



White paper:

THE HIGH COST OF POOR PERFORMANCE

High cost & unseen risks of unsupported
open source commercial deployment

High cost & unseen risks of unsupported open source commercial deployment

Introduction

THE FOUNDATION OF THE INTERNET as we know it is built on open source software. Its infrastructure largely depends on open source and open standards. The most popular content management systems, including WordPress, Drupal and Joomla, are open source; one of the most widely used databases, MySQL, is open source, and according to Netcraft, 85 percent of all websites are hosted on open source web servers¹. Additionally, we know that Varnish Cache is used by more than two and a half million websites worldwide, among them some of the world's biggest, including Pinterest and the New York Times. Some of technology's stalwarts, such as Google, Amazon, Facebook and Twitter, were built on open source platforms but have a proprietary layer on top of the open source foundation. Apple, too, bases many of its platforms and products on open source² (the history of WebKit can tell its varied tale³).

What you will learn:

- How to calculate the true cost of downtime and performance loss
- Why it's not just technological risks you should think about, it's reputation too
- How supported solutions can have lower overall deployment costs
- The critical questions to ask yourself when evaluating unsupported open source solutions

Looking at business and organizations today - and at changing business models and modes of doing business as well as means of organizing and delivering information - we know that things are in a constant state of flux. The technology underpinning these workings needs to be flexible, future-proof, scalable and easy to integrate. Users expect everything with which they interact on the internet to be instant, reliable and foolproof. When users' expectations are not met, they will go elsewhere. In virtually any situation - from an e-commerce platform satisfying all its customers with the most up-to-date product and inventory information and seamless purchases, to a media company ensuring smooth streaming video and the most up-to-date news scaling to meet traffic peaks, to a public sector organization ensuring access to critical (tax, healthcare) or emergency (crisis) information - an organization lives or dies by how well and reliably it delivers.

Varnish Cache is used by more than two and a half million websites worldwide, among them some of the world's biggest

1. <http://news.netcraft.com/archives/category/web-server-survey/>
2. Economic Impact of Open Source on Small Business: A Case Study By Mike Hendrickson, Roger Magoulas, Tim O'Reilly, July 2012
3. <https://webkit.org/>

Poor performance and why it's important

The most successful companies with internet-driven business processes keep their eyes and KPIs on one common attribute: performance. What does performance entail in this context? Speed, reliability and scalability.

Need for speed

As long ago as 2012 (an eternity in the digital space), a New York Times article quoted Harry Shum, a Microsoft computer scientist: **“users will visit a website less if its loading time is slower than its competitors by 250 milliseconds, or one-quarter of a second.”** That is less time than a single eye blink⁴. Performance has only continued to grow in importance. Faster web performance leads to higher search ranking⁵, better user follow-through/conversions and engagement (staying on-site, making purchases, etc.), which leads to a higher return on investment and overall better user experiences⁶.

According to research from Aberdeen Group, one second delay in page response will result in 11 percent fewer page views, a 16 percent decrease in customer satisfaction and seven percent reduction in online sales⁷. In monetary terms this means that if your site earns somewhere around USD 85,000 every day, you could lose more than USD two million annually. The same statistics on one-second page load are borne out in other research/surveys from Akamai and Gomez.com⁸.

Users have more options than ever and have no problem abandoning a site that does not perform to their expectations and perceptions.

Meanwhile, the number of websites proliferates, the size of websites and pages balloons, as more users attempt to access these pages on more and more devices. Speed and immediacy are increasingly important, but potential traffic jams threaten to overload servers. The server load is the primary roadblock to achieving high performance. Yet, this is one problem that can easily be addressed with caching technology.

Reliability: Can you count on it?

For business-critical applications, it is not just speed that makes up performance, even if speed is the most obvious metric. Reliability and availability are fundamental. What difference does it make, after all, how fast your pages load if the entire site crashes when traffic peaks? Businesses relying on seamless performance and functionality - particularly in high-traffic periods - have no choice but to ensure reliability and uptime, insofar as possible. An e-commerce site may have a planned sale, a travel site is running a promotion, a media outlet knows a big event is coming (e.g., the Olympics): in these predictable cases, there is no excuse for not preparing. But even in unplanned peaks, for example when a major news event floods a news site, a business-critical application should be protected and ready for anything - all the time. Similarly, cyberattacks/DDoS attacks can cause downtime (a 2014 Kaspersky study estimates that a DDoS attack can cost a company anywhere from USD 52,000 to USD 444,000⁹) - even if a company cannot predict when or how this will happen, it can have a contingency plan in place if it does.

