A Quantitative Profile of a Community of Open Source Linux Developers

Bert J Dempsey, Debra Weiss, Paul Jones, and Jane Greenberg
UNC Open Source Research Team[1]

School of Information and Library Science
University of North Carolina at Chapel Hill
Chapel Hill, North Carolina 27599-3360
osrt@metalab.unc.edu
October 6, 1999

Abstract

Open source software, or free software, has generated much interest and debate in the wake of a number of high-impact applications and systems produced under open source models for development and distribution. Despite the high degree of interest, little hard data exists to-date on the membership of collaborative open source communities and the evolutionary process of their repositories. This paper contributes a baseline quantitative study of one of the oldest continuous repositories for the Linux open source project (the UNC MetaLab Linux Archives), including demographic information on its broad community of developers. Our methodology is a close examination of collection statistics, including custom monitoring scripts on the server, as well as an analysis of the contents of user-generated metadata embedded within the Archives. User-generated metadata files in a format known as the Linux Software Map (LSM) are required when submitting open source software for inclusion in non-mirrored portions of the MetaLab Linux Archives. The over 4500 LSMs in the Archives then provide a demographic profile of contributors of LSM-accompanied software as well as other information on this broad subset of the Linux community. To explore repository evolution directly, an instrumented Linux Archives mirror was developed, and aggregate statistics on content changes seen over a month-long period are reported. In sum, our results quantify aspects of the global Linux development effort in dimensions that have not been documented before now, as well as providing a guide for more detailed future studies.

Introduction

Open source development communities have successfully created, distributed, and continued to evolve many important software projects---the GNU project's utilities and libraries including the gcc compiler and Emacs editor, the Perl and Tcl languages, the Apache WWW server, and the Linux and FreeBSD operating system. Open source, or free software, means more than access to source code (see Appendix A), and there is not universal agreement on a single open-source development model. Nonetheless, the guiding principle for open source software is that, by sharing source code, developers

[1] See **http://metalab.unc.edu/osrt/** for related work by the Open Source Research Team.

cooperate under a model of rigorous peer-review and take advantage of "parallel debugging" that leads to innovation and rapid advancement in developing and evolving software products. Open-source licensing, moreover, ensures an open market in integration and support for these products downstream.

Software production and distribution driven by the open source model thus has strong practical advantages as well as its strong appeal to those who, in Richard Stallman's words, see open source software in a "social advantage, allowing users to cooperate, and an ethical advantage, respecting the user's freedom. [3]" Advocates emphasizing the business reasons for adopting an open source model have engendered in recent years an on-going---and often acrimonious--- debate over the ultimate impact of open source communities. Some have proposed that free software methods leveraging the Internet represent an alternative economic model for engendering and managing robust software that will dramatically reshape the multi-billion dollar commercial software industry. Skeptics meanwhile continue to challenge the idea that the technical and organizational approach represented by open-source development can really scale up in the coming years and produce the robust software required for large-scale mainstream computing [1]. The stakes in this debate are clearly quite high.

A prime difficulty in understanding and drawing conclusions about open source collaborative development has been the sketchy information available on exactly who participates in open source development and how their software archives evolve. This lack of information is understandable given the distributed, organic process of collaborative development in open source communities.

The contribution of this paper is a baseline quantitative study of a broad community of developers within the Linux open source effort, which, due to its influence and increasing user base, is widely regarded as a cornerstone project for large-scale open-source development. Our work characterizes a very large repository of Linux-related materials and analyzes information embedded within the collection on the nature of its contributors. Derived from a variety of collection meta-data statistics, the data and analysis here supports the assertions that Linux community is indeed very vibrant, geographically diverse, and engaged in a broadening the quantity and scope of the freely available Linux software and documentation.

Background on Open Source Development

The genesis of the open-source model for software development and distribution goes back to the earliest days of software in university environments. Open-source software is an alternative term for "free software", which was popularized by the seminal Free Software Foundation, founded in 1984 by MIT researcher Richard Stallman. The Free Software Foundation is the parent organization for the GNU (GNU's Not Unix) project. Stallman's vision was to develop a free operating system, complete with standard software tools such as compilers, interpreters, text editors, mailers, and so forth, in order to recreate a community of cooperating hackers that he felt had been lost [3]. Under his direction, the Free Software Foundation popularized the term "free software" as explained in the now-classic distinction, free as in "free speech", not "free beer". That is,

free software may or may not be distributed with a monetary cost, but the knowledge that underlies the program, i.e., the source code, should be freely available in order to empower future innovation. Software source code is a form of scientific knowledge, and just as scientists publish so that other scientists can build on their results, computer scientists must publish their source code in order to foster continued innovation in computing.

