

Secrets of Free Open Source Software

Rafid Daoud
SENECA COLLEGE

Table of Contents

Introduction	2
First Look on Open Source	3
Open Source in Action	5
Different types of licenses	5
Economics of Open Source	7
Costs Saving	8
Sale of Hardware.....	10
Supporting of OSS	10
Dual Licensing	10
Donations.....	11
An Overview of Databases.....	11
Open Source Database Management System Software Examples.....	12
Open vs Closed Source Databases	13
Open Source Advantages.....	13
Open Source Disadvantages	14
Proprietary Advantages	14
Proprietary Disadvantages.....	15
The Future.....	15
Bash on Windows 10	16
Final Remarks.....	17
References	18

Abstract

Purpose – this paper aims to study the economics of open source software and how things work. Followed by a comparison of a commercial versus an open source software that is free to use. The paper will focus on open source database software that is currently available on the market and in use by consumers of different sizes, whether they are a single individual or an enterprise company.

Introduction

In today's world, people are heavily reliant on technology; everyone owns at least one device. However, these devices would be useless without software to drive them and make them into something that is usable. Software is created to make our daily lives better, and help us improve it. It increases the efficiency of accomplishing tasks that range in a wide variety of scenarios. These tasks can differ in their complexity and length; these tasks can be as simple as pressing 'print' to print a report, all the way to controlling and running a nuclear power plant. All of this can be achieved by using software. However, most software applications are commercial, and the consumer is required to pay a given amount in order to harness its benefits. This is not the case with open source software where it is available freely to consumers. This paper will be exploring the economics of open source software and looking into how open source database software has the potential to generate more revenue as opposed to the proprietary solution.

First Look on Open Source

To start with, we can ask the question, what is open source software? According to Wikipedia Open-source software (OSS) is computer software with its source code made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose. Open-source software may be developed in a collaborative public manner.

While it seems to be that open source is a new approach to software currently, the basics of the origins are much older. In fact, there had been a tradition of sharing and cooperation in software development. However, in the more recent years, this activity has expanded in scale and formalization with the help of the internet (Lerner and Tirole 2002, 200).

In the early years between the 1960s and the 1980s, many efforts were focusing on the development of an operating system that would be able to run on many platforms. This process was quickly sped through Usenet, a computer network that began in 1979 to link all of the UNIX programming community. In turn, the development of the operating system saw a massive increase in development and enhancements as programmers from universities and corporations began to participate in building it. This was done through an informal way and no property rights (Lerner and Tirole 2002, 201).

As UNIX was developed, there were many issues with some rights of intellectual property. Thus, the second stage of open source started, from the 1980s to the 1990s. This is where people started to have a more formalized approach to such projects. This was also when the Free Software Foundation was established. Their goal was to develop and distribute a wide variety of software without any costs (Lerner and Tirole 2002, 201-202).

Around the 1990s is when the Internet came about, and it took open source to yet another stage. With the introduction of the Internet, the number of contributions to open source projects significantly climbed higher. This also allowed the birth of many new open source projects for people to take on. However, not everything is positive in

the world, and along with the benefits of the internet to these projects came some downsides as well. First, projects began to go in different directions; each project would have many other variants. This was due to the fact that each person had a different vision of design for a certain aspect of the project. Second, there was a small amount focusing on the documentation of these projects. In addition, there was no support, user interfaces, and backwards compatibility in some of these open source projects (Lerner and Tirole 2002, 202-203).

Software developers from around the world worked on this single project where all of them were from different organizations and backgrounds. They were encouraged to work on these projects due to the following three factors:

I. The quick distribution of open source software. A few open source projects dominate their categories in the market. One of the notable mentions is Apache web server. Another is the Linux operating system in the personal computer system category. It has between seven and twenty million users worldwide, with a 200% rate of growth annually (Lerner and Tirole 2002, 197).

II. The significant capital investments in open source development projects. Large firms and companies such as Hewlett-Packard, IBM, and Sun have launched their own open source projects. Also, companies such as Red Hat and VA Linus who specialize in commercializing open source software have completed initial public offerings. Meanwhile, other open source companies like Coolab.Net and Sendmail have received venture capital financing (Lerner and Tirole 2002, 198).

III. The new organization structure. The open source projects are very collaborative in nature which has been noted by businesses and technical press as a new important organizational innovation.