Can you come up with a basic calculation of what downtime will cost you¹⁰? One simple equation to get a rough estimate of lost revenue from downtime: $(GR/TH) * I * H^{22}$.

To put this into context, let's look at the effect of a four-hour downtime for a small US company with a four million USD revenue: $\$4,000,000/2000*10*4=80,000$. In just four hours of downtime, a company with a relatively small GR would lose USD 80,000, or two percent of their gross yearly revenue.

$$\text{LOST REVENUE} = (\text{GR}/\text{TH}) * I * H$$

GR: gross yearly revenue

TH: total yearly hours

4. <http://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html>

5. <https://searchenginewatch.com/sew/how-to/2409674/every-second-counts-why-page-speed-should-be-your-next-focus>

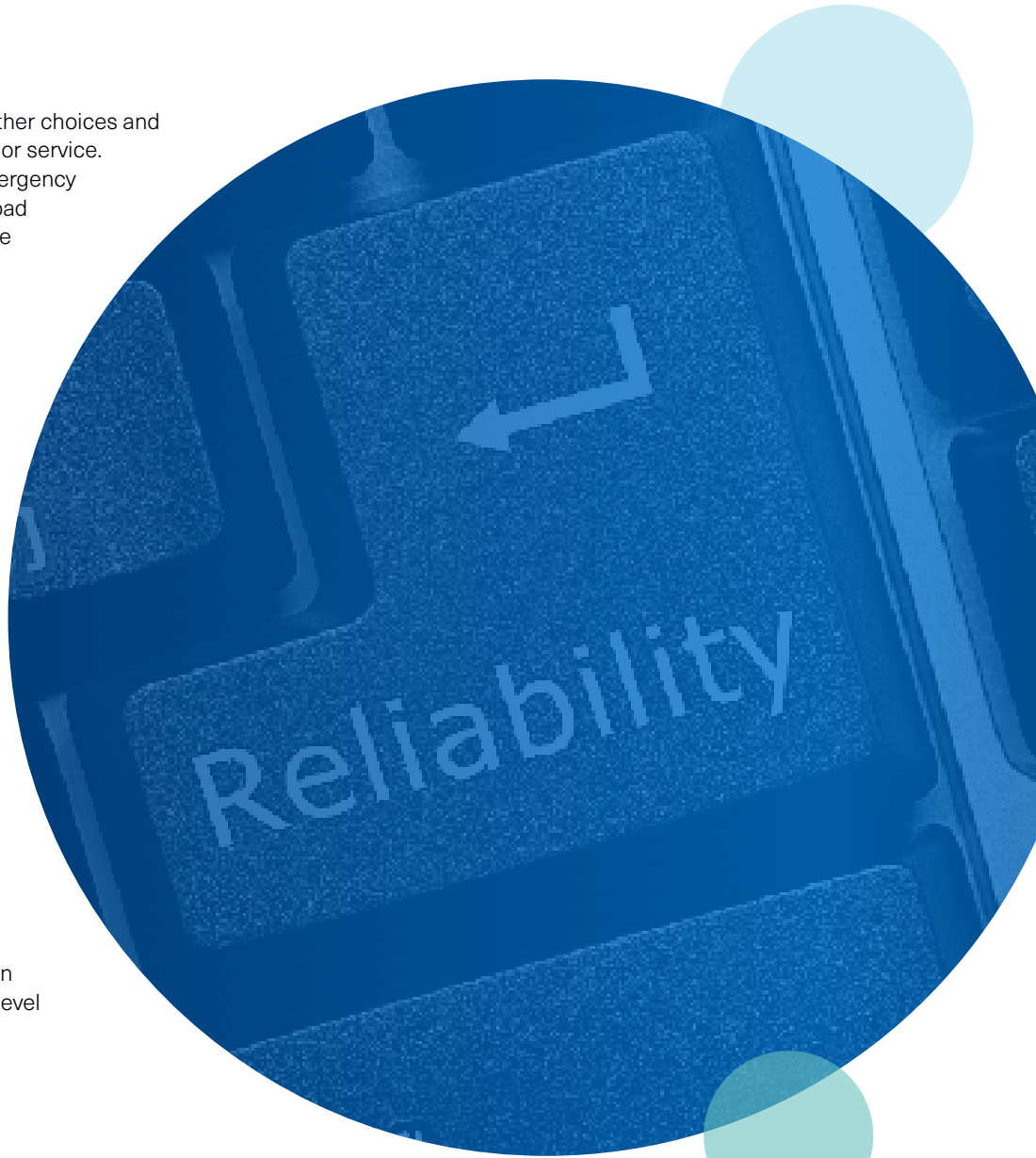
6. <https://www.linkedin.com/pulse/20140516013608-1981105-the-importance-of-website-loading-speed-top-3-factors-that-limit-website-speed>