Unfortunately, the term "free software" has negative connotations for many in the commercial computing world, and the tone adopted by Stallman, the most prominent free software advocate for some time, was distinctly anti-business. In early 1997, a group of leaders in the free software community decided to address this problem head-on with a marketing campaign designed "argue for 'free software' on pragmatic grounds of reliability, cost, and strategic business risk." [4]. They were goaded to action largely by frustration over what they felt was the unrecognized potential of free software as a driver of innovation and the basis for the development of commercial-grade software, despite the successes of Apache, Linux, and other projects. An initial decision of the group, which would become the Open Source Initiative, was to choose the term "open source" for their campaign to avoid the baggage being carried by the term "free software".

A key component of Stallman's effort in developing a successful free software organization was to formulate a licensing agreement that would prevent businesses from taking free software and using it in binary-only redistributions for commercial gain. Stallman developed the GNU General Public License, known as the GPL or "copyleft", to address this issue. In subsequent years, other open-source efforts adopted variations on copyright statements designed to enable open-source works to thrive while not hampering the ability of developers to incorporate open-source work effectively[2]. For its part, The Open Source Initiative adopted a set of criteria, titled "The Open Source Definition", for open-source licensing. Based on an earlier document by Bruce Perens, the Open Source Definition explicitly mentions some example licenses that fit its criteria, including that of the GNU project (the GNU GPL), the Berkeley Unix Project (BSD), the X Consortium, and a few others. For reference, the Open Source Definition, Version 1.7, is reproduced in Appendix A.

Linux: Open-Source Development on a Global Scale

Internet connectivity has enabled the open-source notion of cooperative, peer-reviewed software development to be deployed on a global scale. Perhaps the most influential open-source project to-date has been and continues to be the Linux operating system. Linux began as a personal project of a graduate student in Finland, Linus Torvalds, in 1991. The Linux project now represents a mature operating system that runs on the popular hardware platforms. Linux is playing an increasingly significant role in the business plans of established computing companies, in university research labs, and in the development of a new set of companies focused on Linux support and integration issues.

[2] It is interesting to note that the GNU Emacs FAQ (http://www.gnu.org/software/emacs/emacs-faq.text) dated February 1999 points out: "The real legal meaning of the GNU General Public License (copyleft) will only be known if and when a judge rules on its validity and scope. There has never been a copyright infringement case involving the GPL to set any precedents…"

According to April 1999 statistics at the Internet Operating System Counter site [5], Linux is now the operating system at over 30% of Internet server sites. Linux has been estimated to have 10 percent of the server market share in the Unix market with growth trends suggesting it will dominate the Unix arena in a few years.

The Linux Kernel Project continues to be led by Linus Torvalds himself, with a significant array of co-developers throughout the world. In addition, the Linux community of application-level developers and documenters has grown in proportion with the rising tide of users and installed systems. In a recent interview, Linus has said that in the near future "the most exciting developments for Linux will happen in user space, not kernel space [6]." Thus, increasing focus and energy are now being directed towards creating applications and utilities that will spread the use and usefulness of the Linux kernel work.

The focus of this paper is with this latter group, which we call *Application Contributors* to distinguish them from the contributors within the Kernel Project itself. *A priori*, the overlap between these Application Contributors and developers in the Kernel Project is unknown. Linus and other prominent Linux developers do show up in the set of Application Contributors. As a group, the Application Contributors would be expected to represent a broader range of contributors since their contributions are non-juried and include those who make small contributions of specific applications or utilities. (Application contributions can also be very large and complex programs, e.g., a Linux port of the Sendmail server.)

To explore the nature of contributions by Application Contributors and their collective profile, the paper presents a discussion of collection and server statistics gleaned from one of the oldest and most comprehensive Linux repository site, metalab.unc.edu, run by UNC MetaLab (formerly the original SunSITE). With components of the repository at MetaLab mirrored from other key Linux sites (e.g., www.redhat.com and kernel.org), the MetaLab collection includes virtually all Linux-related materials available on the WWW, including all major distributions of the base kernel code, the Linux Documentation Project (coordinated and hosted by MetaLab), and a large archive of contributed software and auxiliary materials. Our study specifically focuses on this latter portion of the MetaLab archives, and its contributors are, by definition, the community of Application Contributors that we profile.

Towards this effort, we analyze the approximately 4500 user-generated metadata files (Linux Software Maps (LSMs)) embedded within the collection materials. The LSMs offer self-reported information on the demographics of Linux developers involved mostly in application-level tools and utilities (as opposed to the largely separate Linux Kernel Project) as recorded over a 5-year period. Also, we collected data on repository changes across a large portion of the MetaLab archives by instrumenting a month-long monitoring experiment on the MetaLab server. This experiment yielded data on the global pattern of change in repository contents.

