

Impact of open source software on software industry

Tapan Kumar Sahu¹

¹(CS, Berhampur University, India)

Abstract:- The emergence of open source software (OSS) has triggered several changes in the software industry that has long been dominated by proprietary softwares. OSS has been cited as an archetypal form of open innovation; it consists of the convergence and collaboration of like-minded parties. An increasing number of software firms have taken upon this approach to link outsiders into their service development and product design. Also, software firms have been increasingly grounded their business models on user-centric and service-oriented operations. OSS has also provided the user a number of options to choose for using the software like to install software on premise, deploy it in the cloud, or use it in a Software-as-a-Service (SaaS) model This discussion relates to the changes brought about by open source softwares(OSS) to the IT industry over the last 20 years.

Keywords :- OSS,SAAS,proprietary software,closed source,opensource

I. INTRODUCTION

Open source software is software whose source code is available for modification or enhancement by anyone. Source code is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software application works. Some software has source code that cannot be modified by anyone but the person, team, or organization who created it and maintains exclusive control over it. This kind of software is frequently called "proprietary software" or "closed source" software. OSS has become widely popular in the last decade because programmers have more control over that kind of software. They can examine the code to make sure it's not doing anything they don't want it to do, and they can change parts of it they don't like. Users who aren't programmers also benefit from open source software, because they can use this software for any purpose they wish. In this discussion, the focus will be on the change brought out by OSS to the software industry that has been dominated by proprietary software since a long time.

II. OSS-A RELIABLE OPTION TO PROPRIETARY SOFTWARE

Looking back at the last 20 years, today an individual or a company has a choice between proprietary software and open source software. OSS gives so many options for the way in which the software can be use like free software, software with annual subscription licenses, etc. so that the way we consume software has also changed. One can choose to install software on premise, deploy it in the cloud, or use it in a Software-as-a-Service (SaaS) model. What one ends up with is more combinations available to the software buyer, and open source is one of the driving forces behind that change. Before open source, all one could do was buy licenses of proprietary software with limited options with software. Open source's price tag is clearly an important driver, but it's not the key advantage of open source. While cost is definitely a factor – especially in recent, tough economic times – the first and foremost reason why people are choosing open source is because it's flexible – and reliable – enough to meet their needs. IT organizations are increasingly valuing the fact that they can adopt open source technologies on their own terms, at their own pace and within their most mission-critical environments. They can customize the software and choose a pricing model that's much more flexible than what they would have gotten with proprietary software. Open source has been one of the most significant cultural developments in IT and beyond over the last two decades, and has shown that individuals, working together over the Internet, can create products that rival and sometimes beat those of giant corporations.

III. BENEFITS OF OSS

3.1 Reliability

With OSS, severe defects tend to be fixed within hours of their being detected, a process which is undoubtedly assisted by the availability of the source code. Able developers who discover a bug will commonly also fix it and then report it to the maintainers as well as issuing an updated version of the software on their own authority. Users of the software can choose whether to use the unofficial fix or wait for an 'official' version. By 'official' we mean a release blessed by the project team itself or a trusted authority such as one of the main distributors of Open Source packages. The pattern with closed-source software is typically that a defect report needs to be filed and then there will be a delay before the vendor determines when or whether to issue an updated release. Users of the software are much more at the mercy of the vendor's internal processes than with the Open Source arrangement and the personal experience of the authors is that it can be extremely frustrating to move from the Open Source to the closed model.

3.2 Stability

Software vendors can apply a number of tactics to persuade their customers to upgrade more or less willingly. Typical tactics include moving to allegedly new and improved file formats (which require the new and improved software to read them) or to withdraw support and bug fixes for older versions after a short period. The problem for users of the software is that they rarely have much control over that process and are left isolated if they choose to remain with older versions that they consider to be acceptable. This has cost and control implications for the business.

With OSS, the worst effects of vendor-push can be mitigated. The way that Open Source products tend to conform closely to standards efforts has an inertial effect, since standards change but slowly and interchange formats are often particularly stable. As a result, incompatible file formats can be less of an issue. If they are standards-based then they typically aren't an issue at all, and if they are formats unique to the software product — proprietary formats in a sense - then they cannot be undocumented since the source code that uses them is itself published. It is commonplace for a Perl or similar converter program to be shipped with them which will upgrade data to the new format.

3.3 Auditability

A rarely-understood benefit of OSS is its auditability. Closed-source software forces its users to trust the vendor when claims are made for qualities such as security, freedom from backdoors, adherence to standards and flexibility in the face of future changes. If the source code is not available those claims remain simply claims.

By publishing the source code, authors make it possible for users of the software to have confidence that there is a basis for those claims. Whether this takes the form of an cursory and informal inspection or more rigorous auditing, what's clear is that without access to the source, third party inspection is impossible. At present the industry does not insist on third party inspection or certification, but it's possible that as open source models become more popular then expectations of audits will rise.

