Determining the Geographical distribution of a Community by means of a Time-zone Analysis

Jesus M. Gonzalez-Barahona
Universidad Rey Juan Carlos
Madrid, Spain
jgb@gsyc.urjc.es

Gregorio Robles
Universidad Rey Juan Carlos
Madrid, Spain
grex@gsyc.urjc.es

Daniel Izquierdo-Cortazar
Bitergia
Madrid, Spain
dizquierdo@bitergia.com

ABSTRACT

Free/libre/open source software projects are usually developed by a geographically distributed community of developers and contributors. In contrast to traditional corporate environments, it is hard to obtain information about how the community is geographically distributed, mainly because participation is open to volunteers and in many cases it is just occasional. During the last years, specially with the increasing implication of institutions, non-profit organizations and companies, there is a growing interest in having information about the geographic location of developers. This is because projects want to be as global as possible, in order to attract new contributors, users and, of course, clients. In this paper we show a methodology to obtain the geographical distribution of a development community by analyzing the source code management system and the mailing lists they use.

CCS Concepts

•Human-centered computing → Collaborative and social computing;

Keywords

FLOSS; distributed development; time zones; open source software;

1. INTRODUCTION

Although free/libre/open source software (FLOSS) can be produced in many different ways, it is common practice nowadays that it is developed by a geographically distributed community. In this case, developers and other kinds of contributors share code, suggestions, comments, bug reports and discussions on the Internet, using specific-purpose communication and development tools. These communities have been subject of many research studies, some of them focusing on how they manage to work in a distributed manner [5, 15].

OpenSym '16, August 17 - 19, 2016, Berlin, Germany

ISBN 978-1-4503-4451-7/16/08…$15.00

DOI: http://dx.doi.org/10.1145/2957792.2957802

As communities have become more organized and institutionalized themselves, may it be through the creation of foundations or other legal entities [9], or with the growing interest in FLOSS by industrial companies, the information of where developers, contributors and users are has gained more importance. Some communities have shown interest to include such type of information in their software development dashboards[1].

An example of a community with a long lasting interest in learning about the geographic information of its developers is Debian. The Debian project maintains a map with the location of their developers[2]. It shows how Debian is an international project, primarily based in Western countries, with a certain balance between the number of North American and European developers. This is an interesting fact, for example, to choose the location of the DebConf, the annual Debian conference. It is also useful for Debian applicants to locate developers geographically close to them, since their admission process [11] requires face-to-face contact (for instance, to get the RSA key signed by an already Debian developer).

As another case, Mozilla's mission states that they want that "people worldwide can be informed contributors and creators of the Web"[3], so information about virtual participation is useful to design a global strategy. This type of information can be used as well by companies to assess the interest on their FLOSS products in certain geographic areas, and as information helpful in opening new markets.

But getting this information is not easy in most cases, since there are little data sources to collect it. The goal of this paper is to show a methodology to obtain the geographic distribution of a FLOSS community by means of analyzing artifacts produced as side-product of the software development effort. Therefore data will be extracted from source code management systems (such as Git) and mailing list archives, and don't require the active collaboration of developers.

The structure of the rest of this paper starts with the presentation of related research. In Section 3, we detail the proposed methodology. Then we explain how we apply it to analyze the CloudStack project, as an illustrative case study (Section 4). Finally, some conclusions are drawn and future directions are discussed.

[1]For example, dashboards for some communities offered by Bitergia, the company, can be found at http://bitergia.com/dashboards/.

[2]https://www.debian.org/devel/developers.loc

[3]https://www.mozilla.org/en-US/mission/

2. RELATED RESEARCH

The coordinated development of a software product by geographically distributed teams has been a matter of study since the late 1990s [2]. The specific case of FLOSS communities has been considered in several cases [5, 4, 14], even with some attention to regions where FLOSS development is rare, such as Africa [8].

Some of these efforts tried to estimate the location of the global FLOSS development community, such as the study of the geographical data obtained from the accounts of over one million registered users at SourceForge [10, 7]. In other cases, massive surveys were performed [6, 3]. Some of those asked for the country of origin and the current country of residence, in order to find developer migration patterns, finding how talent is attracted by the United States from all over the world [6].

The target for this paper is not to obtain an statistical estimate of the global FLOSS community, but to obtain a picture as accurate as possible of the community of a given project. Therefore, instead of performing massive surveys or analyzing software development platforms with many projects, we target project-specific tools. A similar approach can be found in Bird et al. who have examined how two large FLOSS projects perform their work in a distributed manner [1]. Related to the methods used in this paper, Tang et al. propose a set of techniques for identifying the country origin of mailing list participants [13], which they use to perform a case study on the impact of global participation on mailing lists communications in FLOSS projects [12].