Below we briefly review the history of the Linux repository at MetaLab and the role of LSMs in the archive. The sections following then present our quantitative profile of the collective set of LSM-based contributors and patterns of content change.

History of Linux Repository at UNC MetaLab

Not long after Linus Torvalds released his first copy of the Linux kernel to the Internet in 1991, an American mirror of his FTP site was established at banjo.concert.net in the Raleigh-Durham area of North Carolina. However the Linux project was growing quickly and soon banjo was short of disk space. In 1992, Jonathan Magid, who had just become the student systems administrator of a new project called SunSITE at the University of North Carolina, agreed to take on not only the mirror but to collect contributions of Linux-related software. Since then the site, now MetaLab.unc.edu, has been a major resource for the Linux community—both developers and users.

The over 125,000 files in the Linux repository at UNC MetaLab is available through FTP and HTTP under the /pub/Linux portion of the server. We will refer to these materials henceforth in this paper as the MetaLab (ML) Linux Archives or the MetaLab (ML) Linux repository. As noted above, our LSM-based analysis focuses on the subset of the ML Linux Archives in which LSMs play a prominent role, namely the /pub/Linux tree excluding the *docs* and *distributions* subdirectories. Along with other collections, /pub/Linux is served from high-performance machines connected with excellent high-speed Internet connectivity. Not infrequently, access counts to the ML Linux Archives via HTTP and FTP exceed 100,000 transactions per 24-hour period.

Contributions to the ML Linux Archives are required to be accompanied by a small metadata file in a format called the Linux Software Map (LSM). This convention arose naturally from the Linux community needs, as detailed in the history below, and as such is widely adhered to by contributors. Since our analysis relies in part on summary information taken from the set of all LSMs, we present a short overview that clarifies the origins and role of LSMs.

History of Linux Software Maps

Although the Linux Archives in 1992 was minuscule compared to its size today, the speed of access for many users and contributors was so slow that just downloading random but interesting files to see if they contained useful software or not. Jeff Kopmanis of Michigan decided that a small descriptive metadata file to be called a Linux Software Map or LSM should be associated with each entry[3]. After surfing several gopher sites and posting to various Linux newsgroups and an e-mail exchange with Jonathan Magid, Kopmanis took a close look at a working proposal to the IETF for metadata called Internet Anonymous FTP Archives or IAFA. Kopmanis was working from the 1992 version of IAFA, which he modified to better suit the specific needs of the Linux community. (Interestingly enough the IAFA metadata description has yet to become an RFC and is not widely used).

[3] Personal correspondance, Jeff Kopmanis, August 1999.

Kopmanis slightly revised the LSM description (in a version noted by a beginning tag of Begin2) after receiving feedback from other users and contributors. Before changing jobs in August 1994, Kopmanis posted notes to Linux newsgroups asking for a new LSM keeper. Lars Wirzenius of Finland was selected as the new LSM keeper. He instituted several changes to the LSM including:

- no limits on line lengths
- less awkward multi-line format
- nicer method (for the user) for specifying the FTP site and files
- one entry for all files comprising a package

This new LSM (Noted by *Begin3* as the first tag) went into effect on August 4, 1994. The template is included in Appendix B.

In October 1996, Wirzenius passed on the job of keeping the LSMs to Aaron Schrab of Indiana who has the job of LSM keeper presently. People with specific interest in LSMs are subscribed to the LSM-workers mailing list at execpc.com. This list is maintained by Schrab.

Aaron Schrab notes in the LSM.README[4]:

> All entries have been entered by volunteers all over the world via email using the template below [in the file named LSM.README]. New versions [of the LSM] will appear first on sunsite.unc.edu and will be announced in the newsgroup *comp.os.linux.announce*. Discussions pertaining to the LSM will be held in the newsgroup, *comp.os.linux.misc*.

Purpose of LSMs within MetaLab Linux Archives

From their beginning, Linux Software Map entries were designed to help developers make their contributions highly available to users and to other developers by serving as finding aids as well as a standardized means of announcing new software (to *comp.os.linux.announce* and other newsgroups). LSMs also insure that authors are properly credited if and when their software is integrated into Linux distributions.

LSMs are created according to the LSM metadata template consisting of 14 metadata elements, five of which are mandatory. The five mandatory fields are: Title, Version, Entered-date, Description, and Primary-site fields. Information on creating LSMs is keep at http://metalab.unc.edu/pub/Linux/docs/linux-software-map/lsm-template. The LSM metadata schema is based on the IAFA (Internet Anonymous FTP Archives) metadata schema that was developed for Archie **[7]**. As noted above, the LSM metadata schema has gone through a series of revisions that were initiated and overseen mainly by Jeff Kopmanis, with input from members of the Linux community, and it is now in its third revision.