According to Riehle a foundation is accountable for the following when working on a project:

- organizing the project community;

Open source has come a long way since it first started. It is only getting better as time marches forward. It has evolved from a casual approach to a more standardized procedure for developing software. There are bodies that govern an open source project in order for things to run smoothly during the development and maintenance phases of the software.

- actively marketing the software;
- clarifying and managing intellectual property rights;
- setting strategic directions for the software;
- responding and remaining accountable to its members; and
- running all relevant back office processes.

Open Source in Action

There are different types of licenses that could apply to a project. Choosing a suitable license for the software would depend on the goals a foundation is planning to achieve. They can make a choice between liberal more open licenses versus a more restrictive one. Also, they could have one or a combination to license the software.

Different types of licenses

This section of the paper will be discussing some of the most common licenses that open sources generally use.

Copylefts are popularly used with open source projects. To copyleft a software, the contributor, alongside copyrighting the software to themselves, also signs a general public license. With this agreement, everyone would have free access to the software. However, it is protected from becoming someone else's private intellectual property (Mustonen 2003). In other words, copyleft can be thought of as an extension to having a license.

GNU General Public License

The GNU General Public License (GNU GPL or GNU) is one of the most common free software licenses that are used. It was published by the Free Software Foundation and released in June 2007. The license allows the end users whether they are a company, organization or individual to run, share and modify the software as they see fit ("Gnu.org" 2016).

The GPL makes use of copyleft so that the freedoms of the license are preserved when the work is published and distributed even when the work is changed or appended to. In other words, the derived

work must have the same licenses of the original work. Thus, making all the work and what follows absolutely free for everyone (“Gnu.org” 2016).

Apache License

The Apache License is a permissive free software license made by the Apache Software Foundation. This license requires the preservation of the copyright notice and disclaimer. It allows the user of the software to use the software in any way, distribute it, modify it and even distribute the modified version with the stipulation of the terms of the license without the hassle of paying royalties. Apache license was last published in January 2004 (“Licenses” 2016).

The Apache license is compatible with the General Public License, meaning that they could be combined together as one license. However, this case only works if version 3 of the license is used. The previous versions are not compatible with the Apache license (“Licenses” 2016).

BSD License

Published in 1990 by the Regents of the University of California. The BSD license is a permissive free software license that has minimal restrictions on the redistribution of the software that is using it (“BSD License Definition” 2016).

Common Public License

The Common Public License is a free software or open-source software license written by IBM. The Open Source Initiative and the Free Software Foundation approve of its terms and conditions. It was published in May 2001.

The CPL was designed to support and encourage collaboration on open source development. It has some similarities to the General Public License, but the variation is in the way the distribution of modified software works. One must make the source code of the newly revised version of the software available to others (“Common Public License, Version 1.0 (CPL-1.0) | Open Source Initiative” 2016).

There are different types of licenses that an open source software could have. There can be a single license or a combination of several to keep things intact and protect the intellectual property of the software developers. Some licenses are more open than others, thus it depends on the governing body to decide how license their project.

It lacks compatibility with the General Public License because of one of the sections “choice of law”. Also, the fact that CPL has a different way of handling copyleft requirements.

MIT License

The MIT license is another free software license which comes from Massachusetts Institute of Technology, polished in 1988 making it one of the oldest licenses that were written. It places a very minimal amount of restrictions to reuse software code, thus making it highly compatible with other licenses. With the MIT license, one could reuse it within a proprietary software given that they provide a copy of the license (“MIT License” 2016).

According to GitHub, a website allowing the hosting of source code repositories giving the users the ability to manage their source code and its revisions. In 2015, the MIT license had become one of the most popular licenses used after the GPL (“Open source license usage on GitHub.com · GitHub” 2016).

Mozilla Public License

This license mainly concerns the Mozilla products. It is written by the Mozilla Foundation and was last published in January 2012. MPL is a free, open source and detailed software license where it is considered to be a hybrid of the GNU General Public License and a modified version of the BSD license. The goal of combining the two licenses is to solve the issue of proprietary and open source developers. Some of the components of the work would be available freely versus other elements that would be licensed under the BSD license to keep them for more proprietary use (“Mozilla Public Licence” 2016).

