

# On the Pareto distribution of Sourceforge projects

Francis Hunt<sup>1</sup> and Paul Johnson<sup>2</sup>

<sup>1</sup> Centre for Technology Management, Cambridge University Engineering Department, Mill Lane, Cambridge CB2 1RX

<sup>2</sup> Marconi Labs, Gates Building, J J Thomson Avenue, Cambridge CB3 0FD

## Abstract

Open source software has risen to prominence within the last decade, largely due to the success of well known projects such as the GNU/Linux operating system and the Apache web server, amongst others. Their significant commercial impact, with GNU/Linux reportedly running on 25% of server machines and Apache on 60% of web servers, has prompted many companies who use and who develop software to reassess their traditional modes of functioning. A number of companies such as IBM, HP and Sun have invested significantly in developing open source software. Much early written work on open source software development aimed at raising awareness and advocating its uptake. More recently the interest has been in quantifying and qualifying the advantages, disadvantages and other features of open source software. This paper aims to contribute in this second area.

Most work on open source implicitly treats all projects as equivalent, for want of ways of classifying them. Benefits of 'typical' projects are claimed, with little attention to what constitutes a 'typical' project. In this paper we look at data available on SourceForge, a web site hosting upward of 30,000 open source projects and characterise the distribution of projects. Considering the number of downloads per week of the software, we show that for the most part the data follows a Pareto type distribution i.e. there are a small number of exceptionally popular projects, most projects being much less popular, and the number of projects with more than a given number of downloads tails off exponentially. We offer explanations for this distribution and for the places where the actual distribution deviates from the model and propose ways that these explanations could be tested. In particular there seem to be fewer than expected projects with a small number of weekly downloads. Likely explanations for this would seem to be either that projects with a small number of downloads per week do not tend to use SourceForge, or that this small number of downloads indicates a low level of interest in the project and such projects are inherently unstable (either they die or become more popular).

Two practical applications of this work are: it is useful for people or companies starting an Open Source project to have an idea of what a 'typical' project might entail; secondly, it enables analysis of best practice and benefits to be tied to some sort of classification of projects and allows questions such as how benefits scale with project size to be examined in detail.

## Introduction

Open source software has risen to prominence within the last decade, largely due to the success of well known projects such as the GNU/Linux operating system and Apache web server, amongst others. Their significant commercial impact, with GNU/Linux reportedly running on 25% of server machines and Apache on 60% of web servers, has prompted many companies who use and who develop software to reassess their traditional modes of functioning. A number of companies such as IBM, HP and Sun have invested significantly in developing open source software. Much early written work on open source software development was aimed at raising awareness and advocating its uptake, the most famous example being Eric Raymond's "The Cathedral and the Bazaar" essay (1999). More recently the interest has been in quantifying and qualifying the advantages, disadvantages and other features of open source software. This paper aims to contribute in this second area.

This paper analyses and draws conclusions from statistics on the Sourceforge website, a site hosting over 30,000 open source software projects. This site was set up in November 1999, providing freely available infrastructure for running open source software projects. It is the premier site for hosting open source projects, and conclusions about open source development drawn from the Sourceforge site are likely to be relevant to a broad range of open source projects. In this paper we analyse the distribution of projects according to the frequency that their software is downloaded.

Quantitative studies of open source software development are relatively rare. Notable exceptions include studies of the Apache project (Mockus, Fielding et al. 2000), the Gnome project (Koch and Schneider 2000) and of Linux e.g.(Wheeler 2001). Rene

Kienzle (2001) has also investigated the statistics on Sourceforge, in particular the number of developers associated with different projects. His results do not overlap those of this paper and can be considered complementary.

The rest of this paper is structured as follows: first we discuss the data used in the analysis; then we examine the distribution of projects according to the frequency of project downloads; finally we draw conclusions and suggest avenues of further research.