LSM generation permits the authors to record their expert knowledge about the resource that has been created, rather than a second hand representation, which is practiced with

[4] http://MetaLab.unc.edu/pub/linux/docs/LSM/LSM.README

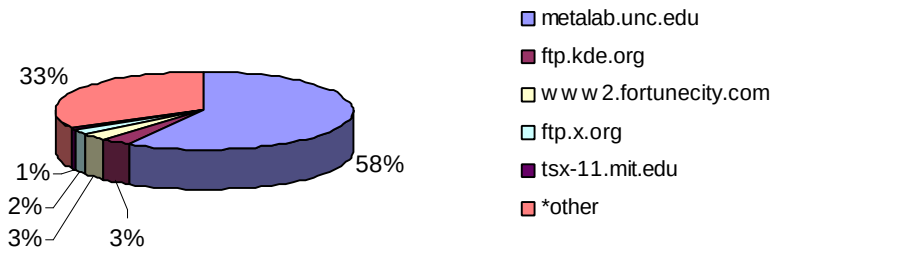
many other metadata schemas in the networked environment. When a contributor submits software to the MetaLab Linux Archives, he places the software and an associated LSM into ftp://metalab.unc.edu/incoming/Linux/ This area is inspected daily by the Archives maintainers using a program called *keeper* which was written by Eric Raymond then modified by Miles Efron. The Linux Archivist using *keeper* reviews the LSM information and places the software and LSM in their correct home in the Archive. LSMs help the archivist replace older obsoleted versions of software by use of standard names and version numbers. The LSM is then forwarded to the LSM maintainer for inclusion in the definitive LSM list at *execpc.com*. The LSM list at execpc.com is regularly mirrored by major Linux sites worldwide including MetaLab[5].

Many, but not all by any means, contributors of Linux software use LSMs as their means of describing their software as they send announcements to comp.os.linux.announce and other newsgroups. Several LSM searches assist users and developers in finding software including:

- http://www.linux.org/apps/lsm.html (last updated 13-Nov-1998)
- http://www.boutell.com/lsm/ (keyword and title searching only)
- http://metalab.unc.edu/linsearch/ (complete LSM template-based searching with support for complex searches)

Profiles of the ML Linux Archives and LSM-Based Contributors

In this section we present data sets and analysis designed to provide selected quantitative characterizations of the open source contribution process as reflected in the ML Linux Archives. The data presented is derived from server-side programs that (1) summarize



*Includes 825 other distinct primary sites

Figure 1: Primary-Site Field in the LSMs (3835 LSMs)

filesystem statistics of the repository, (2) analyze the contents of the over 4,500 LSM files found in the ML Archives, and (3) monitored and recorded content changes over a month-long period. For the data reported on LSM content, the number of LSM files

[5] The numbers of active LSMs in the MetaLab archive actually outnumber those in the official LSM listing maintained by Aaron Schrab by about 500 entries in September 1999.

varies across statistics since some LSMs contain missing or unusable (e.g., a date field as "Thursday") metadata information in some of their fields.

Role of the ML Archives in Linux Community

Since MetaLab mirrors most other popular Linux archives including most distributions, MetaLab constitutes a superset of most archives and includes most, if not all, current LSM documented software. Evidence of the central role of the ML Linux Archives can be found by examining the Primary-Site field in LSMs. This field allows authors to specify the primary Internet server (and path on that server) on which their Linux contribution resides. As seen in Figure 1, 58% of all LSMs list *metalab.unc.edu* (or, the

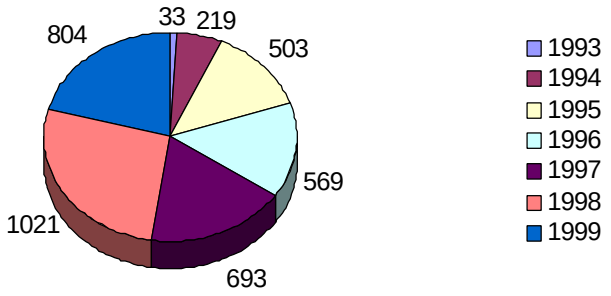


Figure 2: LSMs by Date-Entered Field

(3842 LSMs, through June 1999)

older domain name, *sunsite.unc.edu*) as the primary repository site, and the next most popular sites appear in 3% or less of the LSMs. This statistic may be biased by the fact that the ML Archives require an LSM with each submission whereas other repositories may not.