Economics of Open Source

Consider the following famous quotation said by Stallman:

The idea that the proprietary software social system - the system that says you are not allowed to share or change software - is unsocial, that it is

unethical, that it is simply wrong may come as a surprise to some people. But what else can we say about a system based on dividing the public and keeping users helpless?

Simply put, proprietary software publishers are implanting the idea in people's mind to look at things from one perspective. Thus, we should change our view of things and find new ways to examine issues.

Some questions need to be answered when looking into open source software projects. The biggest question that often arises when discussing this topic is, why should a programmer contribute to a said project? At first, one may assume that it might be an act of kindness from the programmer that they care about others' concern, but that explanation can only go so far. Upon a closer look, there are many other valid reasons as to why this happens. In many cases, solutions developed by particular users for individual issues have become a more general solution for wide classes of users (Lerner and Tirole 2002, 198-199).

People may contribute as a gesture of goodwill, but also gain a good reputation in the community which may translate into a higher paying job (Hawkins 2004, 104).

Let us examine how open source software generates revenue.

Costs Saving

One of the most economic motivations to using open source software is to save costs, they are cut down by saving on development time, and this is achieved by splitting the tasks of development amongst many parties (Riehle 2010, 87). This is the most accepted reason for why open source projects are chosen over a more proprietary solution.

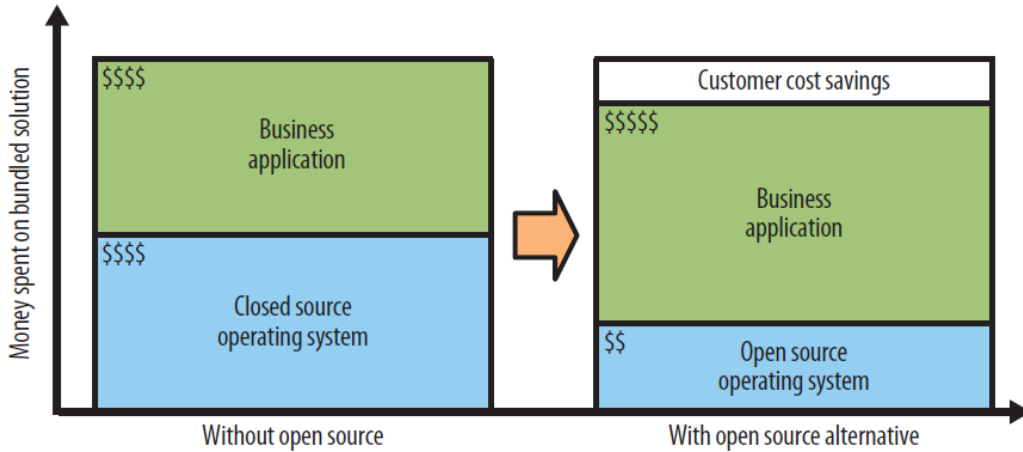


Figure 1 – Riehle, Dirk. “The support of open source software lets the vendors sell at a higher price.” Digital image. The Economic Case for Open Source Foundations. 2010. Accessed April 11, 2016.

As seen from the figure 1 above, both the consumer and the vendor will save money by choosing an open source operating system. For example, if a consumer would like to purchase the Adobe suite. They would have to pay for the licensing fees and on top of that they would have to buy more licenses for the operating system. However, with an open source operating system like Linux, they would save costs on buying the proprietary operating system. Thus, it is a win-win situation for both Adobe, the vendor, and the consumer.