## **Data**

Sourceforge hosts over 30,000 open source development projects. It provides free of charge infrastructure for running such projects, including version control, space for a project website, bug tracking and mailing lists. Significantly (for this paper) it also collects and displays statistics on the various projects, such as the number of times a piece of software has been downloaded on each of the last 30 days, the number of times its web pages have been viewed and the number of times software has been checked in to the version control system. These are all measures of project activity; other such measures which are not displayed on the statistics page include activity on the project mailing lists, bugs reported and new official releases of code. A number of measures are combined into an overall measure of project activity<sup>1</sup>, and the most active projects are listed.

Having originally obtained permission to study their website in February 2001, we collected the statistics from all the projects listed on the most active project list on 22<sup>nd</sup> October and again on the 22<sup>nd</sup> November, providing two contiguous sets of 30 days<sup>2</sup>. The statistics collected for each day and project were the number of downloads, the number of webpage views and the number of CVS commits.

A natural first question is how reliable are these data. The statistics collection and processing routines on Sourceforge are known to contain bugs (there are open bug reports #462957 from the 19<sup>th</sup> September on download statistics; #455161 from 24<sup>th</sup> August on CVS statistics; and #451204 from 15<sup>th</sup> August on the pageview statistics). However we believe that the download statistics are in general reliable, but care is needed in making generalisations from outliers. In terms of the reliability of the data capture operation from the Sourceforge site, a visual check was made between the data as displayed on Sourceforge and the data as recorded in our database on five of the projects and we are moderately confident that there were no systematic errors in transcribing the data.

The second question is how representative are these data. This question splits into: how representative the data are of active projects on Sourceforge; and how representative the data are of open source projects in general. The data is representative of active projects on Sourceforge since it contains all active projects. Sourceforge is also the premier site for hosting open source projects, so it is plausible that the data are representative of open source software development as a whole. Some very well known projects do not use Sourceforge e.g. Linux, Apache, Mozilla. There is

---

<sup>1</sup> The actual metric used in the source code is (Scholl 2001):  
 $\log(3*\text{forum\_msgs}) + \log(4*\text{project\_tasks}) + \log(3*\text{bugs}) + \log(10*\text{patches})$   
 $+ \log(5*\text{supports}) + \log(\text{cvs\_commits}) + \log(5*\text{developers}) + \log(5*\text{filerereleases})$   
 $+ \log(0.3*\text{downloads}) * \text{servey\_rating\_agregate}.$

<sup>2</sup> This data is available for download from <http://www-mmd.eng.cam.ac.uk/people/fhh10/fhh10.htm>. Although the two sets of 30 days are contiguous, data for 27<sup>th</sup> October is missing for unknown reasons.

also a cluster of GNU projects that are hosted elsewhere and a recently formed competitor to Sourceforge called Savannah<sup>3</sup>. There are also a number of open source projects that are hosted by companies. Nonetheless, it seems safe to assume that the majority of active open source projects in the world are hosted on Sourceforge and hence studying the Sourceforge data tells us something about open source development.

## Analysis

We investigate 3 issues in this paper:

- time series of total sourceforge downloads
- cross sectional distribution of projects at a moment in time
- differential behaviour of segments of the cross section

