Open Source ecosystems have become a critical part of the infrastructure that controls our daily lives, it is imperative to understand how companies interact with and collaborate in these OSS ecosystems.

Previous work has primarily focused on collaboration at the individual level rather than the company level; examples include knowledge sharing through developer interactions [78, 79, 82], coordination of work in globally distributed environments [17, 18], and investigation on how social characteristics affect their onboarding and growth [12, 83, 97]. Research on commercial participation has mainly focused on motivations, business models, and strategies to engage in OSS ecosystems [40, 90, 93, 98]. Only a few studies have investigated the collaboration *between* companies [38, 52, 80, 86]. While they tend to be limited to social network analysis, without seeking to understand the nature of these collaborations and the mechanisms associated with them. Hence, our research goal is to understand how companies collaborate in large open source ecosystems that comprise hundreds of companies and projects. Specifically, we select the OpenStack ecosystem, which offers a platform for cloud-computing infrastructure. The OpenStack ecosystem has thousands of repositories that are co-developed by more than 600 companies in a wide range of domains, including hardware manufacturers, software vendors, system integrators, and consultancyfirms. [2] OpenStack represents a high-potential arena for these companies to play a role in the rapidly evolving cloud computing technology.

In addressing our research goal, we focus on three aspects. First, we are interested in understanding how companies contribute to specific OpenStack projects within the wider ecosystem. Hence, we ask: How do companies participate in the projects of OpenStack (RQ1)? Further, we posit that a relationship exists between companies if they work together on the same project, and we expect patterns to form. Hence, we ask: What collaboration patterns exist within the OpenStack ecosystem (RQ2)? Lastly, prior research has suggested that collaboration may increase developer productivity [12]. Might company collaboration in ecosystems make companies more productive (RQ3)?

Answers to these questions are of great interest to companies as they provide insights into how ecosystems work, and how other firms participate successfully in large ecosystems. Forfi rms aspiring to engage with OSS ecosystems, such insights are invaluable.

To answer these questions, we adopt a mixed-method research approach (i.e., using both quantitative and qualitative methods [16]), drawing on OpenStack's commit history and the widely available online documents and records on OpenStack and the numerous companies that contribute to it. Through a quantitative analysis of the OpenStack commit history, we create a network that represents the collaborations between companies as well as the contributions that companies make to projects. Using cluster analysis, we identify 32 clusters, each representing an *ensemble* of companies and projects that are closely related. To characterize the various collaboration relationships, we define two dimensions: a company's business strategy (which describes how a company creates business with OpenStack) and project category (describing the types of functionality). We qualitatively identify four recurring business strategies, each of which is associated with specific project categories. We also identify several patterns of company collaboration. Some collaborations are *intentional* whereas others are *passive*. Other companies are rather *isolated* within the collaboration network. Finally, using regression analysis, wefi nd statistically significant evidence that a company's collaboration position within the network correlates positively with its productivity in terms of the average number of commits its developers make to the OpenStack projects.

This paper makes methodological, substantive, and theoretical contributions to the literature onfi rm participation in open source projects:

- We propose and demonstrate a methodological framework for studying company participation in OSS projects which uses two-mode social network analysis, clustering technique, and characterization of company collaborations.
- We document a set of companies participating in project combinations, demonstrating different ways in whichfirms contribute to OSS projects within an ecosystem. While specific to the OpenStack, these patterns can provide a theoretical foundation for studies of other ecosystems.
- We document a set of collaboration patterns between companies, which helps to understand the nature of OSS ecosystems and provides a reference for community management.
- We present quantitative evidence for the relationship between the extent offi rm collaboration and their productivity.

In the remainder of this paper, we review related work in Sec. 2, outline our multi-method research approach in Sec. 3, and present the results of our study in Sec. 4. Section 5 discusses the implications for research and practice, as well as threats to validity with suggestions for future work. Section 6 concludes the paper.

2 RELATED WORK

The traditional notion of OSS projects that are driven by voluntary developers is now long outdated. Fitzgerald observed that OSS has become a "commercially viable alternative," presenting a contemporary characterization that he labeled "OSS 2.0" [28]. In the last decade or so, many companies are explicitly defining open source strategies, and strategically invest and engage in OSS projects [39].

Early research on "OSS 2.0" explored why companies adopt open source [18, 43]. Compared to individual developers, companies focus less on social motivations such as reputation and learning

[2] We use the terms 'firm' and 'company' interchangeably in this paper.

benefits but emphasize economic and technological reasons instead [6, 43]. Some studies focused on business strategies around firm participation in OSS [20, 21, 88]. For example, Daffara [20] analyzed 120 firms that derive their main revenue stream from OSS, and classified them into six business strategies, such as twin licensing, platform providers, and consulting. Our recent study discovered eight unique contribution models based on companies' commercial objectives on OpenStack and found they differ in terms of the extent and focus of contributions to 14 types of projects [93]. Other studies have looked at the practices that companies use to implement their strategies [9, 10, 56]. For example, Butler et al. [9] investigated a variety of work practices used by companies to contribute to eight OSS communities, such as employing core project developers, making donations, and joining project steering committees. Further, the impact of commercial participation on OSS has been studied [11, 88, 92, 93, 98]. Zhou et al. observed that a company's control mechanisms and a high degree of involvement are linked to a decrease in new developers joining the project but with improved retention of existing developers [98]. Similar to that, a company's domination is found to be positively associated with the productivity of contributors and the quality of issue reports [92]. More recently, we observed the diversity of contribution models in a project to be associated with the number of volunteers in OpenStack [93].

Knowing why and how *individual* companies participate in OSS, and what impact their participation might bring to an OSS ecosystem is not enough, however, because companies do not operate in isolation when contributing to OSS ecosystems [86]. Company collaboration within ecosystems has largely remained an unexplored topic [26]. Early work on collaboration explored whether companies would collaborate when jointly participating in an OSS ecosystem. Henkel [42] found that companies in the embedded Linux ecosystem revealed a considerable share of their development, and other companies within the ecosystem benefit from their competitors' publicized contributions. Furthermore, Teixeira et al. [84, 86] observed that companies' competition does not necessarily affect their collaboration within an OSS ecosystem. Oručević-Alagićand Höst used network analysis to study company participation in the Android project, highlighting the potential issues caused when a project is dominated by a single company [64].

A few studies address how companies collaborate with each other. For example, Snarby [80] investigated company collaboration in communication channels (e.g., mailing list), and identified three patterns: gatekeeper (i.e., having one person as a representative to navigate code and information flow), secure channel (i.e., using private communication channels to discuss sensitive issues), and open-core collaboration (i.e., contributing all the code they develop to the OSS project's public sources). Finally, some studies focus on the impact of company collaboration on the OSS ecosystem. For example, Duc et al. [25] found that company collaboration can have a positive influence on the time to close reported issues. Linåker et al. [52] found that company collaboration patterns can influence the innovation and time-to-market in the Apache Hadoop ecosystem. Data mining and social network analysis techniques are widely used in these studies to explore the collaboration between companies.

Despite these studies on company collaboration, the reasons for company *collaboration* (as opposed to mere participation) on specific projects remain unclear, because prior work has largely ignored the relationships *between* companies and the projects they jointly contribute to. Further, some companies have close collaborations, while others have little interaction in an OSS ecosystem [52, 86]. The reasons for this phenomenon have not yet been studied. This paper bridges that gap by reporting an empirical study of OpenStack, a large OSS ecosystem with extensive company participation and containing thousands of project repositories. In contrast to prior studies that looked only at companies' collaboration as a social network, we also consider the characteristics of the projects. Moreover, we complement prior studies by uncovering the reasons underpinning several collaboration patterns among companies. We also investigate the correlation of companies' collaboration with their productivity in OSS. Thus, this study is a first attempt to systematically characterize company collaborations and participation in a large OSS ecosystem.