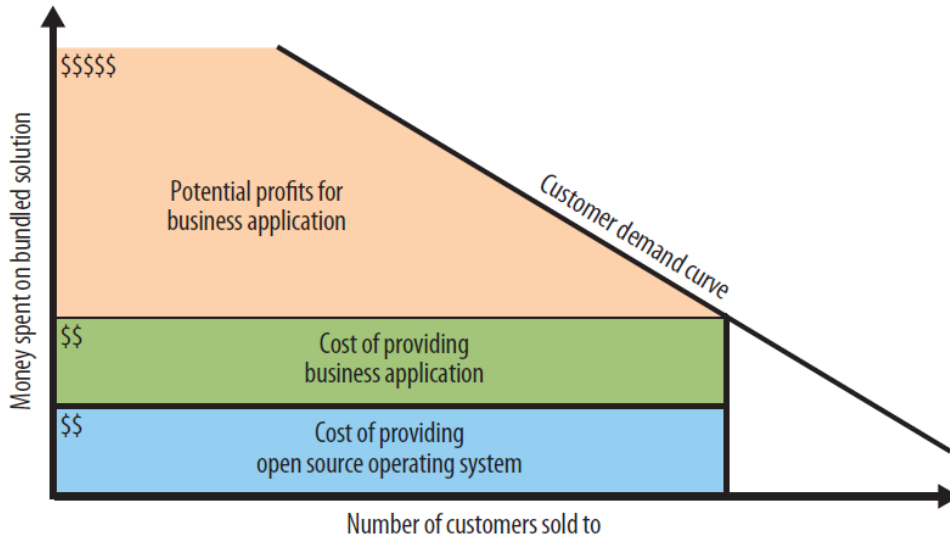


Figure 2 - Riehle, Dirk. “The support of open source software lets software firms sell more customers.” Digital image. The Economic Case for Open Source Foundations. 2010. Accessed April 11, 2016.

By using an open source operating system, the developer is able to save more money. Since the utilization of an open source operating system costs less to create the vendor selling their applications at a lower price attracting more customers. Therefore, the vendor of the application will generate more sales leading to more profit. This is illustrated in figure 2.

Sale of Hardware

Companies and organizations could grow their profits through the sale of the hardware when it is combined with free open source software to run the hardware sold to consumers (Hawkins 2004, 103). In other words, software is not seen as a product; it is only supplementary to the hardware. The software is only there to drive up the sale of hardware components. An example of this model is Apple Computers, they make their own software, but their company is focused on selling their devices.

Supporting of OSS

Vendors of proprietary software often try to deviate customers from open source solutions as they do not have much support to go with it. However, this is not the case; there is plenty of support found online, in newsgroups, websites, and mailing lists. In addition, support for an open source product can be purchased from vendors like IBM and Oracle (Hawkins 2004, 105). Thus, a consumer is saving on the purchase but only pays for support when needed from the vendor of their choosing.

Red Hat is famous for supporting open source software. They rely on this model to generate their revenue (Hawkins 2004, 105). The customer could download the software from Red Hat's website at no cost; then they would purchase a contract to support the software, such as helping in the installation process and setting up of the software.

Dual Licensing

With this method, a company may offer the same product under an open source license and a commercial license. The consumer has the choice to pick the plain product or with service level agreements,

To generate revenue with open source software, there are many approaches on doing so, such as saving on costs of purchasing the software and its development. Using OSS will increase the complementary sale of hardware. Selling professional consulting and support services to consumers of OSS. Another way is by setting up a dual-licenses on the software where one of the licenses is proprietary will encourage consumers to buy the software for the added benefits and advantages.

warranties, etc. with a commercial license. The vendors tend to apply different versions to limit the functionality of the software (Popp 2011). Thus, the consumers are encouraged to make the purchase.

Donations

Free open source software are always looking for the most support they can get. One of these methods is donations. Some open source projects cannot go further because of financial situations such as legal fees, patents or expensive testing equipment. These donations usually come from individuals and organizations, and widespread projects are more likely to get more contributors. Open source projects that receive financial donations have a greater subscriber base per months as opposed to those that do not, to put this in perspective 54.5% greater (Sen, Singh, and Borle 2012).

DuckDuckGo, a search engine company, donated a total of \$125,000 to five different open source projects. Their primary goal was to support projects that dealt with security and privacy tools (“DuckDuckGo Blog” 2016).

An Overview of Databases

A database is a store for data on a computer system. It is an organized arrangement of collections of records or data. They are maintained by a program that would receive different kinds of queries to interact with it, based on the query the database would make decisions. The program is usually known as a database management system which is responsible for receiving of queries and running them in order to produce reports of information. The database is comprised of one or more table containing records of value for fields that belong to the object being referenced by the table. An example of the most database types is a relational database, where it would contain many tables that are linked by field keys. (Berrington 2013).

A database is comprised of one or more tables; each table contains a number of fields (columns) and records (rows), the intersection of the two results in a data value.

ID Number	Student	Date of Birth	Email	Phone
423423	John Smith	10/24/1993	jsmith@school.com	416-222-1234
234654	Jane Roberts	05/06/1990	jroberts@school.com	647-348-7765
138986	Aaron Paul	09/05/1989	apaul@school.com	416-324-6542
324645	Walter Green	11/11/1985	wgreen@school.com	647-123-1577
357853	Mary Allens	03/12/1991	mallens@school.com	647-1254-1221

Figure 3 – Basic database table with its main components highlighted.