3. METHODOLOGY

The analysis we propose is based on the traces left by developers in the Git repositories and mailing lists, and it can be easily extended to other data sources which log some geographical information.

In the specific case of Git, each commit includes as a part of its metadata some geographical information: a timezone. It is set according to the timezone obtained from the configuration of the computer where the commit took place, usually the one of the committer. And it is kept as the commit is merged in the main code base of the project. In the case of mailing lists, similar information is maintained in the message headers referring to dates. In particular, one of them keeps the timezone of the machine originating the message, usually that of the developer (or one configured with its timezone of residence).

To obtain and use this information, we follow the following four steps:

1. Identification of data sources. This process needs of the understanding of the infrastructure used by the community or project to analyze. In FLOSS communities, the required information is typically accessible publicly. Since we focus on the development community, the relevant repositories are source code Git repositories, and development mailing lists.

2. Extraction of information from data sources. The data process extraction is done with *CVSAnalY*[4] (for Git) and *MLStats*[5] (for mailing lists), two FLOSS data extraction tools that store the information they retrieve in a MySQL/MariaDB database. It is part of the *Metrics Grimoire toolset*[6], built and maintained by our team.

3. Analysis of the dataset. Given the amount of information provided by CVSAnalY and MLStats, and that not all of it is needed for this analysis, several filters can be applied before starting the analysis. In the case of Git, we ignore *merge* commits were ignored, and commits performed by bots: merges commits are in most cases performed automatically, or are not directly reflecting changes to the code, and bots do not directly correspond to the activity of a human. In the case of mailing lists, we ignore messages by bots, for the same reasons.

4. Timezone analysis. We use *GrimoireLib*[7], a library specialized in the analysis of information organized in SQL databases produced by *Metrics Grimoire tools*. This library provides a framework to deal with all of the resultant databases supporting the use of the output of *CVSAnalY* and *MLStats*. For our purposes, some new code was developed to perform the timezone analysis. It groups activities depending on the specified timezone, as a time difference from GMT, in integer hours (non-integer time zones, such as GMT+5:30, used in India, are rounded to their floor integer).

4. CASE STUDY: CLOUDSTACK

The CloudStack project[8] is a FLOSS project under the umbrella of the Apache Software Foundation. Its main goal is to provide easy deployment and management of large networks of virtual machines. This project provides infrastructure as a service (IaaS) highly available and scalable.

The first traces of information about CloudStack activity in their Git repositories start in August 2010. These repositories hold, at the end of 2015, close to 40,000 commits and 300 different developers with at least one commit of activity during these years.

The methodology presented in the previous section was applied to all these repositories.

We apply our methodology to two different sources from CloudStack: Git source code management (SCM) repositories and mailing lists archives (MLs). From SCM repositories we can obtain geographic information about how distributed the team of software developers is. Data from MLs provides information about the development community in general, including contributors to activities different than coding.

4.1 Analysis of SCM

We have grouped commits on a yearly basis, and showed the results graphically for number of distinct authors and number of commits for each period. The number of distinct authors is a good proxy of the number of developers working on the project, while the number of commits gives an idea of the overall development activity.

Figure 1 shows the results for the early phases of the project (year 2010). The horizontal axis references the detected time-zone relative to UTC. As it can be observed

[4]https://github.com/MetricsGrimoire/CVSAnalY
[5]https://github.com/MetricsGrimoire/MailingListStats
[6]https://github.com/MetricsGrimoire
[7]https://github.com/VizGrimoire/GrimoireLib/
[8]http://cloudstack.apache.org/

from the chart for authors, there is just one developer in UTC+0 (the timezone for UK, Ireland and Portugal), and three in UTC+5 (India). Almost all developers are located in timezones corresponding to the U.S. West Coast (UTC-7 and UTC-8, for Winter and Summer time), with some very likely in the U.S. East Coast (UTC-5 and UTC-4), although these timezones are shared with some countries in South America.

The chart for commits is even more revealing, which most of the activity clearly focused on the U.S. West Coast: so much, that the contributions from other regions are almost negligible.

In summary, CloudStack was in 2010 a project that had developers from several regions, but the main development activity was clearly concentrated in the U.S. West Coast.