3.4 Cost

From a business perspective the purchase cost of software is only one factor; total cost of ownership (TCO) is what really matters. Other things being equal, the solution with lowest TCO is usually the most desirable one. Arguments in favour of low TCO for open source software include:

- Possibly zero purchase price
- Potentially no need to account for copies in use, reducing administrative overhead
- Claimed reduced need for regular upgrades (giving lower/nil upgrade fees, lower management costs)
- Claimed longer uptimes and reduced need for expensive systems administrators

- Near-zero vulnerability to viruses eliminating need for virus checking, data loss and downtime
- Claimed lower vulnerability to security breaches and hack attacks reducing systems administration load
- Claimed ability to prolong life of older hardware while retaining performance

3.5 Flexibility

What happens most often is that a vendor will make a 'feature sale', emphasising something which cannot be done through the standard infrastructure. If they succeed then the business can become dependent on that particular solution and unable to choose alternatives at a later date. Any astute vendor will attempt to do this, only vigilant managers can avoid the lock-in that follows. Open Source projects have very little motivation to attempt this kind of lock-in strategy. Since there is no commercial benefit to be had, adherence to de-jure or de-facto standards (where they exist) is typically high. Where standards for interworking do not exist, the fact that the source code is published means that proprietary data formats can't be used to manipulate lock-in. This explains the relative success of Open Source software in infrastructure areas

3.6 Support

In most situations, open-source consultants provide training and/or support for software they recommend. However, the fundamental advantage of open-source software when it comes to support is that it's always possible to retain a company to provide support. Because the source code is freely available, organisations are not limited to obtaining support from the authors. There is no restriction on other suppliers learning enough about the software to provide adequate support whenever demand exists.

IV.APPLICATIONS OF OSS

4.1 OSS in Cloud Computing

OSS solves the main concern about the cloud -- user control of software and data to avoid extreme lock-in .open source provides open access to application programming interfaces (APIs) and the open standards they are written against. Typically, open source projects are innovative, with aggressive release cycles that push the technology forward. In fact, users often determine the next feature release cycle based on real-world business needs. The following are the various Open source Cloud computing softwares:

Distributed Computing : Hadoop,Cassandra

- Infrastructure as a Service(IaaS): OpenStack, CloudStack, Eucalyptus, OpenNebula
- Software as a Service (SaaS):OrangeHRM,vTiger

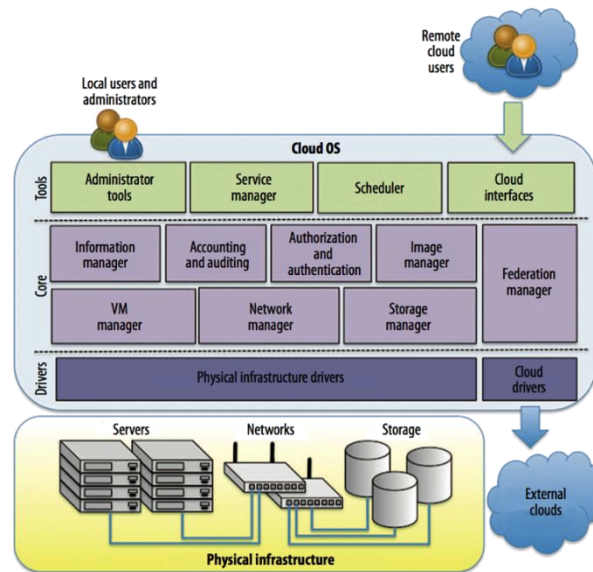


Fig 1. Open source cloud IAAS architecture

4.2 OSS in Application Development

Originally popularized from the phrase "Linux, Apache, MySQL, and PHP", the acronym "LAMP" now refers to a generic software stack model. This particular software combination has become popular because it is entirely free and open-source software. This means that each component can be theoretically interchanged and adapted without overt [vendor lock-in](#), and that the complete software stack is optionally available free of cost. Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution. The Apache HTTP Server has been the most popular web server on the public Internet. MySQL is a multithreaded, multi-user, SQL database management system (DBMS). PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code is interpreted by a web server via a PHP processor module, which generates the resulting web page

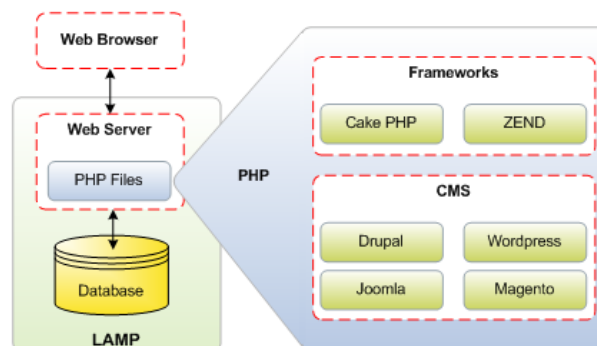


Fig 2. LAMP architecture

V.FUTURE OF OSS

The open source community will continue to challenge the closed source one as far as quality based service is concerned. As industries and technologies continue to evolve at a faster pace, closed source companies will be dead in the water if they aren't leveraging the open source community and the projects they're working on. When it comes to software development, it will be increasingly difficult to integrate, develop, and extend closed source software--and extremely costly compared to open source alternatives. Twitter's messaging app is a good example of how open source can be powerfully used, and of how well-known companies are influencing the community. Companies have come to understand the importance of open source and the key role it plays in developing innovative software today.

VI. CONCLUSION

OSS-based software development urges software developers to implement development processes in a way that calls increasing attention to user involvement throughout the software development and delivery processes. Also OSS-based software development change the focus of competition in the software business from product-centric to service-centric operations.

REFERENCES

- [1] OPEN SOURCE: A MULTIDISCIPLINARY APPROACH (SERIES ON TECHNOLOGY MANAGEMENT) BY MORENO MUFFATTO
- [2] CLOUD COMPUTING BLACK BO*OK PAPERBACK – 16 JUL 2014 BY KAILASH JAYASWAL, JAGANNATH KALLAKURCHI , DONALD J. HOUDE , DEVEN SHAH
- [3] PROFESSIONAL LAMP: LINUX, APACHE, MYSQL AND PHP5 WEB DEVELOPMENT PAPERBACK – DECEMBER 5, 2005