3 STUDY DESIGN

We adopted a mixed-method approach [16] that combines an analysis of the version control history with an examination of the peer-reviewed literature and other online documents. This section introduces OpenStack, which is the ecosystem that we selected for this study (Sec. 3.1). We outline the data collection and cleaning procedures in Sec. 3.2 and data analysis procedures in Sec. 3.3.

3.1 Background to the OpenStack Ecosystem

OpenStack, founded in July 2010 by NASA and Rackspace (a large IT web hosting company [32]), is a collection of OSS projects for building and managing cloud computing platforms for public, hybrid, and private clouds. An OpenStack solution (or distribution) is composed of a number of individual projects to address various components of cloud computing. OpenStack follows a six-month, time-based release cycle [85]. OpenStack components serve a variety of functions, including computing, storage, and networking [14]. By July 2019, OpenStack comprised over 20 million lines of code, contributed by more than 100,000 contributors based in 194 countries, and received support from more than 600 companies [35].

We selected OpenStack for several reasons. First, it is a large ecosystem with thousands of repositories. Second, it is a highly active and mature ecosystem that has been actively developed for almost a decade, ensuring a sufficiently long commit history. The commit data are maintained in GitHub publicly [61] and offer easy access for research analysis. Finally, the ecosystem involves many different types of companies [93] for investigating collaboration; we expected this heterogeneity (i.e., startups, high-tech giants in different sectors) to be a fruitful source for discovering diverse collaborations [86]. Furthermore, OpenStack maintains its contributors' profiles, which can be used to identify companies with high accuracy [93].

3.2 Data Collection and Cleaning

We used OpenStack's version control data to quantify company collaboration. We describe the data retrieval and preparation process next.

Table 1: Statistics of dataset before and after cleaning

	# Projects	# Commits	# Developers	# Companies
Before	1,292	383,664	13,836	n/a
After	1,292	338,035	9,653	602

3.2.1 Version Control Data. OpenStack uses the Git version control system (VCS). We obtained the commit meta-data from GitHub, which provides a mirror of OpenStack's Git repositories [61], by querying GitHub's REST API. The time span of the dataset is from OpenStack's creation date (July 21st, 2010) until January 16, 2019, covering 18 complete releases of OpenStack.

Each commit in the dataset captures author information (full name, email) of the local Git repository to which the commit is made. Prior studies [2, 51, 72, 93] suggest that some commits are submitted by automated bots rather than human developers. We collected these bot accounts identified in prior studies [2, 51, 72, 93] and removed commits submitted by these accounts from our dataset (the list of removed accounts can be found in the online appendix [94]), leaving 338,035 commits for analysis. We cleaned the remaining data for further analysis following the procedures described below. Table 1 summarizes the dataset before and after cleaning.

3.2.2 Merging Multiple Identities. It is not uncommon for developers to have multiple accounts (sometimes with alternative spellings of their name or email) [5, 37, 58, 71, 93]. Thus, it is necessary to merge multiple identifies that belong to the same author. Developer identity merging is a well-known problem [5, 48, 71]. We addressed this problem by using a novel machine-learning method [2], which augments three behavioral 'fingerprints' including time-zone frequencies, the set of files modified, and a vector embedding the commit messages in addition to the author's name and email address. This method has been proved highly-accurate [93].

After applying the technique on 13,836 author identities, 4,183 identities were merged, resulting in 9,653 distinct authors. For the subsequent analyses, we established a unique identity for each single author with or without multiple names and email addresses.

3.2.3 Identifying Affiliations of Developers and Commits. Many firms today have their developers contribute to OSS projects as part of pursuing their business goals [39, 98]. As our study seeks to identify company contributions that are made by individual developers, we first have to accurately identify these developers' affiliations. This is not straightforward because developer affiliations are not directly recorded in Git commits. Many OpenStack developers have changed their jobs and thus their affiliations over time [96]. Similar to our previous work [93], we establish developer affiliation at the time of each commit they made to OpenStack as follows.

Step 1. Identifying developers' affiliations. Developers may have had several affiliations during the time they were contributing to OpenStack. To determine these affiliations and their 'start' and 'end' date for each, we used the OpenStack community member list [33], which provides the individual profiles of its community members. Each profile has an "Affiliations" field, containing all the companies that employed the developer to work on OpenStack and the corresponding time periods for those affiliations.

Table 2: Example of a developer's affiliations

ID	Identities	Affiliation	Start Date	End Date
1	Monty Taylor	Red Hat	2016-06-13	*2019-01-16*
		IBM	2015-08-17	2016-06-12
		HPE	2011-11-21	2015-08-01
		Rackspace	2010-07-06	2011-11-20

We obtained all profiles via a crawler script; if the end date indicated "current," we replaced it with the date of data collection to facilitate further analysis. Developers may use multiple names and email addresses in their commits. We matched a developer profile to an author ID when the author name matched *and* at least one of the email domain matched (with one of the affiliations in the profile). We were able to automatically link profile information to approximately 90% of developers.

For the remaining 10% of developers whose affiliations could not be confirmed (i.e., they could 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," we considered Red Hat as the affiliation. Developers from consumer domains: "gmail.com," "outlook.com," "hotmail.com," etc., were classified as "Volunteer." Consumer domains were identified based on a publicly available list [46], which has been verified and has also been used in other studies [87]. In this process, we determined developers' tenure in each affiliation by considering the range of dates of the commits associated with the email account. Some developers were submitting code using their enterprise email and personal email over the same period. In such cases, we linked all commits from both the enterprise and personal account to the enterprise account, so that the commits made with the personal account would also count as a company contribution rather than a volunteer contribution. Table 2 shows an example of a developer and his affiliations that we determined.

Step 2. Linking affiliation to commits. After identifying the affiliations for developers, along with their start and end date of their tenure with their affiliations, we linked the relevant affiliation to each commit. Specifically, a commit within the tenure of that author with a given company is linked to that company. This ensures that commits are correctly attributed, even when developers are moving from company to company over time.

The final and cleaned dataset covers 1,292 Git repositories, involving more than 600 companies and 9,600 developers (see Table 1).

3.3 Characterizing Company Collaboration

Companies collaborate in OSS ecosystems through several channels, such as mailing lists, issue trackers, and VCS [80]. In this study, we use the commit data produced in VCS to study collaboration, because it represents a clear audit trail of such collaborations. Collaboration exists between two companies if they work together on a project. Similarly, a relationship between two projects exists if they are contributed to by the same set of companies. Such a scenario can be represented as a special kind of two-mode social network that represents the affiliations of a set of actors (i.e., nodes in one mode, representing companies) with a set of "social" events (i.e., nodes in the other mode, representing OpenStack projects) [77]. In a two-mode social network, relationships among nodes in one mode are based on linkages established in the other mode. Social network analysis has been successfully applied to investigate collaboration in several studies, whether on the individual level [5, 55, 78, 83] or on the organizational level [52, 80, 86]. Hence, we deemed this approach appropriate to investigate company collaborations. Companies usually select a subset of projects to contribute to, due to their specific background. For example, SwiftStack [81], powering cloud storage for enterprises, mainly focuses on Swift (a storage project in OpenStack), and its commits contributed to Swift represent 75% of its total commits to OpenStack. Furthermore, the technology stack of OpenStack consists of over one thousand repositories. Some projects cooperate together to offer a complete service, while others may have similar functions that differ in their usage scenarios. For example, Swift and Cinder both provide storage services, but use different units of storage (e.g. 'object' vs. 'block') [34].

Given these types of relationships between projects, the two-mode social network of companies and projects tends to contain clusters. That is, companies with similar or related backgrounds contribute to a group of related projects. Thus, we applied a cluster detection algorithm to the network to identify these clusters of closely related companies and projects.