### *Time series of total sourceforge downloads*

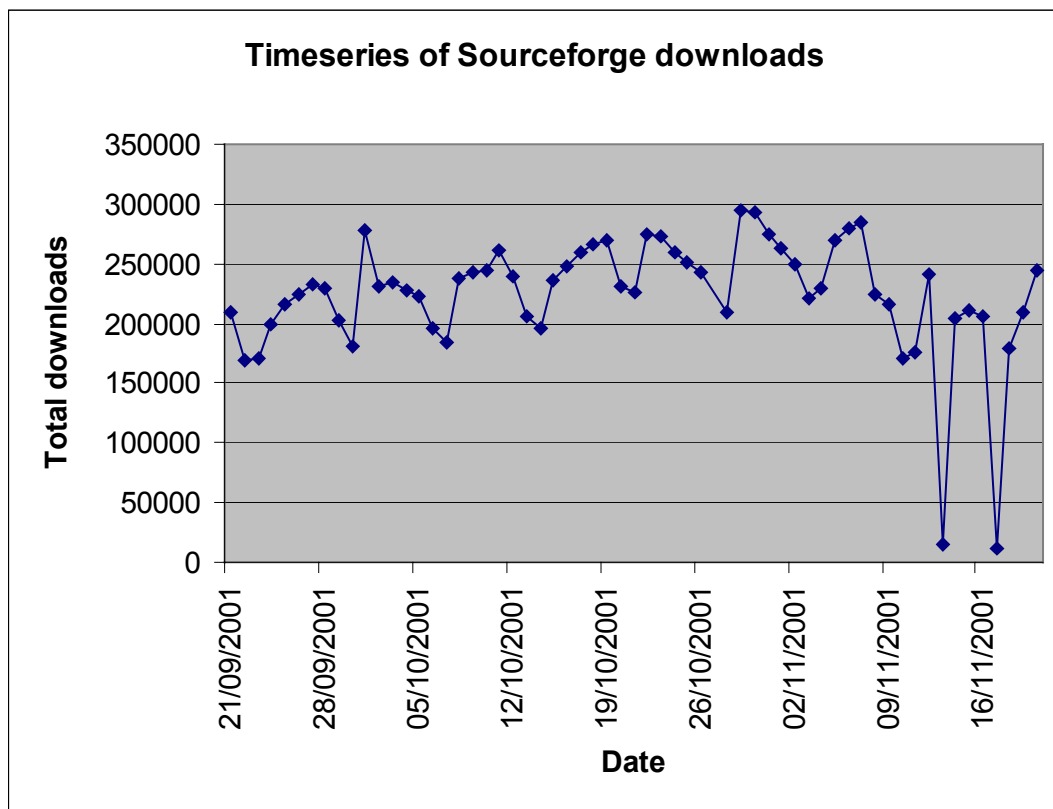


Figure 1: Total downloads from Sourceforge site

Summing the downloads on all the active projects and plotting the time series of the total downloads in figure 1 immediately highlights a number of interesting points. Firstly the initial six weeks indicate a general increase in download activity, though it is possible that this increase is in fact an a return to normality after terrorist attacks in

---

<sup>3</sup> Savannah was set up by people who thought it ethically unacceptable that VA, the company which own Sourceforge, should produce an enhanced closed source version of the Sourceforge software.

the US on September 11. Secondly, something peculiar happens in mid November, in particular on Tuesday 13<sup>th</sup> and Saturday 17<sup>th</sup> November. We do not use this data in our subsequent analysis, because of this unexplained turbulence. Thirdly there is a noticeable weekly cycle. The first day plotted is Friday 21<sup>st</sup> September and the troughs of the weekly cycle occur at weekends. This is perhaps slightly surprising and suggestive of commercial use of software on Sourceforge, or at least use of commercial resources to download the software.