Characteristics of the Content in the ML Linux Archives

Table 1 provides summary statistics on the six top-level subdirectories in the ML Linux Archives with the most LSM metadata files. This summary was taken on September 21, 1999. At this time, the ML Linux Archives had a total file count of 129,109 files with 4633 LSMs. However, most of these files (94, 401) are located in the *distributions* subdirectory of the archive where only 104 of the LSM files are found. LSM contributions may find their way into individual distributions of Linux, although this study does not investigate the extent to which this occurs. The distributions generally are using the Redhat Package Manager (over one-third of files under the *distributions* subdirectory have the extension *.rpm*) or other archiving tools to encapsulate related files and their metadata. Another 14,075 files are under the *docs* subdirectory, where only 23 LSMs were located since LSMs are oriented towards metadata for software contributions.

Top-level directory in /pub/Linux on MetaLab	Number of LSMs	Total Number of Files and Bytes	Number of .tar.gz or .tgz Files
apps	1312	3904 files (994 MB)	1382
system	1301	3865 (391MB)	1555
X11	397	2495 (567MB)	1039
utils	373	1670 (218 MB)	579
games	297	896 (130 MB)	356
devel	288	1433 (1.3 GB)	487

Table 1: Six Subdirectories in ML Linux Archives Containing the Most LSMs

Excluding the *distributions* and *docs* subdirectories, there are 4455 LSMs covering the 20,633 files in the remaining portion of the ML Linux Archives. Table 1 shows the top six subdirectories in terms of number of LSMs. It also presents the total number of files ending in *.tar.gz* or *.tgz* to indicate the high density of compressed file archives of this form, the most common format for submission of software source code.

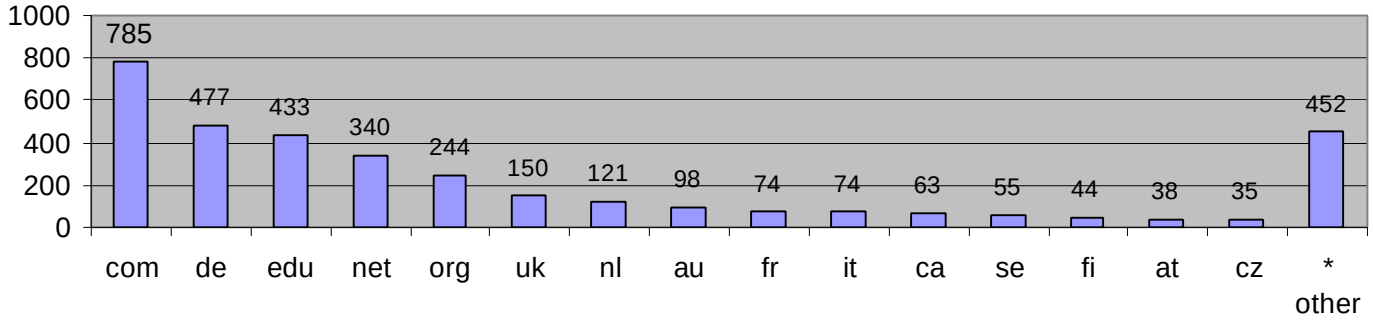


Figure 3: LSMs by Email Suffix from Author Field (3483 LSMs)

*Category of "other" includes 60 other distinct author email endings

Profile of LSM-Accompanied Contributions

As noted earlier, the ML Archives require the Linux Software Map metadata file to accompany contributions sent to the ML Linux Archives. Figure 2 breaks down the LSMs in the ML Linux Archives by year for the LSM files that included full date information. In interpreting this graph, it is important to remember that the policy of the MetaLab archive has been to replace old LSMs when a new version of a software package arrives. Thus, this data is not an accurate longitudinal study of how many contributions have been made in which years. Rather, it shows that portions of the existing archive extend back to 1993, but many of the contributions have been added or updated in the recent past, e.g., almost one-fourth of the LSMs are additions or changes during the first 6 months of the 1999 year.

The LSM files also contain a field for including an author's email information. Figures 3 and 4 give a breakdown of this information by email suffix. The data shows that Linux Application Contributors indeed come from both the commercial and the educational

world. Moreover, while *com*, *edu*, and *net* are difficult to accurately associate with geographical information, the demographics of contributors reveals a strikingly strong

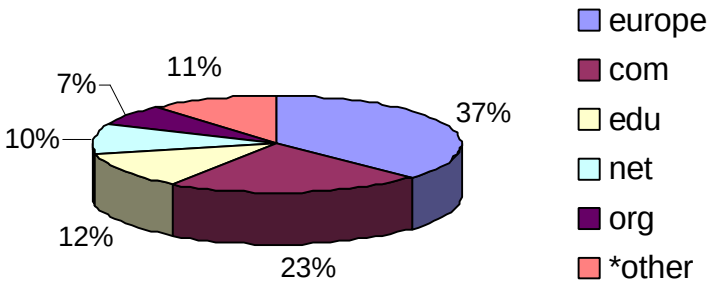


Figure 4: LSM Contributors by Email Suffix from Author Field (3483 files)

*The "other" category includes 38 other distinct author email endings.