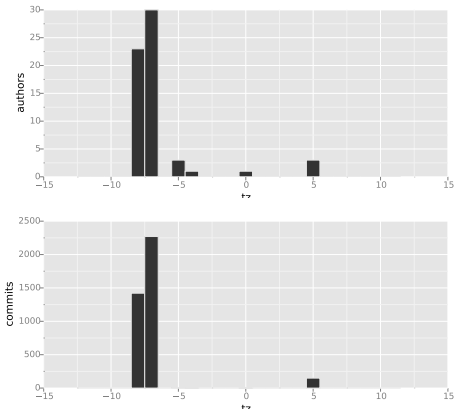

Figure 1: Time-zone analysis for authors (top) and commits (bottom) for the CloudStack Git repositories (2010).

Figure 2 shows the same analysis for CloudStack Git repositories during 2014. As we can see from the authors chart, their number increased significantly in those four years. In addition, the project was more geographically distributed. Now we can observe authors from all America time zones, although the West Coast is still predominant. The three European timezones (UTC+0 to UTC+2) have all over 20 developers, and India (UTC+5) has the maximum number of authors, 55. Developers from other Asian areas, or from Australia, are marginally present too (UTC+7 to UTC+10).

If we observe the commits chart in the same Figure 2, we see that CloudStack development activity is performed in three regions: the U.S. (with a peak in the West), Europe (with a peak in Eastern Europe, UTC+2) and India (UTC+5).

4.2 Analysis of MLs

Figure 3 shows the results for the analysis of the CloudStack MLs for the year 2012. The vertical axis in the authors chart represents provides the number of different authors, while for the messages chart it corresponds to number of messages. MLs in 2012 already showed a global distribution of participants of the CloudStack project, hinting that authorship and activity in MLs possibly precedes development activity. However, there are two interesting aspects that require a specific analysis by themselves: the cases of UTC+0 and UTC+8. UTC+0 includes authors and activity

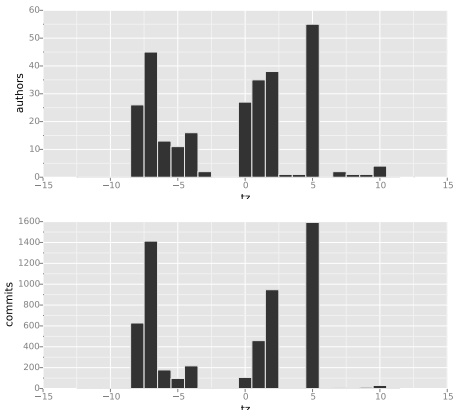

Figure 2: Time-zone analysis for authors (top) and commits (bottom) for the CloudStack Git repositories (2014).

from countries such as UK, Ireland or Portugal, but as well all those who configure their timezone as "GMT" in their mail clients, such as for example many frequent travelers do. Therefore, the data for UTC+0 has to be considered with some precaution. In the case of UTC+8, which corresponds to China, Southeast Asia and Australia's Western Standard Time, among other territories, it is strange how there is a large number of authors, but few messages have been sent. This very corresponds to a much lower participation per person than in other regions.

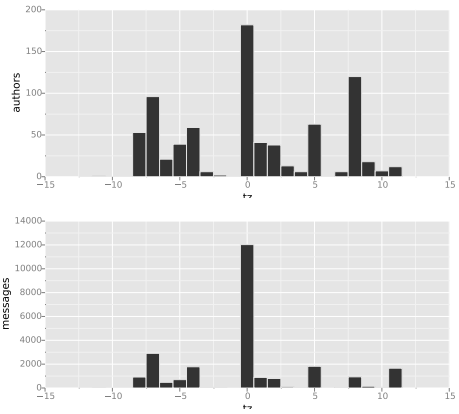

Figure 3: Time-zone analysis for authors (top) and messages (bottom) for the CloudStack development mailing list archives (2012).

Figure 4 shows the same analysis for MLs during 2014. Again, UTC+0 is the most significant timezone, which heavily biases results, thus limiting the possibility of performing a proper analysis.

5. DISCUSSION AND CONCLUSIONS

We have shown how the proposed methodology allows for the determination of the main geographical areas with activity related to development. In the case of CloudStack we could find out how it started in a certain region, and later expanded to be globally distributed.

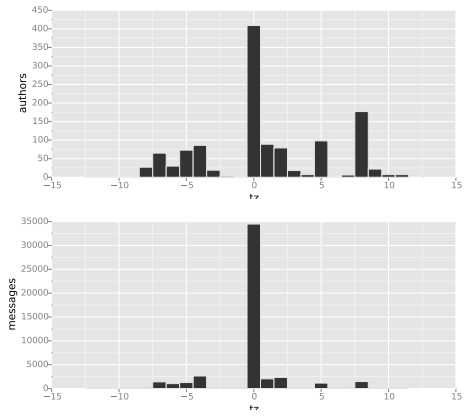

Figure 4: Time-zone analysis for authors (top) and messages (bottom) for the CloudStack development mailing list archives (2014).