### **Cross sectional distribution**

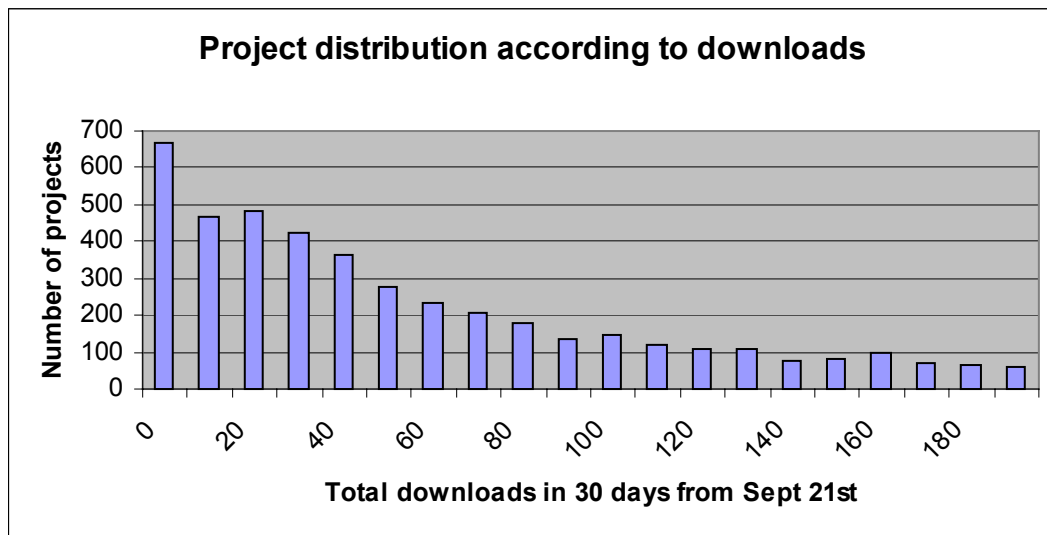


Figure 2: Sourceforge project distribution

If we examine the distribution of projects according to the total number of downloads they received in the 30 days from September 21<sup>st</sup>, plotting in figure 2 the projects as a histogram with bins width 10, we obtain a very heavily skewed distribution with a tail that extends out to more than 600,000 downloads. The median number of monthly downloads is 70 i.e. half the active projects have between 0 and 70 downloads whilst the other half have between 70 and 600,000 downloads. Such heavily skewed distributions, though not as common as the omnipresent normal distribution, do occur in a wide range of phenomena such as the distribution of incomes (Pareto 1896), earthquake severities (Richter 1958), forest fire sizes, word usage frequencies (Zipf 1949) and web site popularities (Adamic and Huberman 2000). Of particular interest in all the examples just mentioned is that the distribution tails off exponentially e.g. in the case of incomes, this means the ratio of the number of people who earn more than you to the number of people who earn 10 times more than you is the same regardless of your income. Distributions with this property are variously called Pareto, Zipf or power law distributions. Plotting the logarithm of the frequency of an event against the logarithm of its 'size' yields a straight line graph for the tail of such a distribution.

In figure 3 we plot the number of projects with a given number of downloads. We see that the distribution does have the characteristic Pareto tail. There are a number of proposed explanations for such distributions e.g (Bak 1997), but most relevant here is likely to be the winner-takes-all nature of project popularity: as a project grows in popularity it becomes more attractive since there is more likely to be good documentation, knowledgeable people to provide support, further development work, software tools that work with it, ...etc.