3.3.1 Discovering Company-Project Clusters. We applied the two-mode social network analysis to demonstrate that hundreds of companies contribute commits to a large number of projects in the OpenStack ecosystem. In this *company-project network*, nodes represent either companies or projects (i.e., two modes); a link between two nodes indicates that a company has contributed to a project. We retained volunteers as a "company" node in this network, because some relationships between companies exist only through volunteer nodes—removing volunteers from the network would reduce this information, which would lead to an incomplete network.

The company-project network exhibits the following properties:

- Bipartite: companies and projects represent disjoint and independent subsets of nodes in the network; each edge connects nodes from different subsets. This is because there are no direct links between two companies or between two projects in this scenario.
- Undirected: even though it seems intuitive to think of commits as directed edges from companies to projects, in this specific network we seek to capture the duality of the relation between the two sets of nodes; that is, we are interested in both the types of projects companies contribute to, and the types of companies that contribute to a project.
- Weighted: by putting weights on the edges, we capture information on how many commits a particular company has submitted to the same project (for our analysis we need to take into account not only to which projects companies contributed, but also their exact commit distribution among those projects).

More formally, the company-project network can be represented as a graph G = (C, P, E), where C is the set of companies, P is the set of projects, and E is the set of edges. An edge $e \in E$ between a company $c \in C$ and a project $p \in P$ exists if c has contributed commits to p, weighted according to the number of commits.

As groups of companies contribute to specific subsets of projects, a clustering emerges; within each cluster, the density of links (representing the volume of contributions) is higher than the density of links between nodes in other clusters. To identify those clusters, we used the Bi-Louvain algorithm, a greedy algorithm based on network modularity for two-mode graphs [95], on the graph G to group together closely-related nodes.

As companies join and leave an OSS ecosystem over time, the ecosystem evolves. We selected the commit history generated during the production of the 14th release of OpenStack, because this release has the highest number of participating companies compared to other releases. The generated graph contains a total of 1,067 nodes (of which 250 represent companies and 817 represent projects). Among those nodes, the 4,264 edges represent contribution relationships between company nodes and project nodes. Using the Bi-Louvain algorithm we identified 32 collaboration clusters. The *modularity* of a graph is a measure to quantify the strength of partitions [59]. Its value falls in the range $[-1, 1]$, and is positive whenever the fraction of edges falling within the same cluster is higher than expected on the basis of chance. Good modularity values typically lie in the range between 0.3 to 0.7 [59]. The modularity of our clustering result is 0.51, indicating a good partition.[3]

3.3.2 Characterizing Company-Project Clusters. For each cluster of companies and projects, we seek to understand why the companies contribute to these projects, and how they collaborate with one another. By investigating earlier studies of company involvement in OSS, online documents pertaining to OpenStack, and OpenStack's version control history, we identified two dimensions along which company participation can be positioned: business strategy and project category. Business strategy always drives companies' actions when they participate in OSS ecosystems [73, 93, 98]. The category of projects helps in understanding why companies contribute to projects. We discuss these two dimensions next.

Business strategy. Business strategy refers to a company's motivation to join an OSS ecosystem [6, 43]. To identify the strategy for a company, we conducted Internet searches (using "OpenStack" and the company's name as keywords) and inspected the first 20 results. We also collected documents from the marketplace page on the official OpenStack website [35] regarding the products, services, or solutions offered by companies. We analyzed these records to identify a series of categories of strategies by using thematic analysis, a widely used technique for identifying and recording "themes" in textual documents [7, 8, 19]. The process involved 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 to support these steps. To increase the reliability of this analysis, the first two authors independently performed steps 1 to 4 [74]. After this, we held a number of meetings to resolve any disagreements and to finalize the set of themes (step 5). If the first two authors

[3] Because measuring small clusters has its uncertainty [30], we also applied an alternative metric, i.e., conductance [50], to quantify the strength of the partitions. The average conductance is 0.20, indicating a good clustering.

failed to reach an agreement on a code or theme, a third author acted as an arbitrator. This happened on four occasions during the labeling of the 32 core companies' business strategies.

Project category. Project category is used to group a set of projects, which either collaborate with each other to offer a complete service or provide similar functions but differ in details regarding usage scenarios. We manually looked for the documents (created and maintained by OpenStack Foundation) and the READMEfi le of projects to determine their category. Following the same set of steps of thematic analysis described above, we identified the categories of projects in OpenStack.

4 RESULTS

4.1 Company-Project Clusters in OpenStack

Fig. 1 shows the 32 clusters we detected in the company-project network based on the commit data produced in the development of the 14th OpenStack release. The density of edges between nodes *within* each cluster is significantly higher than the density of edges between nodes belonging to different clusters. Specifically, the total weight of edges connecting companies and projects within the clusters amounts to 23,729 commits, representing over 69% of the total number of 34,192 commits. This high degree of cohesion and low coupling suggest a high quality of the resulting clustering. The size of the clusters varies greatly, ranging from two nodes (one company, one project) to 235 nodes (60 companies, 175 projects).

For each cluster, we investigated its companies and projects and their relationships along the two dimensions described in Sec. 3.3.2. Given the very large number of companies and projects within the OpenStack ecosystem, we limited our analysis to a sample of them. Specifically, for each cluster we selected the company that contributed the most commits—together, these companies contributed 83.7% of commits. We refer to these companies as *core companies.*[4]

Of the 32 core companies, we identified four types of business strategies that motivate them to participate in OpenStack: Full Solution (FS), Partial Solution (PS), Business Integration (BI), and Complementary Services (CS), which are consistent with the primary commercial models discovered in our earlier study [93]. Table 3 describes these strategies, including representative examples of companies, and the number of core companies that we classified using these four categories. Furthermore, we identified 26 project categories based on their functionality which cover the 14 project types discovered in [93] but with an improved granularity. Due to space constraints, we include 10 representative categories that account for the most commits in at least one cluster (see Table 4). (The full list of the 26 categories is available in the online appendix [94].)

After investigating the core company (in terms of business strategy) and the projects (functionality category) of each cluster, we observed several typical company-project combinations in the development of an OpenStack release. We discuss these combinations next.

The most common combination is that companies contribute to plugins or drivers to integrate their own business with OpenStack (BI): the projects in 14 among 32 clusters (about 44%) are mostly plugins or drivers (which occupy about 79% of projects in a cluster). These clusters are relatively small, with 2 to 13 nodes. Among them, 12 clusters' core companies share the BI strategy. Further, the core companies of the remaining two clusters are small cloud computing companies (i.e., Axilera and Cloudbase Solutions). One offers complementary services (CS) to help other companies integrate with OpenStack, whereas the other provides cloud computing services targeting enterprises using the Windows platform by integrating OpenStack with Windows-based infrastructures. Thus, these two companies' contribution interests coincide with the companies holding the BI strategy. This pattern reflects the popularity of OpenStack and related cloud technologies.

The second typical combination represents companies contributing to deployment tools. There are eight clusters (25%) in which the majority of projects are related to deployment tools, and the core companies share the same strategy (FS), i.e., making profit through providing cloud computing service based on OpenStack. As pointed out by prior studies [15, 67], a primary problem faced by OpenStack is how to deploy various cloud services in production environments. This is a critical feature and might explain why companies make contributions to deployment tools within the ecosystem.

The third combination is centered on large IT companies in large size (with over 100 nodes). There are three clusters like that, which have IBM, Huawei, and Fujitsu as their core companies, respectively. These clusters involve many other companies and diverse projects, reflecting the extensive participation of the leading large companies and their partners or followers. It also suggests that, in addition to plugins/drivers and deployment tools, there are types of projects which are closely related to each other and attract a wide range of companies to participate, forming a sub-collaborative network.