European influence within the Linux community. To clarify, Figure 4 presents the same data but with all suffixes representing European countries aggregated. Note that Figure 4 underrepresents European participation in Linux development since some authors with

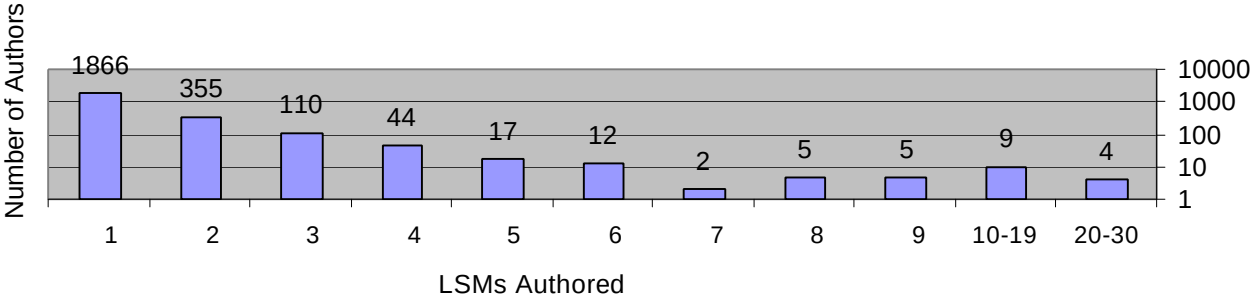


Figure 5: Number of LSM Contributions for Each Contributor (4184 LSMs)

.*com* email suffixes are presumably located in Europe.

A separate question of interest is how often Linux developers contribute to open-source software. We extracted a frequency count of authors' last names from the LSM files that contain author data and present this information in Figure 5. As seen there, the vast majority of LSM authors have contributed only one or two items, with only a very small number of developers having produced five or more contributions. Only thirteen contributors have ten or more contributions to their credit. This data then speaks to the breadth of the Linux developer community: as seen in the data for Application Contributors, the open-source development effort has not been dominated by a few very prolific developers, but rather, over time, many participants adding isolated contributions.

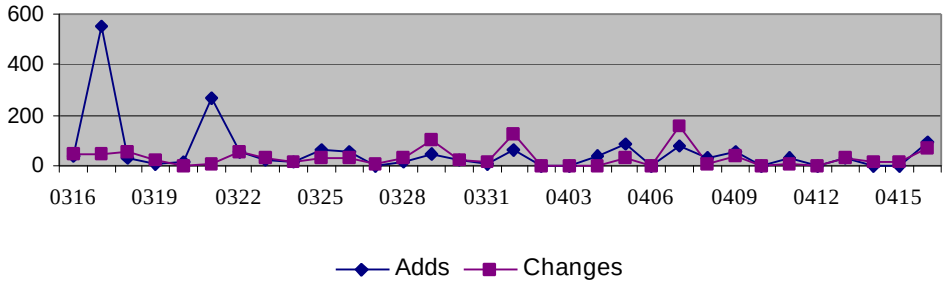


Figure 6: Daily Contributions Summary (3/16/99 to 4/16/99)

Studying the ML Linux Repository Dynamics

As suggested in the data of Figure 2, the ML Linux Archives is an active repository where contributors continue to submit new and updated materials on a daily basis. To provide a baseline profile of the repository contribution process, specifically that driven by the Application Contributors, a repository monitoring experiment was undertaken.

The experiment took the form of a local mirror that gathered change information from a portion of the ML Linux Archives on a twice-daily basis. Due to resource limitations, we restricted our observations to an 8-gigabyte subset of the ML Linux Archives, specifically the six subdirectories under /pub/Linux with the most LSM files (see Table 1) as well as the *docs* subdirectory. These directories then represent important areas where Application Contributors send LSM-accompanied software and documentation. The mirror script for the experiment ran twice a day, at 8:30 a.m. and 8:30 p.m., to synchronize with the source directories on *metalab.unc.edu***.** Full details of the mirroring experiment are described in [8].

Figure 6 shows the rate of activity seen in our month-long monitoring experiment. As seen here, the Archives receive constant traffic from open source contributors updating existing materials and adding new items. At times, the process is very bursty, most probably due to external events such as a major release of new software as well as occasional update backlogs when MetaLab staff are off-line over holidays.

More significantly, Figure 7 shows that about one-third of all activity observed in this month-long period resulted from changes to existing files in the repository, e.g., new releases of existing software utilities, whereas about two-thirds represents the addition of new files to the repository. Figure 8 breaks out this activity by some common file types. We note that about 1.5% (59) of the LSMs in the Archives were updated in this month while over 4% (179) of the total were added in this month. In line with these changes then, many of the changes are seen to involve compressed archives using the .gz compression; in fact, over one-half of activity represents new or updated .gz files as software packages are added and changed. (Note that the standard practice at MetaLab

has been to remove old files when new versions of open source software or documentation is submitted. The data in the figures does not track file deletions as a separate operation in the update process.)