Upon first look, a database may seem like a spreadsheet. However, it much more complicated than that, a database can retrieve data with a given specific conditions, update record data in bulk, cross-reference data from multiple tables, and it could perform complex calculations (Berrington 2013).

Open Source Database Management System Software Examples

This section will be focusing on some of the most popular solutions for open databases that are used in the market.

MySQL

It is the second most popular open source database (“MySQL :: About MySQL” 2016) and the second most commonly used relational database management system as of 2013. It is licensed under the GNU General Public License. However, it is also licensed under a variety of proprietary agreements. MySQL is one of the popular database choices for web applications as it comes in with the LAMP (Linux, Apache, MySQL, and Perl/PHP/Python) stack. It is a free open source software (“What Is LAMP (Linux, Apache, MySQL, PHP)? - Definition from WhatIs.com” 2016). Even large-scale websites such as Google, Facebook, and YouTube utilize MySQL to operate some of their databases (“MySQL :: About MySQL” 2016).

In summary, databases can be used to store a large amount of data and the user is able to manipulate or view it as they require.

MariaDB

This software was forked from the original MySQL and run by the original developers. It is licensed the same way as MySQL with GNU GPL. MariaDB initially released in January 2009. It is compatible with all of the leading Linux distributions such as Red Hat Enterprise Linux and CentOS (“About MariaDB | MariaDB” 2016).

PostgreSQL

Another open source database software is PostgreSQL initially releasing in July 1996. One of the differences it has compared to the other is that it is object-relational database management system. It was developed by PostgreSQL Global Development Group, in addition to many companies alongside individual contributors. It has its own permissive free software license (“PostgreSQL: About” 2016).

MongoDB

One of the newest free open software on the market that was released in 2009. Unlike the examples above, MongoDB is classified as NoSQL database (“NoSQL Databases Explained” 2016). Thus, it is not a relational database like the others. As of July 2015, MongoDB is the fourth most popular database software (“DB-Engines Ranking - Popularity Ranking of Database Management Systems” 2016). It is heavily relied on in document stores. MongoDB is licensed under a dual license of Apache and GNU GPL.

Open vs. Closed Source Databases

This section of the paper will be outlining the advantages and disadvantages of choosing which side. The comparison will be based on some of the key points highlighted by Astera, a data integration company based in California, United States.

Open Source Advantages

Source code control: Consumers have the ability to view the code, thus giving the capacity to change, modify or add to it as they see fit. This makes it easy to customize the software, in terms of adding new specific features to the given company’s business needs.

Cost: It is free just as the name of it implies. There is not cost for consumers to pay when acquiring the software. This process is as simple as just downloading the software from a website or a server hosting it.

Community: This is a great advantage for open source software especially if the community is active. The more eyes on the code, the more bugs will be found and fixed. The kinks can be ironed out quickly and submitted to the next patch of the project.

Open Source Disadvantages

Incomplete: Open source software may be immature, resulting in some incomplete features or work-in-progress features. However, if the function is vital to the success of the project, then it would be quickly implemented by the community.

Bug Fixes: Any bug that the software may come with may take a while for it to be fixed unless the open software's community is very active and involved.

Availability of Support: There is no dedicated support for the software. The consumer would have to find it themselves through different means. This can be avoided if the consumer would hire a professional party to do that for them.

Proprietary Advantages

Maturity: When acquiring a commercial software, the consumer would know that it is fully developed and tested before its release. Thus, it is ready for deployment with little or no time.

Support: There is a dedicated support personnel by the vendor to help the consumer with any issues that may arise in the lifetime of the software.

Ownership: The purchasing party would have a license entitling them to everything provided by the software vendor such as updates, bug fixes, and new versions.

Proprietary Disadvantages

Source code control: The consumer is not able to view the source code, in turn, they will not be able to modify anything to add any customization to fit their needs.

Payment: Often a proprietary software requires the consumer to pay a large initial cost. This may be a problem for small companies and even larger companies if they are unsure of what their software needs are, thus ending up wasting their capital.