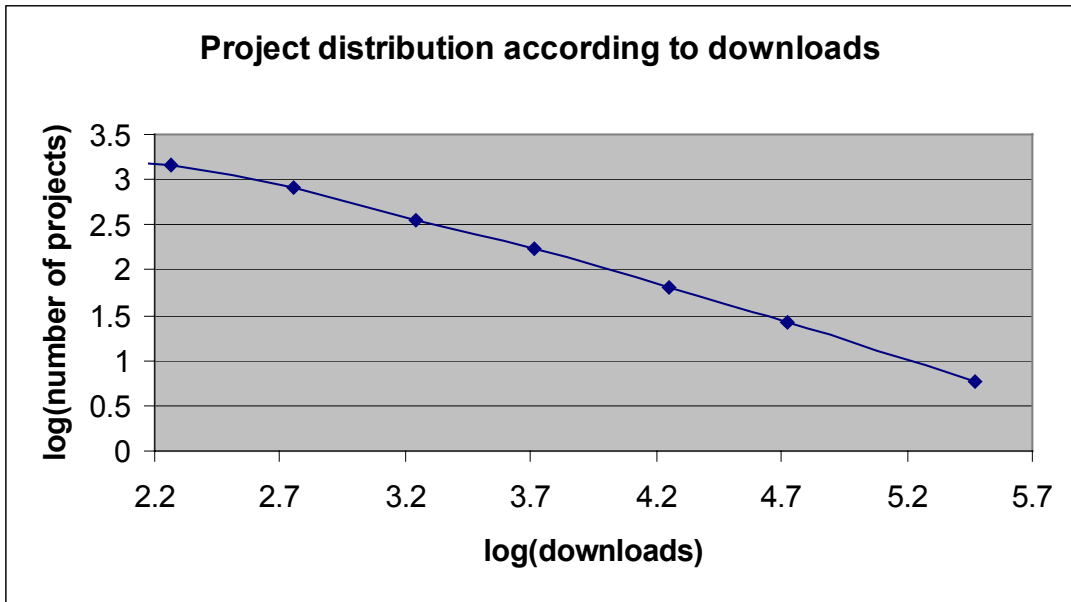


Figure 3: Log-log plot of download frequencies exhibiting a power law over three orders of magnitude

Looking more closely at the number of project downloads in figure 4 using reduced bin sizes we see that there are fewer than expected projects with a low number of downloads. Having checked our data collection, this does not seem to be due to an error in the data collection, or to be an artefact of the sampling strategy. Thus projects with low numbers of downloads are relatively uncommon on Sourceforge. Two competing explanations for this are that either projects with low numbers of

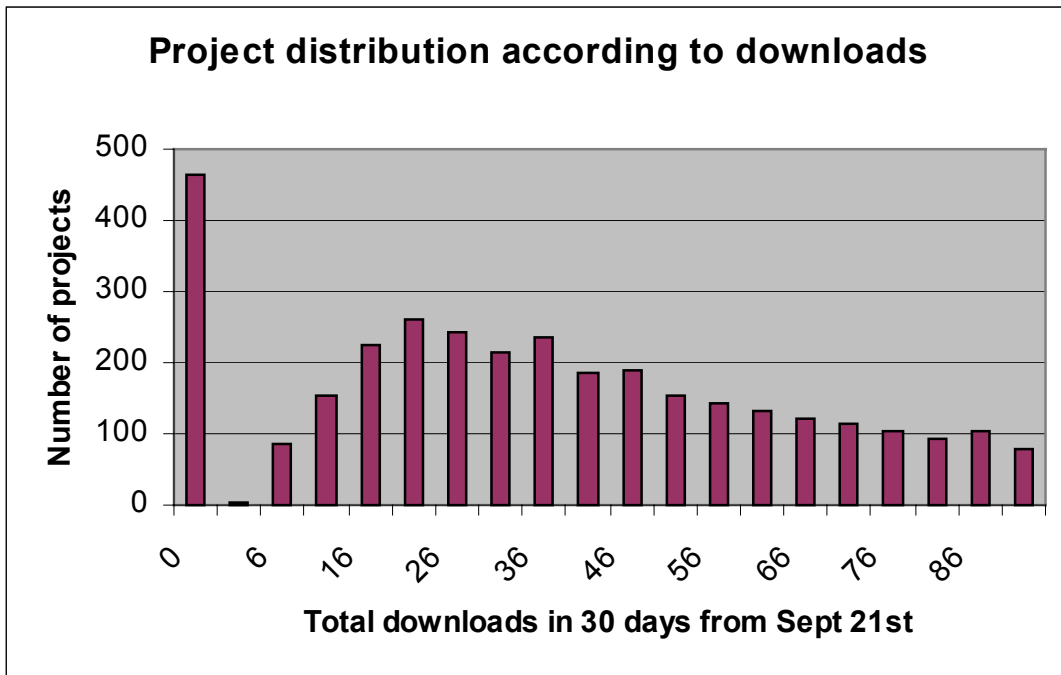


Figure 4: Distribution of projects with few total downloads

downloads do not use Sourceforge, or that such projects are inherently unstable i.e either they rapidly grow or they rapidly die off. We investigate whether this is the case in the next section.

### **Differential behaviours of segments of the cross-section**

In this section we examine firstly the relative growth in downloads for projects with various numbers of downloads. Are projects with low numbers of downloads more likely to grow or to shrink than other projects? To measure the relative growth of each project, we divided the slope of the best fit line through the downloads of 30 days from September 21st, by the mean number of downloads in these 30 days. Averaging these relative growth figures over bins of ~200 projects and plotting this relative growth against the log of the mean number of downloads, we see clearly in figure 5 that low download projects grow faster than high download projects. (This does not seem to be a quantization effect i.e. due to number of downloads being an integer, since such an effect should not bias the direction of growth.)