The methodology can be fully automated, and it uses as the basis for its analysis data that is in general available or can easily be obtained.

However, there are some limitations:

1. Some regions have a different time zone during Winter and Summer. This can mangle the results, at least visually.
2. The nature of timezones, comprising several regions, makes it impossible to know to what specific zone belongs a developer. For example the Central European time zone includes from Spain to Poland in Europe, but many African countries as well.
3. The UTC+0 configuration problem in email clients may cause a severe *bias* for this time zone, as it has been shown in the analysis of the MLs in CloudStack.
4. Sometimes, obtaining the data for the mailing list analysis is not straightforward as the archives only store the server date. To perform the analysis correctly, the time and time zone data in the original messages has to be preserved, and it not always is.

In summary, we have presented a methodology that can be used to measure the geographic distributed of a FLOSS project. It is based on data that is in general publicly available, as a side-product of software development processes. With the help of a case study, we have explored how we can infer information about the geographic dispersion of contributors to the project, and how it changes over time.

As future work we plan to apply this methodology to other projects to find out if different patterns can be found.

Acknowledgments

The work of Jesus Gonzalez-Barahona and Gregorio Robles has been funded in part by the Spanish Gov. under SobreVision (TIN2014-59400-R) and by Comunidad de Madrid under eMadrid (S2013/ICE-2715). Daniel Izquierdo-Cortazar is supported by the Spanish Gov., Torres Quevedo grant (PTQ-12-05577). All three authors are supported in part by the European Comission, under Seneca, H2020 Program (H2020-MSCA-ITN-2014-642954).

6. REFERENCES

[1] Christian Bird and Nachiappan Nagappan. Who? where? what? examining distributed development in two large open source projects. In *Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on*, pages 237–246. IEEE, 2012.

[2] Erran Carmel. *Global software teams: collaborating across borders and time zones*. Prentice Hall PTR, 1999.

[3] Paul A David and Joseph S Shapiro. Community-based production of open-source software: What do we know about the developers who participate? *Information Economics and Policy*, 20(4):364–398, 2008.

[4] Sebastian Von Engelhardt and Andreas Freytag. Geographic allocation of oss contributions: the role of institutions and culture. *Jena Economic Research Papers*, 51, 2009.

[5] Daniel German. The GNOME project: a case study of open source, global software development. *Software Process: Improvement and Practice*, 8(4):201–215, 2003.

[6] Rishab A Ghosh, Ruediger Glott, Bernhard Krieger, and Gregorio Robles. Free/libre and open source software: Survey and study, 2002.

[7] Jesus Gonzalez-Barahona, Gregorio Robles, Roberto Andradas-Izquierdo, and Rishab Ghosh. Geographic origin of libre software developers. *Information Economics and Policy*, 20(4):356–363, 2008.

[8] Hadja Ouattara, Jonathan Ouoba, and Tegawendé F Bissyandé. Open source in africa: An opportunity wasted? In *e-Infrastructure and e-Services for Developing Countries*, pages 184–188. Springer, 2013.

[9] Dirk Riehle. The economic case for open source foundations. *Computer*, 43(1):0086–90, 2010.

[10] Gregorio Robles and Jesus M Gonzalez-Barahona. Geographic location of developers at sourceforge. In *3rd International workshop on Mining software repositories*, pages 144–150. ACM, 2006.

[11] Gregorio Robles, Jesus M Gonzalez-Barahona, and Martin Michlmayr. Evolution of volunteer participation in libre software projects: evidence from debian. In *1st International Conference on Open Source Systems*, pages 100–107, 2005.

[12] Ran Tang, Ahmed E Hassan, and Ying Zou. A case study on the impact of global participation on mailing lists communications of open source projects. *Proc. KCSD 2009*, pages 63–76, 2009.

[13] Ran Tang, Ahmed E Hassan, and Ying Zou. Techniques for identifying the country origin of mailing list participants. In *Reverse Engineering, 2009. WCRE'09. 16th Working Conference on*, pages 36–40. IEEE, 2009.

[14] Sebastian von Engelhardt, Andreas Freytag, and Christoph Schulz. On the geographic allocation of open source software activities. Technical report, Jena economic research papers, 2010.

[15] Liguo Yu, Zhong Guan, and Srini Ramaswamy. The effect of time zone difference on asynchronous communications in global software development. *International Journal of Computer Applications in Technology*, 53(3):213–225, 2016.