While our data is not conclusive, the data in Figures 7 and 8 corroborate that in Figure 2 to suggest that the ML Linux Archives is vibrant and contains little stale or outdated materials as a percentage of the entire collection. An interesting follow-on study would be to connect our baseline numbers directly with the question of how often do Application Contributors update (or, alternatively, abandon) their contributions after

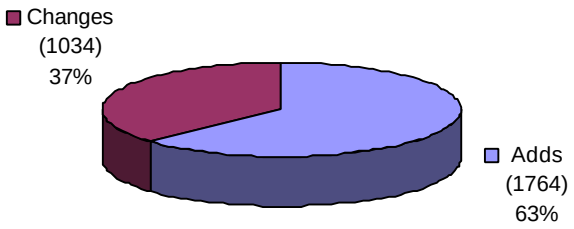


Figure 7: New Contributions (adds) and Updated Files (changes) (3/16/99 to 4/16/99)

creating them? Definitive information on this aspect of open source contribution would speak to the robustness of the open-source process over time and the extent to which the community is able to carry forward the work of other contributors from the past.

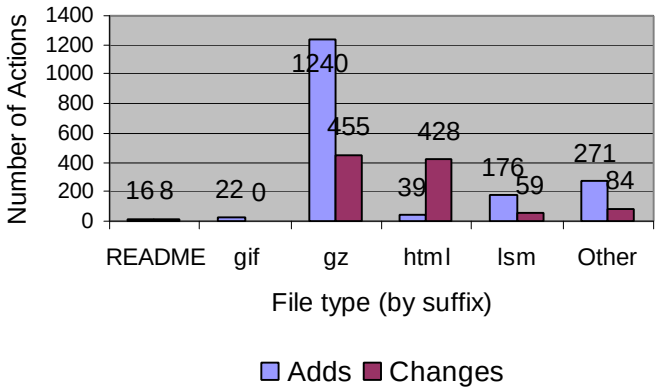


Figure 8: Changes by File Type (3/16/99 to 4/16/99)

Conclusion

This article reports on quantitative summaries of the large repository of Linux-related materials found in the MetaLab Linux Archives, one of the oldest continuous repositories

for Linux materials. User-generated metadata files in a format known as the Linux Software Map (LSM) are required when submitting open source software for inclusion in the MetaLab Linux Archives. Our study examined information inside the over 4500 LSMs, along with longitudinal collection statistics from the MetaLab filesystem, in order to extract a quantitative profile of the large group of Linux developers represented there, a group we have called Application Contributors in the Linux community.

Our results confirm a very broad participation at this level in the open source software associated with the Linux Archives. A strong bias towards European developers is revealed, reflecting the European roots of Linux perhaps, as well as a balance between *.com* and *.edu* contributors. Date information in LSMs and our month-long monitoring experiment reveal a very active repository where users submit many changes to existing files as well as adding new files. This phenomenon along with other questions raised by our baseline study warrant further investigation to develop a fuller understanding of the dynamics of large-scale cooperation over time, as exhibited in the Linux open-source development effort. Such work will shed light on what aspects of the Linux effort can be taken to be representative in determining the viability and robustness of large-scale open-source development projects in the future.

References

[1] **T. Lewis, "Asbestos Pajamas: An Open Source Dialogue," *IEEE Computer*, vol. 32, pp. 112ff, 1999.**

[2] **CACM, Special Section on *Open-Source Software Development*, Tim O'Reilly, Ed., Communications of the ACM, April 1999.**

[3] **R. Stallman, "The GNU Operating System and the Free Software Movement," in *OpenSources: Voices from the Open Source Revolution*, C. DiBona, S. Ockman, and M. Stone, Eds.: O'Reilly Books, 1998, pp. 53-70.**

[4] **OpenSource.org, "Open Source Initiative Launch Announcement," http://opensource.org/press_releases/osi-launch.html, 1998.**

[5] **H. U. Zeobelein, "Internet Operating System Counter," http://leb.net/hzo/ioscount, 1999.**

[6] **L. Torvalds, "The Linux Edge," in *Open-Source Software Development*, CACM, April 1999, Tim O'Reilly, Ed., pp. 38-40.**

[7] **P. Deutsch, A. Emtage, M. Koster, and M. Stumpf, "Publishing Information on the Internet with Anonymous FTP (Working Draft)," 1995.**