When considering a software a firm should look into various solutions. Also, they need to outline their business needs and software requirements clearly so that they are able to choose the right solution. Some of the aspects they can consider include: payment, source code visibility, support, and completeness.

The Future

The future is looking bright for open source database systems as more and more companies are beginning to realize the true power and potential of free software. More applications are relying heavily on open source databases (Packer 2007). The rapid growth can be

303 systems in ranking, April 2016

Rank			DBMS	Database Model	Score		
Apr 2016	Mar 2016	Apr 2015			Apr 2016	Mar 2016	Apr 2015
1.	1.	1.	Oracle	Relational DBMS	1467.53	-4.48	+21.40
2.	2.	2.	MySQL +	Relational DBMS	1370.11	+22.39	+85.53
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1135.05	-1.45	-14.07
4.	4.	4.	MongoDB +	Document store	312.44	+7.11	+33.85
5.	5.	5.	PostgreSQL	Relational DBMS	303.73	+4.10	+35.41
6.	6.	6.	DB2	Relational DBMS	184.08	-3.85	-13.56
7.	7.	7.	Microsoft Access	Relational DBMS	131.97	-3.06	-10.22
8.	8.	8.	Cassandra +	Wide column store	129.67	-0.66	+24.78
9.	9.	↑ 10.	Redis +	Key-value store	111.24	+5.02	+16.69
10.	10.	↓ 9.	SQLite	Relational DBMS	107.96	+2.19	+5.67

Figure 3 – DB-Engines rankings of database management systems according to their popularity. Screenshot. April 2016. Accessed April 11, 2016. <https://www.db-engines.com/en/ranking>

observed from figure 4. Companies are driving that growth because of the cost savings that can be achieved.

With the above in mind, it does not mean that proprietary solution will go away anytime soon. As they run on many legacy systems and companies cannot afford to throw away their databases in order to switch. In addition, proprietary solutions have counter strategies in place to fight back, such as buying out competitors, buying a competitor would give you ownership of their intellectual property. Another strategy is a price reduction of propriety databases to attract newcomers. Finally, bait and switch strategy (Lazear 1995). By offering a cut-down version of the proprietary software for free targeting developers making it easy to develop with the specified software, then charge the consumer heavily when it is time to deploy (Packer 2007).

Bash on Windows 10

There have been a number of articles shining some light on the fact that Microsoft is starting to be interested in free open source software. According to Selvan “By incorporating bash in Windows, Microsoft has implicitly acknowledged the indispensability of the power of Linux-based tools, especially for the developer community.” It will be much easier to develop and join the open source community with this update to Windows. In turn, the number of contributors should see a major increase. Selvan writes “Windows still owns about 50 percent share of the desktop operating systems market,” this means individuals will be able to develop as easy as others who have Linux or Mac as their operating system. There will be no struggle in setting up virtual machines and environments for Windows users. In the article, the author discusses how Bash will be natively available as a full Ubuntu environment on Windows, without any virtualization or emulations. Microsoft is trying to bring the idea of free open source software to the widest audience possible. A new way for Windows developers to explore the Linux world. This should allow for more possibilities and capabilities that can be down through Microsoft Windows.

The current trends show a huge support of the open source software movement. It is evident through the climb of MySQL and MongoDB as they compete with their rival proprietary solutions. In addition, with the introduction of Bash on Microsoft Windows, the number of contributors to work on open source software projects should rise.

Final Remarks

Consumers' have a broad range of choice on what system to choose between free open source or proprietary software to run on their devices. There are advantages and disadvantages to choosing and open versus a proprietary solution. It all depends on a given company or firm's goals that they would like to achieve in order for them to be a successful business. The consumer has to make the decision of what they are willing to sacrifice and what they will be able to work with when choosing software. Free open source software is rising high amongst the old traditional commercial software, things may not be perfect now, but based on the last few years' trends, it appears that the market is headed that way.

References

262588213843476. 19:07:16 UTC. “Open Source License Usage on GitHub.com.” *GitHub*. <https://github.com/blog/1964-open-source-license-usage-on-github-com>.

“About MariaDB | MariaDB.” 2016. Accessed April 12. <https://mariadb.com/about>.

“Are Proprietary Databases Doomed? (Allan Packer’s Weblog).” 2016. Accessed April 10. https://blogs.oracle.com/allanp/entry/are_proprietary_databases_doomed.