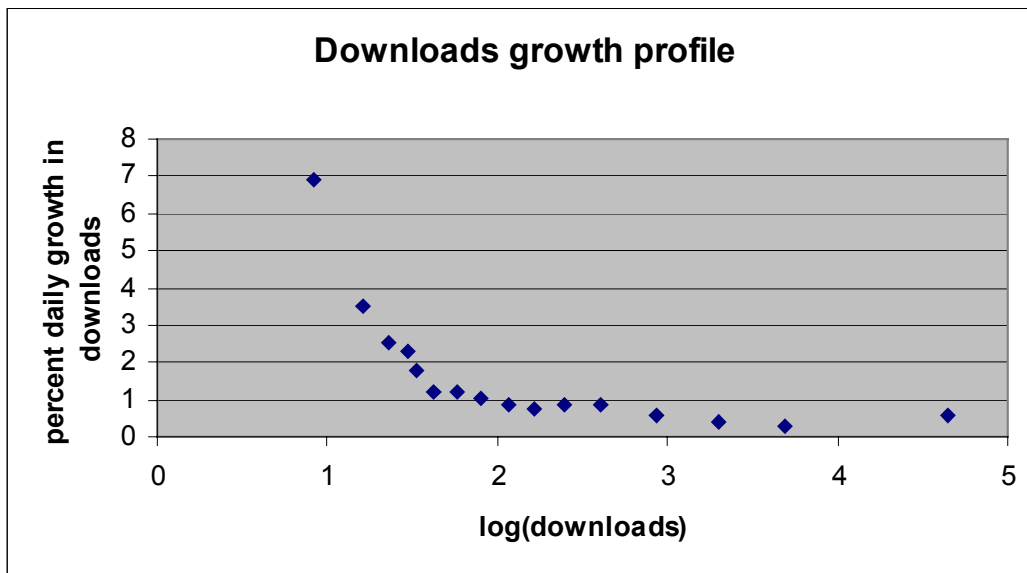


Figure 5: Growth profile of projects with differing numbers of downloads

It is worth noting that although the relative growth of projects was stronger for projects with few downloads, the overall trend in total downloads in the period observed is upwards. It may be that the correct conclusion is that overall trend in downloads over all projects is amplified in the projects with few downloads. To test this, we would need data for a period when the overall number of downloads from Sourceforge was declining<sup>4</sup>.

The second issue we consider in this section is whether projects with low numbers of downloads are more unstable. Figure 5 has shown that projects with low numbers of downloads on average grow more strongly than those with high numbers. Is this true of all such projects, or does this average disguise a wide variety of outcomes? Intuitively, it is to be expected that a project with a low number of downloads is more easily influenced by external events. In figure 6 we plot the standard deviation of the

---

<sup>4</sup> If such a period does not exist, then the hypothesis could still be tested by looking at periods of differing overall growth in downloads from Sourceforge.

number of downloads over the 30 days from September 21, normalised by the mean number of downloads. As expected this show that projects with low numbers of downloads are subject to greater variations in downloads.