[8] **D. Weiss, "Towards An Efficient, Scalable Replication Mechanism for the I2-DSI Project," School of Information and Library Science, University of North Carolina at Chapel Hill, Master's Thesis, 1999.**

Appendix A

The Open Source Definition, Version 1.7
(reproduced from http://opensource.org/osd.html)

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale. (rationale)

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost -- preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed. (rationale)

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software. (rationale)

4. Integrity of The Author's Source Code.

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software. (rationale)

5. No Discrimination Against Persons or Groups.

The license must not discriminate against any person or group of persons. (rationale)

6. No Discrimination Against Fields of Endeavor.

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research. (rationale)

7. Distribution of License.

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties. (rationale)

8. License Must Not Be Specific to a Product.

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution. (rationale)

9. License Must Not Contaminate Other Software.

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software. (rationale)

Conformance

(This section is not part of the Open Source Definition.)

We think the Open Source Definition captures what the great majority of the software community originally meant, and still mean, by the term "Open Source". However, the term has become widely used and its meaning has lost some precision. The **OSI Certified** mark is OSI's way of certifying that the license under which the software is distributed conforms to the OSD; the generic term "Open Source" cannot provide that assurance, but we still encourage use of the term "Open Source" to mean conformance to the OSD. For information about the **OSI Certified** mark, and for a list of licenses that OSI has approved as conforming to the OSD, see this page.

Bruce Perens wrote the first draft of this document as `The Debian Free Software Guidelines', and refined it using the comments of the Debian developers in a month-long e-mail conference in June, 1997. He removed the Debian-specific references from the document to create the `Open Source Definition'.

Appendix B

Annotated Linux Software Map Template
(excerpted from http://metalab.unc.edu/pub/Linux/docs/linux-software-map/lsm-template)

Begin3
Title: The name of the package. Please use the same title for the LSM entry of each version, so as to make it easier to find entries for new versions of packages that already have one in the data base.
Version: Version number or other designation. Use a date if nothing else is appropriate.
Entered-date: Date in format ddMMMyy of when the LSM entry was last modified, where dd is 2-digit day of month, MMM is ALL-CAPITALIZED first 3 English month letters, and yy is last two digits of the year in the Gregorian calendar. Note that you should fill in both Version and Entered-date.
Description: Short description of the package.
Keywords: A short list of carefully selected keywords that describe the package.
Author: Original author(s) of package. In RFC822 format (i.e., something that will fit into a From: or To: header of a normal Internet mail message). Preferred format:

mailname@site.domain.top (Full name)

Other formats will be converted to this format, if time and energy of LSM maintainer will allow it.

Multiple persons may be given, one per line.
Maintained-by: Maintainer(s) of Linux port. Same format as Author.
Primary-site: A specification of on which site, in which directory, and which files are part of the package. First line gives site and base directory, the rest give the sizes and names of all files. Names are either relative to the base directory, or full pathnames. If the ftp site does not use Unix style pathname syntax, then the full pathname must be given every time. The pathname must not contain spaces. Example:

Primary-site: sunsite.unc.edu /pub/Linux/docs
10kB lsm-1994.01.01.tar.gz
997 lsm-template
22 M /pub/Linux/util/lsm-util.tar.gz

The file size may be given in bytes (no suffix), kilobytes (k, kb), or megabytes (M, MB). The suffix may be separated with spaces, and may be in upper case or lower case. The

size can be left off.

For very large packages that are contained within one directory (say, a distribution), only the directory need be listed. Adding a trailing slash makes it clear that it is a directory.

The filename should be the final location, not an "incoming" directory. If you don't know the final location, at least make a good guess (since files _will_ be moved from incoming, it is not a good guess).

Alternate-site: One alternate site may be given. It should not be a site that mirrors the primary site (these are best found from a list of mirror sites), but should be one that maintained separately. More sites carrying the package can be found using Archie. The syntax is the same as for Primary-site, but if there is only one line (i.e., no files are specified), they are assumed to be the same as for Primary-site.

```
Alternate-site: ftp.funet.fi /pub/OS/Linux/doc/lsm
Alternate-site: foo.bar /pub/lsm
                11 kB lsm-1994-01-01.cpio.Z
                0.1 kB lsm-template.Z
                22 MB lsm-util.tar.gz
```

Original-site: The original package, if this is a port to Linux. Syntax is as in Primary-site, with the same handling of missing filenames as in Alternate-site.

Platforms: Software or hardware that is required, if unusual. A C compiler or floppy disk would not be unusual, but a Python interpreter or tape drive probably would be. If the requirements are evident from the description, it need not be repeated here.

Copying-policy: Copying policy. Use "GPL" for GNU Public License, "BSD" for the Berkeley style of copyright, "Shareware" for shareware, and some other description for other styles of copyrights. If the use or copying requires payment, it must be indicated.

End