Berrington, James. 2014. “Databases.” *Anaesthesia & Intensive Care Medicine*, Intensive care, 15 (2): 59–61. doi:10.1016/j.mpaic.2013.12.002.

“BSD License Definition.” 2016. Accessed April 12. <http://www.lininfo.org/bsdlicense.html>.

“DB-Engines Ranking - Popularity Ranking of Database Management Systems.” 2016. Accessed April 12. <http://db-engines.com/en/ranking>.

“DuckDuckGo Blog :” 2016. *DuckDuckGo Community Platform*. Accessed April 11. https://duck.co/blog/post/154/donations_2015.

“Gnu.org.” 2016. Accessed April 7. <http://www.gnu.org/licenses/gpl.html>.

Hawkins, Richard E. 2004. “The Economics of Open Source Software for a Competitive Firm: Why Give It Away for Free?” *Netnomics : Economic Research and Electronic Networking* 6 (2): 103–17. doi:<http://libaccess.senecacollege.ca:2131/10.1007/s11066-004-2717-z>.

“IBM developerWorks : Common Public License (CPL)-V1.0.” 2005. CT801. May 3. <http://www.ibm.com/developerworks/library/os-cpl.html>.

Lardinois, Frederic. 2016. “Microsoft Is Bringing the Bash Shell to Windows 10.” *TechCrunch*. Accessed April 11.

<http://social.techcrunch.com/2016/03/30/be-very-afraid-hell-has-frozen-over-bash-is-coming-to-windows-10/>.

Lazear, Edward P. 1995. "Bait and Switch." *Journal of Political Economy* 103 (4): 813–30.

Lerner, Josh, and Jean Tirole. 2002. "Some Simple Economics of Open Source." *The Journal of Industrial Economics* 50 (2): 197–234. doi:10.1111/1467-6451.00174.

"Licenses." 2016. Accessed April 7. <https://www.apache.org/licenses/>.

"MIT License." 2016. *Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=MIT_License&oldid=712472309.

"Mozilla Public Licence." 2016. *Mozilla*. Accessed April 7. <https://www.mozilla.org/en-US/MPL/>.

"Mozilla Public License." 2016. *Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Mozilla_Public_License&oldid=707049216.

Mustonen, Mikko. 2003. "Copyleft—the Economics of Linux and Other Open Source Software." *Information Economics and Policy* 15 (1): 99–121. doi:10.1016/S0167-6245(02)00090-2.

"MySQL :: About MySQL." 2016. Accessed April 12. <http://www.mysql.com/about/>.

"NoSQL Databases Explained." 2016. *MongoDB*. Accessed April 12. <https://www.mongodb.com/nosql-explained>.

"Open-Source Software." 2016. *Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Open-source_software&oldid=714772389.

"Open Source wp2.pdf." 2016. Accessed April 11. <http://www.astera.com/media/66282/open%20source%20wp2.pdf>.

Popp, Karl. 2011. *Advances in Software Economics*. BoD – Books on Demand.

"PostgreSQL: About." 2016. Accessed April 12. <http://www.postgresql.org/about/>.

- Riehle, D. 2010. "The Economic Case for Open Source Foundations." *Computer* 43 (1): 86–90. doi:10.1109/MC.2010.24.
- Samuelson, Pamela. 2012. "Is Open-Source Software the Answer? The Comingled Code: Open Source and Economic Development." *Issues in Science and Technology* 28 (3): 92+.
- Selvan, Raghavendra. 2016. "A Window into Free and Open-Source Software." *The Hindu*, April 7.
<http://www.thehindu.com/opinion/op-ed/a-window-into-free-and-opensource-software/article8442499.ece>.
- Sen, Ravi, Siddhartha S. Singh, and Sharad Borle. 2012. "Open Source Software Success: Measures and Analysis." *Decision Support Systems* 52 (2): 364–72. doi:10.1016/j.dss.2011.09.003.
- Stallman, Richard. 2002. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. 14. Lulu.com.
- "The MIT License (MIT) | Open Source Initiative." 2016. Accessed April 7. <https://opensource.org/licenses/MIT>.
- "What Is LAMP (Linux, Apache, MySQL, PHP)? - Definition from WhatIs.com." 2016. *SearchEnterpriseLinux*. Accessed April 12.
<http://searchenterpriselinux.techtarget.com/definition/LAMP>.