The remaining seven clusters are small, with up to 10 nodes, revealing a pattern in which the core companies make contributions to a specific project, adding new services to OpenStack beyond enhancements to deployment, documentation, or plugins/drivers. In three clusters, the core companies provide partial solutions (PS) based on a specific project in OpenStack. The projects in these three clusters (categorized as "Data analytic," "Networking," and "Optimization/policy tools," respectively) are consistent with their core companies' business strategy (see Table 4 for brief descriptions of these categories). For example, the core company of one cluster is "Tesora," a small open source company delivering a database service. The corresponding project in OpenStack is "Trove," belonging to the category "Data Analytic." In the 14th release, approximately 94% of Tesora's commits to OpenStack are focused on this project. Three other clusters (of the seven small ones) have core companies delivering full solutions (FS) and incubating new services to the OpenStack ecosystem. For example, Platform9 systems (a cloud computing company [69]) developed "Mors," which automatically handles resources that are no longer needed [68]. The last cluster consists of a core company (i.e., CCIN2P3, a small cloud computing consulting company) delivering complementary services (CS), and an "Orchestration" project. These observations suggest a phenomenon of "unconscious" and uncoordinated division of labor in the OSS ecosystem.

[4] For the only cluster where volunteers contributed the most commits (56%), we selected the company whose commits (28%) second only to volunteers as its core company.

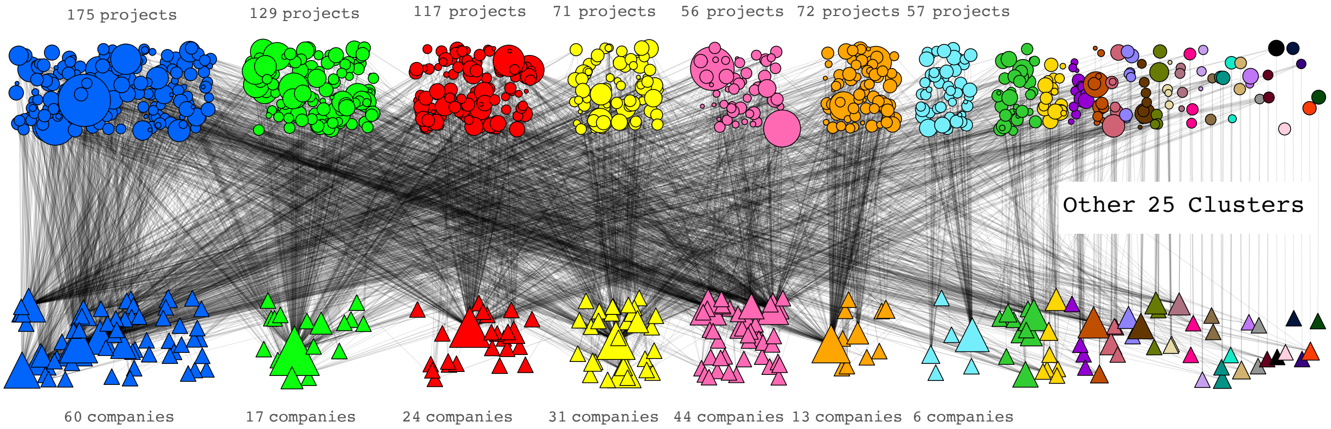

Figure 1: Two-dimensional representation of the company-project network. The position of company and project nodes along the horizontal axis is set according to the clusters they are assigned to, to avoid edge crossing and to demonstrate the effectiveness of the clustering method. Nodes of the same color belong to the same cluster. The size of nodes is proportional to their degree centrality, and the edge thickness is proportional to their associated weight.

Table 3: Four business strategies of company participation in OpenStack projects

Business strategy	Description	# Companies	Example
Full Solutions (FS)	Providing full cloud solutions to users, including private/ public/ hybrid cloud services, deployment, and maintenance services, etc.	15	Rackspace
Partial Solutions (PS)	Providing solutions to users only on the basis of one or two project(s) in OpenStack.	3	Tesora
Business Integration (BI)	Integrating OpenStack with their own business	12	Intel
Complementary Services (CS)	Providing complementary services, e.g., consulting and training services around OpenStack.	2	CCIN2P3

Summary for RQ1: Several typical combinations occurred while companies participate in the projects of OpenStack. Companies holding the BI strategy often contribute to plugins or drivers to integrate their own business with OpenStack. Companies making profit through FS often contribute to deployment tools. The clusters centered on large IT companies involves extensive companies and diverse projects. Companies that select specific projects to contribute to are driven by their particular business, focusing their business on the basis of one project, or incubate new projects.

4.2 Company Collaboration within OpenStack

We found that different companies contribute to a common set of projects, appearing in the same company-project cluster. There might be special causes of this phenomenon. To get a deeper understanding, we manually looked at the background of the *pairs* of companies that appeared in the same cluster, and the functions of the projects they contributed to jointly. Specifically, of the 32 clusters, 19 have more than one company, ranging from 2 to 60 companies. For each of these 19 clusters, we analyzed the collaborations between the cluster's core company and other companies within that cluster. In total, we identified 217 pairs of companies. In addition to understanding the business strategy of each pair of companies, we also conducted Internet searches based on their names, the term "OpenStack," and the projects (or project category) they are involved in. To ensure relevance and reliability of the retrieved records, we only included the information released by the companies or the OpenStack Foundation during the development of the 14th release. For the remaining 13 clusters that only contain a single company, we mainly focus on understanding why these companies do not cooperate with other companies in the same way. We obtained 67 online records and analyzed them following the steps for thematic analysis outlined in Sec. 3.3. Some records explicitly suggested that companies had started collaborations with other companies, indicating intentional collaboration. We label these collaborations as *intentional*. If we could notfi nd records for a pair of companies, we labeled their relationship *passive*. Furthermore, we also observed that some companies make *isolated contributions* to some projects. We discuss these three categories next.

Table 4: Ten categories of projects in the OpenStack ecosystem representing the most commits

Project category	Description	# Projects	Example
Computing	To implement services and associated libraries regarding computing.	17	Nova
Networking	To provide capabilities for managing dynamic host configuration protocol (DHCP), static Internet protocols, or virtual area networks.	73	Neutron
Deployment	To deploy OpenStack production environments, and make it scalable to operate and upgrade.	328	Kolla
Orchestration	To provide interface and tools for the management of OpenStack services.	52	Heat
Billing	To provide billing solution by collecting data generated using cloud services and applying rating rules to calculate prices.	7	Cloudkitty
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.	22	openstack-manuals
Architecture Optimization	To optimize the code architecture by developing and sharing common libraries.	39	oslo
Data Analytic	To implement services and libraries about database, data processing and searching, to realize workload balancing.	20	Trove
Optimization/Policy Tools	To provide services for performance analysis and optimization of OpenStack.	19	Congress
Plugin/Driver	To facilitate integration of OpenStack and other hardware or software infrastructure.	15	Starlingx

4.2.1 *Intentional Collaboration.*

Intentional collaboration refers to a collaboration between companies that they actively seek out and pursue. We identified three patterns: 1) supply and consumption, 2) distribution-oriented ally, and 3) service delegation.

Supply and Consumption. Companies, using OpenStack in their production environment, tend to collaborate with their OpenStack suppliers on the deployment projects, which are incubated and driven by the supply companies. This pattern was found in eight pairs of companies from six clusters. For instance, Walmart (a multinational retail company) relies on Rackspace for their OpenStack solution [89]. More than 96% (150 out of 156) of Walmart's commits to the 14th release of OpenStack are focused on "Ansible," which is funded and dominated by Rackspace [49, 92].

Distribution-oriented Ally. Some companies provide an OpenStack-based solution by seeking collaboration with one company that has maintained a widely-used OpenStack distribution. The primary target of this collaboration pattern is to make one side's infrastructure compatible with the other party's OpenStack distribution. This pattern was found in six pairs of companies from four clusters. For instance, OVH (a French company offering web, dedicated, and cloud hosting solutions [65]) has deployed the OpenStack distribution of Red Hat in its data centres [41], therefore it mainly collaborates with Red Hat on the deployment of Red Hat's OpenStack distribution.