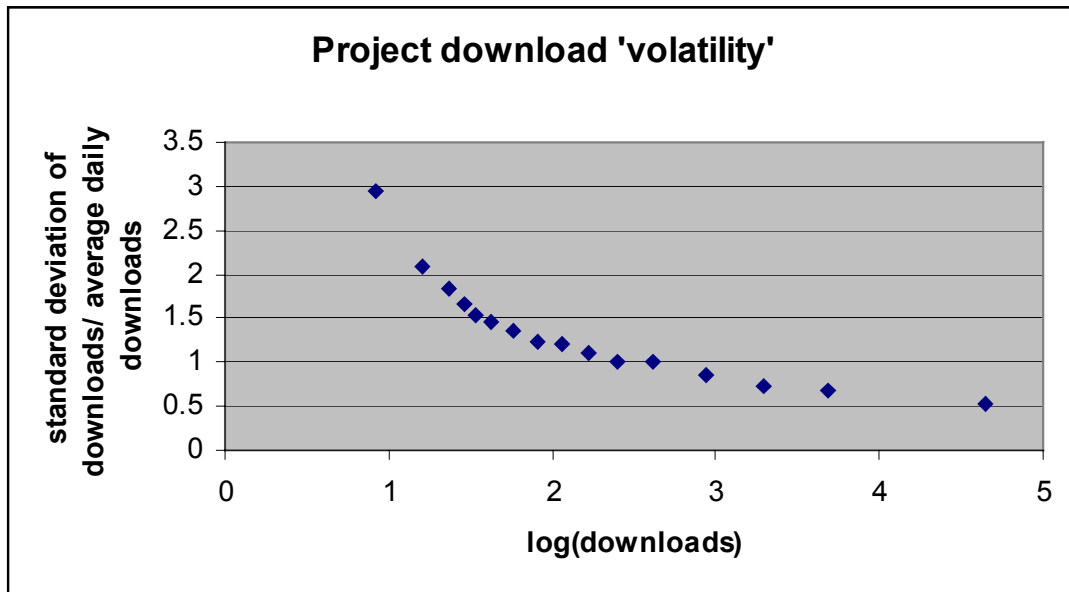


Figure 6: Profile of project download 'volatility'

## **Conclusions and further work**

In this paper we have characterised the distribution of projects according to the number of downloads they receive. The open source projects most frequently studied (such as Linux, Apache, Mozilla) lie at the extreme end of a highly skewed distribution. Although studying these may be a fruitful approach in identifying 'best practice', it is potentially misleading since the problems and solutions found in the running of these projects may be inappropriate for most projects. Studying instead the median projects may prove useful.

In setting up a site to host open source projects, it is reasonable to assume that the resource requirements of the various projects will also be distributed according to a Pareto distribution. This is helpful in planning support for these projects.

The questions considered and answered in this paper bear further study, particularly against different samples of Sourceforge data. Given the richness of the data set there are also many unasked questions – readers are invited to download the dataset from the CTM website and experiment.

## **References**

Adamic, L. A. and Huberman, B. A. (2000). "The nature of markets in the World Wide Web." *Quarterly Journal of Electronic Commerce* 1(1): 5-12.

- Bak, P. (1997). How nature works : the science of self-organized criticality. Oxford, Oxford University Press.
- Kienzle, R. (2001). "Sourceforge preliminary project analysis." available online at <http://www.osstrategy.com/sfreport>
- Koch, S. and Schneider, G. (2000). "Results from software engineering research into open source development projects using public data." available online at <http://opensource.mit.edu/papers/koch-ossoftwareengineering.pdf>
- Mockus, A., Fielding, T., et al. (2000). "A case study of open source development: the Apache server". 22nd International Conference on Software Engineering, Limerick, Ireland.
- Pareto, V. (1896). Course of Political Economy. Lausanne.
- Raymond, E. S. (1999). "The cathedral and the bazaar" in The cathedral and the bazaar : musings on Linux and open source by an accidental revolutionary. Sebastopol, CA ; Beijing, O'Reilly.
- Richter, C. F. (1958). Elementary seismology. San Francisco ; London, W. H. Freeman.
- Scholl, K.-U. (2001). "RE: How are statistics calculated?" posting to discussion forum available online at [http://sourceforge.net/forum/message.php?msg\\_id=222302](http://sourceforge.net/forum/message.php?msg_id=222302)
- Wheeler, D. A. (2001). "More than a gigabuck: estimating GNU/Linux's size." available online at <http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html>
- Zipf, G. K. (1949). Human behaviour and the principle of least effort : an introduction to human ecology. Cambridge, Mass., Addison-Wesley Press.