Service Delegation. We also found companies that provide complementary services around OpenStack to promote their partners' business towards OpenStack. This pattern was found in two pairs of companies from two clusters. For example, Axilera, who provide system integration services to high tech companies embarking on the development of networking and internet-connected products, helps Broadcom to integrate its "Broadview" with OpenStack [3]. Hence, all of Axilera's commits were on "Broadview" related repositories. Another example is the collaboration between Cisco and OneCloud, i.e., Cisco entrust OneCloud to provide consultation and training for its OpenStack-based cloud solution [60]. Thus, all of Onecloud's commits are contributed to "networking-cisco," which provides support for Cisco networking in OpenStack [62].

4.2.2 *Passive Collaboration.*

Passive collaboration refers to a collaboration consisting of companies contributing to a common set of projects purely for their own interest without explicit coordination. Companies may provide or use services based on those projects. For example, both Intel and IBM offer compute services in their cloud solutions, so they both make contributions to "Nova," a computing project in OpenStack. Companies may be users and providers of an OpenStack service, but a supply chain does not exist between them. For instance, both HP Enterprise (a large cloud computing provider) and CERN (a user of OpenStack) contributed to the "Magnum" project (providing container management service for OpenStack), but CERN is not a consumer of HP's OpenStack solution [4]. Further, we found that the central projects (including the categories computing, networking, and storage) always attract more companies with various backgrounds, including cloud providers, hardware manufacturers, software vendors, system integrators, and consultancy corporations. Based on the assumption that this project is becoming increasingly better, these companies contribute to improve these core projects and effectively have formed a passive collaboration.

4.2.3 Isolated Contribution. We also identified a number of companies that were the sole contributor to a project as introduced in Sec. 4.1. In other words, these companies do not collaborate (or very little) with others despite the many collaboration opportunities offered when being part of a large and active OSS ecosystem that involves many companies. Of the 32 clusters, 13 fell into this category. When studying the characteristics of the companies and projects in those clusters, we observe that these companies have a very specific and specialized interest in OpenStack, and the projects that they contribute to are of limited interest to other companies. The specific interests of a company include: 1) integrating its infrastructure with OpenStack and 2) creating a new service. The projects they work on tend to fall in the category "plugins/drivers." These projects can be quite specific to companies, which is why other companies may not participate in co-developing these. For example, the project "fuel-plugin-xenserver," used to deploy OpenStack on XenServer (Citrix's server virtualization platform [1]), is completely developed by Citrix. Other such projects include new service projects. Since they tend to be in an initial incubation stage, their popularity tends to be relatively low, and their sustainability is highly uncertain. Thus, very few other companies tend to get involved in these projects. These observations suggest that projects that are created by companies with specific needs do not attract contributions from other companies.

Summary of RQ2: Most companies collaborate with others, even with their competitors. Some collaborations are actively and intentionally pursued whereas others are passive and incidental. Some companies may still be the sole contributor to some projects, in particular when those are specific to that company's base product or unique interest.

4.3 Company Collaboration and Productivity

As we suggested in Sec. 4.2, some companies collaborate with others to achieve their business strategy of joining OSS. Meanwhile, other companies contribute to specific projects alone and barely collaborate with others. With limited staff and time resources, all companies seek to maximize their employees' productivity [70]. Prior studies have found that collaboration can increase the productivity of developers [12, 22, 83]. Hence, it is of interest to investigate the relationship between companies' productivity and their position in the company collaboration network.

In this study, we consider companies that contribute to the same project to be collaborators, whether *intentional* or *passive* as we have labeled them (see Sec. 4.2). To measure a company's degree of collaboration, we transform the company-project network described in Sec. 3.3.1 into a company-company network by removing all project nodes and connecting company nodes if they have made contributions to the same projects. We then measure a company's collaborations in the network with *degree centrality*, which is a widely used measure to identify nodes that have more influence than others in a social network [24, 63]. For each node in the network, its degree centrality (DC) is defined as the number of edges, or links, it has to other nodes, and normalized by dividing by the maximum possible degree i.e., the number of nodes minus one [36]. We define a company's *productivity* as a ratio of the number of commits submitted by their employees to the number of their employees that have committed. We applied this measure to each of the individual release periods and obtained 2,629 observations. A preliminary investigation of these observations reveals that some companies submitted an exceptionally large amount of commits during a release for special reasons, e.g., developers in a company share an account to submit code [80] and companies open their projects to OpenStack with original commits. Using the *R* package `boxplot.stats` [23] combined with a manual check, we identified and removed 265 outliers. The lowest productivity was 1.0 for 559 of the company/release combinations, while the highest was 28.5 for Rackspace in the first release. The median was 6.22.

Based on previous studies [98], we suggest that productivity may be affected by several factors, and in particular we argue that productivity may change over time as a company gains experience and builds a reputation and gains credibility within the community, leading to a higher level of productivity. Hence, we include *release* (mentioned in Sec. 3.2) as a predictor in the regression model with the response being the productivity of a company (*Company_Productivity*). Additional considerations regarding conditional independence and suitability of linear models are discussed in Sec. 5. The final regression equation is:

$$\mathit{Company\_Productivity} \sim \mathit{Degree\_Centrality} + \mathit{Release}$$

The results of the fitted model are shown in Table 5. The adjusted R^2 of the model is 0.19. While this suggests that other factors may play a role, considering the exploratory nature of this study, this result also suggests that degree centrality is a relevant factor in explaining a company's productivity within an open source ecosystem. The coefficient of the predictor *Degree_Centrality* is 11.8 and is statistically significant with an extremely small p-value. We have included the *Release* (from Release 1 to Release 18, a categorical variable) as a so-called *nuisance* parameter, since productivity may vary for each release. We found that none of the releases are statistically significant ($p > .05$), implying that the time factor has no significant effect on the productivity of companies. For illustration purposes, Table 5 also shows the coefficients for two of the 18 releases, i.e., Release 2 and Release 3 (as compared to Release 1).

The positive coefficient of *Degree_Centrality* indicates that a company collaborating more with other companies tends to have higher productivity under the same time and staff constraints. A possible explanation for the observed effect is that company collaboration allows a company to transfer its needs and resources, and obtain information and help from other companies in an easier

Table 5: Coefficients of the model (*n*=2,364). Adjusted R^2=19%

	Estimate	Std.Err	Pr(>|t|)
(Intercept)	5.14	2.35	0.0288
Degree_Centrality	11.8	0.531	0
Release 2	−5.05	2.80	0.0723
Release 3	−1.10	2.83	0.697

way. Therefore, companies can achieve their strategies based on OpenStack in a more efficient way.

Summary for RQ3: The position of companies in the collaboration network is positively associated with their productivity in OpenStack.

5 DISCUSSION

5.1 Company Collaboration in OSS Ecosystems

As many open source projects grow in size, importance, and complexity, many come to depend on companies for support and contributions, eitherfi nancially or through seconded staff[29, 76]. It is crucial to understand how companies collaborate with each other (and volunteers) to make an OSS ecosystem well-behaving.

A company's contribution strategy is "hidden" in its developers' commits, which is mixed in with millions of other commits to projects within an OSS ecosystem. It is challenging to understand its intention in the OSS ecosystem, along with the impact that might have on the ecosystem. In this study, we analyzed how companies participate in the OpenStack ecosystem by identifying clusters. We interpret each cluster using two dimensions: a company's strategy and a project's functionality category within the ecosystem. We obtained several combinations of companies and projects they participated in. These company-project combinations offer a picture of how hundreds of companies participate in as many projects, forming an ecosystem that delivers a complex product. We observed that the companies' business strategies on OpenStack are also found in other OSS ecosystems [86, 98]. This suggests that the identified combinations of companies and projects could be used as a reference for other OSS ecosystems. Furthermore, we also identified several collaboration patterns by analyzing the relationship among "triplets" consisting of two companies and the project they both participated in. The characteristics of those collaboration patterns uncover company interactions and expand our understanding of the nature of OSS ecosystems with company involvement.

The two-mode social network, relevant clustering techniques that model companies contributing to projects, and the two dimensions we proposed can be used by companies who wish to reflect on their open source engagement and strategy, and if not explicitly stated, to help articulate such a strategy. For example, a company that seeks to create business opportunities with OpenStack software, i.e., providing packaged software-based solutions, it could review the way other companies engage within the OpenStack ecosystem. An OSS foundation works as a collaboration "enabler" between the OSS community and company contributors [66]. For leaders of an OSS community, such a foundation can identify and evaluate its projects' activity level, for example, to establish which kinds of projects are popular, and which do not attract participation from companies. Such insights facilitate making strategic decisions more confidently. Awareness of common collaboration patterns can also help companies to participate more effectively in OSS; ourfi ndings offer descriptive insights about different collaboration strategies. Firms can refer to different ways of co-creating via collaboration when they participate in such an ecosystem.

5.2 Productivity of Companies Involved in OSS

Both time and workforce are essential to companies who continuously seek to achieve their strategic goals with fewer resources. OSS communities, too, aim to improve developer efficiency [57, 96]. This study presents evidence for a significant and positive association between companies' degree of collaboration and their productivity (we discuss limitations regarding the operationalization of productivity in Sec. 5.3). For companies, actively building collaboration seems to be a good practice when participating in OSS ecosystems. On the other hand, OSS communities and foundations may wish to regulate company participation at the macro level in order to achieve efficient and sustainable development, in particular, paying more attention to the projects that only singular companies contribute to. However, the definitive reason for a strong relationship between companies' collaboration and productivity remains unclear—developing a better theory that explains this link is an avenue for future work. For example, additional factors for a regression model or conduct qualitative studies at companies that participate in OSS ecosystems.

Before ecosystems and companies can benefit from one another, it is important to understand how the ecosystem is shaped and how it evolves. The types of companies that get involved in the ecosystem, the categories of projects that are created in the ecosystem, the patterns of different companies participating in different projects, and the collaboration patterns among companies, are all key elements that shape the evolution of the ecosystem, and which could be monitored to improve the sustainability of OSS ecosystems.

5.3 Threats to Validity

We discuss threats to the validity of our study following common guidelines for empirical studies [74, 91].

5.3.1 *Construct Validity.*

Establishing measures that reflect companies' participation to OSS projects remains to be a challenge, and is a trade-off between what one wishes to measure and what one *can* measure. To address this problem, we studied the related literature and the development process of OpenStack. After much consideration, we decided to use commits as an estimate for several reasons: (1) they reflect a contribution that is "validated" through the peer review process of the project; (2) many OSS communities enforce contributors to split up large commits into separate commits with only one "logical change" [31], and the distribution of commit size in different companies does not show statistical differences; (3) commits are commonly used for characterizing companies' contributions; (4) they are simple to calculate [93]. A prior study [38] indicates that developers from companies in OpenStack agreed with this approach to estimate their contributions. Future work could consider other types of contributions, such as participation in online discussions, and reviewing code changes. Furthermore, the collaboration between companies could also be explored by studying other interaction channels, such as IRC, mailing list, and issue trackers, in addition to submitting commits to the same projects. This could be a topic for future studies. Finally, measuring productivity in software engineering is challenging and has been a topic of much discussion [75]. In this study, we defined productivity as the number of commits submitted per contributing employee from a company during a release. Despite its widespread use [13, 79], using this measure has drawbacks [96]. For example, the effort required for commits can vary greatly, from adding a new feature that might take hundreds of lines of code, to changing a single typographic error. To gain a deeper understanding of the association of companies' collaboration and their productivity, future studies could consider a more precise measurement of productivity.

We investigated the variables' distribution to fit the regression models for RQ3 to detect outliers. We also attempted to adjust for a number of factors that could bias our results. One remaining concern is that observations of company productivity may not be independent, as they might relate to company background, e.g., whether a company is primarily focused on software development, cloud computing, or even open source development. We fit a random effects model that includes Company-ID (to ensure conditional independence). The effect of collaboration degree still points in the same direction and is as significant. (Details in the appendix [94]).

5.3.2 External Validity. We purposely selected the OpenStack ecosystem as it is a good representative of large and active ecosystems with intense involvement from many companies. We selected the 14th release of OpenStack to carefully elaborate the company-project network. Yin [91] emphasized that case studies are generalizable to theoretical propositions and not to populations or universes. The method we used to investigate commercial collaboration in OSS, e.g., network analysis and two dimensions (one on company aspect and one on project aspect) can be used to identify more commercial collaboration patterns in other OSS ecosystems. Further, companies can reflect on their role and position within OSS ecosystems and draw on the collaboration patterns identified in this study to evaluate and further develop their collaboration network and strategy. The positive association between company collaboration and their productivity observed in this study may be a 'wake-up signal' for the companies that wish to play a role in OSS but have limited resources. Collaborating with other companies may help them to achieve their goals.

6 CONCLUSION

While there is increasing attention for firm participation in OSS communities, few studies have studied collaboration patterns within OSS ecosystems that comprise many projects. This paper seeks to address this gap by presenting the results of an empirical study of the OpenStack ecosystem which is a highly popular cloud computing platform. This paper proposes and demonstrates a methodological framework for studying company participation and collaboration. We identify different engagement strategies that companies employ to participate in the OpenStack, and we characterize company collaboration patterns. Our study contributes to the understanding of how different companies contribute to different projects and collaborate with each other in an OSS ecosystem. As company participation in OSS ecosystems is becoming the industry norm, these results may help companies to define their own OSS strategy.

ACKNOWLEDGMENTS

This work is supported by the National key R&D Program of China Grant 2018YFB1004201, the National Basic Research Program of China Grant 2015CB352200, the National Natural Science Foundation of China Grants 61825201 and 61690200, and Science Foundation Ireland Grant 15/SIRG/3293 and 13/RC/2094.

REFERENCES

[1] Gohar Ahmed. 2013. *Implementing Citrix XenServer Quickstarter.* Packt Publishing Ltd.

[2] Sadika Amreen, Audris Mockus, Russell Zaretzki, Christopher Bogart, and Yuxia Zhang. 2020. ALFAA: Active Learning Fingerprint based Anti-Aliasing for correcting developer identity errors in version control systems. *Empirical Software Engineering* (03 Jan 2020). https://doi.org/10.1007/s10664-019-09786-7

[3] Axilera. 2019. A comprehensive offering of software and system integration service. http://www.axilera.com/projects.html.

[4] Tim Bell, B Bompastor, S Bukowiec, J Castro Leon, MK Denis, J van Eldik, M Fermin Lobo, L Fernandez Alvarez, D Fernandez Rodriguez, A Marino, et al. 2015. Scaling the CERN OpenStack cloud. In *Journal of Physics: Conference Series*, Vol. 664. IOP Publishing, 022003.

[5] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, and Anand Swaminathan. 2006. Mining email social networks. In *Proceedings of the 2006 international workshop on Mining software repositories.* ACM, 137–143.

[6] Andrea Bonaccorsi and Cristina Rossi. 2006. Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business. *Knowledge, Technology & Policy* 18, 4 (2006), 40–64.

[7] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[8] Virginia Braun, Victoria Clarke, Nikki Hayfield, and Gareth Terry. 2019. Thematic analysis. *Handbook of Research Methods in Health Social Sciences* (2019), 843–860.

[9] Simon Butler, Jonas Gamalielsson, Bjorn Lundell, Christoffer Brax, Johan Sjoberg, Anders Mattsson, Tomas Gustavsson, Jonas Feist, and Erik Lonroth. 2019. On Company Contributions to Community Open Source Software Projects. *IEEE Transactions on Software Engineering* (2019).

[10] Simon Butler, Jonas Gamalielsson, Björn Lundell, Per Jonsson, Johan Sjöberg, Anders Mattsson, Niklas Rickö, Tomas Gustavsson, Jonas Feist, Stefan Landemoo, et al. 2018. An investigation of work practices used by companies making contributions to established OSS projects. In *Proceedings of the 40th International Conference on Software Engineering (SEIP).* ACM, 201–210.

[11] Andrea Capiluppi, Klaas-Jan Stol, and Cornelia Boldyreff. 2012. Exploring the role of commercial stakeholders in open source software evolution. In *IFIP International Conference on Open Source Systems.* Springer, 178–200.

[12] Casey Casalnuovo, Bogdan Vasilescu, Premkumar Devanbu, and Vladimir Filkov. 2015. Developer onboarding in GitHub: the role of prior social links and language experience. In *Proceedings of the 2015 10th joint meeting on foundations of software engineering.* ACM, 817–828.

[13] Antonio Cerone. 2013. Learning and activity patterns in OSS communities and their impact on software quality. *Electronic Communications of the EASST* 48 (2013).

[14] OpenStack Community. 2019. OpenStack Documentation. https://docs.openstack.org/.

[15] Hélène Coullon, Christian Perez, and Dimitri Pertin. 2017. Production deployment tools for IaaSes: an overall model and survey. In *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud).* IEEE, 183–190.

[16] John W Creswell and J David Creswell. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches.* Sage publications.

[17] Kevin Crowston, Hala Annabi, James Howison, and Chengetai Masango. 2005. 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, 197a–197a.

[18] Kevin Crowston, Kangning Wei, James Howison, and Andrea Wiggins. 2012. Free/Libre open-source software development. *Acm Computing Surveys* 44, 2 (2012), 1–35.

[19] Daniela S Cruzes and Tore Dyba. 2011. Recommended steps for thematic synthesis in software engineering. In *2011 International Symposium on Empirical Software Engineering and Measurement.* IEEE, 275–284.

[20] Carlo Daffara. 2007. Business models in FLOSS-based companies. (2007).

[21] Linus Dahlander and Mats Magnusson. 2008. How do firms make use of open source communities? *Long range planning* 41, 6 (2008), 629–649.

[22] Daniela Damian and James Chisan. 2006. An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management. *IEEE Transactions on Software Engineering* 32, 7 (2006), 433–453.

[23] Robert Dawson. 2011. How significant is a boxplot outlier? *Journal of Statistics Education* 19, 2 (2011).

[24] Reinhard Diestel. 2005. Graph theory. 2005. *Grad. Texts in Math* 101 (2005).

[25] Anh Nguyen Duc, Daniela S Cruzes, Claudia Ayala, and Reidar Conradi. 2011. Impact of stakeholder type and collaboration on issue resolution time in oss projects. In *IFIP International Conference on Open Source Systems.* Springer, 1–16.

[26] Anh Nguyen Duc, Daniela S Cruzes, Geir K Hanssen, Terje Snarby, and Pekka Abrahamsson. 2017. Coopetition of software firms in Open source software ecosystems. In *International Conference of Software Business.* Springer, 146–160.

[27] Joseph Feller and Brian Fitzgerald. 2000. A framework analysis of the open source software development paradigm. In *Proceedings of the twentyfirst international conference on Information systems*. 58–69.

[28] Brian Fitzgerald. 2006. The transformation of open source software. *Mis Quarterly* (2006), 587–598.

[29] Darren Forrest, Carlos Jensen, Nitin Mohan, and Jennifer Davidson. 2012. Exploring the role of outside organizations in Free/Open Source Software projects. In *IFIP International Conference on Open Source Systems*. Springer, 201–215.

[30] Santo Fortunato and Marc Barthelemy. 2007. Resolution limit in community detection. *Proceedings of the national academy of sciences* 104, 1 (2007), 36–41.

[31] OpenStack Foundation. 2019. Git Commit Good Practice. https://wiki.openstack.org/wiki/GitCommitMessages.

[32] OpenStack Foundation. 2019. Introduction: A Bit of OpenStack History. https://docs.openstack.org/project-team-guide/introduction.html.

[33] OpenStack Foundation. 2019. OpenStack Foundation: Member Directory. https://www.openstack.org/community/members/.

[34] OpenStack Foundation. 2019. OpenStack Services. https://www.openstack.org/software/project-navigator/openstack-components.

[35] OpenStack Foundation. 2019. OpenStack Website. https://www.openstack.org/.

[36] Linton C Freeman. 1977. A set of measures of centrality based on betweenness. *Sociometry* (1977), 35–41.

[37] Mathieu Goeminne and Tom Mens. 2013. A comparison of identity merge algorithms for software repositories. *Science of Computer Programming* 78, 8 (2013), 971–986.

[38] Jesus M Gonzalez-Barahona, Daniel Izquierdo-Cortazar, Stefano Maffulli, and Gregorio Robles. 2013. Understanding how companies interact with free software communities. *IEEE software* 30, 5 (2013), 38–45.

[39] Jesus M Gonzalez-Barahona and Gregorio Robles. 2013. Trends in free, libre, open source software communities: From volunteers to companies. *it–Information Technology it–Information Technology* 55, 5 (2013), 173–180.

[40] Dietmar Harhoff, Joachim Henkel, and Eric Von Hippel. 2003. Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. *Research policy* 32, 10 (2003), 1753–1769.

[41] Red Hat. 2017. OVH Named a Red Hat Certified Cloud and Service Provider. https://www.redhat.com/en/about/press-releases/ovh-named-red-hat-certified-cloud-and-service-provider.

[42] Joachim Henkel. 2003. Software development in embedded Linux—Informal collaboration of competing firms. In *Wirtschaftsinformatik 2003/Band II*. Springer, 81–99.

[43] Joachim Henkel. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy* 35, 7 (2006), 953–969.

[44] Slinger Jansen, Anthony Finkelstein, and Sjaak Brinkkemper. 2009. A sense of community: A research agenda for software ecosystems. In *2009 31st International Conference on Software Engineering-Companion Volume*. IEEE, 187–190.

[45] Corbet Jonathan and Kroah-Hartman Greg. 2017. 2017 Linux Kernel Development Report. https://www.linuxfoundation.org/2017-linux-kernel-report-landing-page/.

[46] Brian T Jones. 2018. free_email_provider_domains.txt. https://gist.github.com/tbrianjones/5992856/, last accessed 20 Aug 2019.

[47] Nicolas Jullien, Klaas-Jan Stol, and James Herbsleb. 2019. A Preliminary Theory for Open Source Ecosystem Micro-economics. In *Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability*. Springer, 49–68.

[48] Erik Kouters, Bogdan Vasilescu, Alexander Serebrenik, and Mark GJ van den Brand. 2012. Who's who in Gnome: Using LSA to merge software repository identities. In *28th IEEE International Conference on Software Maintenance*. 592–595.

[49] Nicholas Kuechler. 2015. Build Rackspace Cloud Servers with Ansible in a Virtualenv. https://nicholaskuechler.com/2015/01/09/build-rackspace-cloud-servers-ansible-virtualenv/.

[50] Jure Leskovec, Kevin J Lang, and Michael Mahoney. 2010. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*. ACM, 631–640.

[51] Bin Lin, Gregorio Robles, and Alexander Serebrenik. 2017. Developer turnover in global, industrial open source projects: Insights from applying survival analysis. In *IEEE 12th International Conference on Global Software Engineering*. 66–75.

[52] Johan Linåker, Patrick Rempel, Björn Regnell, and Patrick Mäder. 2016. Howfirms 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, 63–81.

[53] Sten R Ludvigsen, Andreas Lund, Ingvill Rasmussen, and Roger Säljö. 2010. *Learning across sites: New tools, infrastructures and practices*. Routledge.

[54] Björn Lundell, Jonas Gamalielsson, Stefan Tengblad, Bahram Hooshyar Yousefi, Thomas Fischer, Gert Johansson, Bengt Rodung, Anders Mattsson, Johan Oppmark, Tomas Gustavsson, et al. 2017. Addressing Lock-in, Interoperability, and Long-Term Maintenance Challenges Through Open Source: How Can Companies Strategically Use Open Source?. In *IFIP International Conference on Open Source Systems*. Springer, Cham, 80–88.

[55] Gregory Madey, Vincent Freeh, and Renee Tynan. 2002. The open source software development phenomenon: An analysis based on social network theory. *AMCIS 2002 Proceedings* (2002), 247.

[56] Juan Martinez-Romo, Gregorio Robles, Jesus M Gonzalez-Barahona, and Miguel Ortuño-Perez. 2008. Using social network analysis techniques to study collaboration between a FLOSS community and a company. In *IFIP International Conference on Open Source Systems*. Springer, 171–186.

[57] Audris Mockus. 2009. Organizational volatility and developer productivity. In *ICSE Workshop on Socio-Technical Congruence*.

[58] Audris Mockus. 2014. Engineering big data solutions. In *Proceedings of the on Future of Software Engineering*. ACM, 85–99.

[59] Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E* 69, 2 (2004), 026113.

[60] Onecloud. 2016. OneCloud Courses. http://1-cloud.net/onecloud-training-courses/.

[61] OpenStack. 2019. Mirrors of opendev.org. https://github.com/openstack.

[62] OpenStack. 2019. networking-cisco. https://github.com/openstack/networking-cisco.

[63] Tore Opsahl, Filip Agneessens, and John Skvoretz. 2010. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks* 32, 3 (2010), 245–251.

[64] Alma Orucevic-Alagic and Martin Höst. 2014. Network analysis of a large scale open source project. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE, 25–29.

[65] OVH. 2019. Overview of OVH. https://www.ovh.com/.

[66] Siobhán O'Mahony. 2005. 20 Nonprofit Foundations and Their Role in Community-Firm Software Collaboration. (2005).

[67] Ken Pepple. 2011. *Deploying openstack*. " O'Reilly Media, Inc.".

[68] Platform9. 2016. Platform9 Blog. https://platform9.com/blog/mors-lease-manager-openstack/.

[69] Platform9. 2017. Platform9 Website. https://platform9.com.

[70] Ray Reagans and Ezra W Zuckerman. 2001. Networks, diversity, and productivity: The social capital of corporate R&D teams. *Organization science* 12, 4 (2001), 502–517.

[71] Gregorio Robles and Jesus M Gonzalez-Barahona. 2005. Developer identification methods for integrated data from various sources. In *Proceedings of the 2005 international workshop on Mining software repositories*. ACM, 1–5.

[72] Gregorio Robles, Jesús M González-Barahona, Carlos Cervigón, Andrea Capiluppi, and Daniel Izquierdo-Cortázar. 2014. 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, 222–231.

[73] Cristina Rossi and Andrea Bonaccorsi. 2005. Why profit-oriented companies enter the OS field?: intrinsic vs. extrinsic incentives. In *ACM SIGSOFT Software Engineering Notes*, Vol. 30. ACM, 1–5.

[74] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering* 14, 2 (2009), 131.

[75] Caitlin Sadowski and Thomas Zimmermann. 2019. *Rethinking Productivity in Software Engineering*. Springer.

[76] Mario Schaarschmidt and Klaas-Jan Stol. 2018. Company soldiers and gone-natives: role conflict and career ambition among firm-employed open source developers. In *39th International Conference on Information Systems*. AIS, San Francisco, CA, USA.

[77] John Scott. 1988. Social network analysis. *Sociology* 22, 1 (1988), 109–127.

[78] Param Vir Singh. 2010. The small-world effect: The influence of macro-level properties of developer collaboration networks on open-source project success. *ACM Trans Softw Engineer Methodol* 20, 2 (2010), 6.

[79] Param Vir Singh, Yong Tan, and Nara Youn. 2011. A hidden Markov model of developer learning dynamics in open source software projects. *Information Systems Research* 22, 4 (2011), 790–807.

[80] Terje Snarby. 2013. *Collaboration Patterns among Commercial Firms in Community-Based OSS Projects*. Master's thesis. Institutt for datateknikk og informasjonsvitenskap.

[81] SwiftStack. 2019. Multi-cloud data storage and management for data-driven applications and workflows. https://www.swiftstack.com/.

[82] Xin Tan and Minghui Zhou. 2019. How to Communicate When Submitting Patches: An Empirical Study of the Linux Kernel. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article Article 108 (Nov. 2019), 26 pages. https://doi.org/10.1145/3359210

[83] Yong Tan, Vijay Mookerjee, and Param Singh. 2007. Social capital, structural holes and team composition: Collaborative networks of the open source software community. *ICIS 2007 Proceedings* (2007), 155.

[84] Jose Teixeira, Gregorio Robles, and Jesús M González-Barahona. 2015. Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem. *Journal of Internet Services and Applications* 6, 1 (2015), 14.

[85] José Apolinário Teixeira and Helena Karsten. 2019. Managing to release early, often and on time in the OpenStack software ecosystem. *Journal of Internet Services and Applications* 10, 1 (2019), 7.

[86] José Apolinário Teixeira, Salman Qayyum Mian, and Ulla Hytti. 2016. Cooperation among competitors in the open-source arena: The case of OpenStack. In *International Conference on Information Systems (ICIS)*. Association For Information System.

[87] Marat Valiev, Bogdan Vasilescu, and James Herbsleb. 2018. Ecosystem-level determinants of sustained activity in open-source projects: A case study of the PyPI ecosystem. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM, 644–655.

[88] P Wagstrom, JD Herbsleb, RE Kraut, and A Mockus. 2010. The impact of commercial organizations on volunteer participation in an online community. In *Academy of Management Annual Meeting*.

[89] Joe Weinman. 2017. The economics of the hybrid multicloud fog. *IEEE Cloud Computing* 4, 1 (2017), 16–21.

[90] Joel West and Scott Gallagher. 2006. Challenges of open innovation: the paradox offi rm investment in open-source software. *R&D Management* 36, 3 (2006), 319–331.

[91] Robert K Yin. 2017. *Case study research and applications: Design and methods*. Sage publications.

[92] Yuxia Zhang, Xin Tan, Minghui Zhou, and Zhi Jin. 2018. Companies' Domination in FLOSS Development – An Empirical Study of OpenStack. In *ACM/IEEE 40th International Conference on Software Engineering Companion, May 27-June 3, 2018, Gothenburg*. Gothenburg, Sweden, 440–441.

[93] Yuxia Zhang, Minghui Zhou, Audris Mockus, and Zhi Jin. 2019. Companies' Participation in OSS Development-An Empirical Study of OpenStack. *IEEE Transactions on Software Engineering* (2019).

[94] Yuxia Zhang, Minghui Zhou, Klaas-Jan Stol, Jianyu Wu, and Zhi Jin. 2019. Online appendix to "How Do Companies Collaborate in Open Source Ecosystems?". https://github.com/yuxia-zhang/ICSE2020-Company-Collaboration-in-OSS.

[95] Cangqi Zhou, Liang Feng, and Qianchuan Zhao. 2018. A novel community detection method in bipartite networks. *Physica A: Statistical Mechanics and its Applications* 492 (2018), 1679–1693.

[96] Minghui Zhou and Audris Mockus. 2010. Developerfluency: achieving true mastery in software projects. In *Eighteenth ACM Sigsoft International Symposium on Foundations of Software Engineering*. ACM, Santa Fe, New Mexico, USA, 137–146.

[97] Minghui Zhou and A Mockus. 2015. Who Will Stay in the FLOSS Community? Modeling Participant's Initial Behavior. *IEEE Transactions on Software Engineering* 41, 1 (2015), 82–99.

[98] Minghui Zhou, Audris Mockus, Xiujuan Ma, Lu Zhang, and Hong Mei. 2016. Inflow and retention in oss communities with commercial involvement: A case study of three hybrid projects. *ACM Trans Softw Engineer Methodol* 25, 2 (2016).