7. <http://www.aberdeen.com/Aberdeen-Library/5136/RA-performance-web-application.aspx>

8. <https://blog.kissmetrics.com/loading-time/>

In some cases, users will have other choices and will use another site, application or service. In other cases, such as in an emergency or disaster situation, a site overload and crash means that citizens are left without crucial information that may even pertain to their safety. In either case, the costs are too high not to have high availability in place to at least ensure users can access the last-known working version of the application and its information.

As far back as 2009, companies indicated in an Information Technology Intelligence Corp (ITIC) survey that they knew availability and reliability were cornerstones of performance and planned to invest in high-availability clusters. One out of ten companies at the time said they required greater than 99.999% availability¹¹. The need for high availability has not decreased - and more than one in ten companies may require this level of availability now.



Will it scale?

Related to reliability - that is, will your website or application continue to stay up and running even in peak conditions - scalability is a crucial question. The same conditions apply, e.g. preparing to scale for whatever level of traffic comes to your site. The expensive way would be to invest in more and more hardware, but this is unsustainable, and even then, the backend could crash because unpredictable traffic levels can overwhelm the servers. The easier and cheaper way is with caching - requests hit the backend only once, which should let a virtually unlimited number of users trying to access the same content the ability to access it at the same time. The sheer volume of traffic should not crash your site.

Users have more options than ever and have no problem abandoning a site that does not perform to their expectations

9. <https://press.kaspersky.com/files/2014/11/B2B-International-2014-Survey-DDoS-Summary-Report.pdf>
10. <http://www.techrepublic.com/article/how-to-calculate-and-convey-the-true-cost-of-downtime/>
11. <http://searchcio.techtarget.com/podcast/Trends-in-high-availability-and-fault-tolerance>

What effects does bad performance have?

The effects of poor performance can vary widely and have quite a broad scope. The major effects include financial/bottom-line losses and consequences, the need for infrastructural change or investment and exposure to greater risk (legally and in terms of brand and reputation).

Bottom-line losses: Revenue and downtime

Earlier cited Aberdeen figures estimate a seven percent reduction in online sales with each second of page-load delay⁸. It is logical that poor performance, whether in the form of a total outage (which obviously leads customers to go elsewhere) or in the form of sluggish page loads or even a **perception of slowness**, will affect direct revenue/sales.

Downtime is another story, and of course varies by industry and scope and length of an outage. On the whole, downtime is expensive and may have lasting consequences.

A 2015 IDC/AppDynamics report cited figures that should alarm any company that relies on its applications:

- For the Fortune 1000, the average total cost of unplanned application downtime per year is USD 1.25 billion to USD 2.5 billion.
- The average hourly cost of an infrastructure failure is USD 100,000 per hour.
- The average cost of a critical application failure per hour is USD 500,000 to USD 1 million¹².

Numerous reports in recent years have attempted to assign a dollar value to downtime/outages based on a number of different factors¹³.

- Gartner, July 2014¹⁴: USD 336k/hr (with variance as high as USD 540k/hr)
- Ponemon Institute, December 2013¹⁵: USD 474k/hr*
- IDC, November-December 2014¹⁶: USD 100k/hr or USD 500k+/hr for critical failure
- Infonetics, February 2015¹⁷: up to USD 100 million/yr

*37% of this cost is due to reputation damages and customer churn

The effects of poor performance can vary widely and have quite a broad scope.

This is certainly not a new problem - in fact, it's one that continues to grow. A real-world example is a well-publicized 2013 Amazon Web Services outage that lasted for about an hour, and affected a number of AWS customers, but also the Amazon.com e-commerce platform itself, which may have led to a loss of up to USD 4 million in revenue¹⁸.

A 2011 study sponsored by CA Technologies found that companies lost more than 127 million work hours annually because of IT downtime¹⁹. Translating this into salary payments (when employees were working at an estimated 63% of their productivity level), downtime becomes a serious financial issue beyond just the direct losses in internet sales/revenue-earning activities (up to USD 138,000 per hour of downtime).

Let's just say that neglecting performance can be an expensive proposition.

Lack of stability = need for investment

For businesses that rely on (or exist solely on) the web, stability is a key to both technical and financial stability. Poor performance ends up leading not only to the damages described in this section of this paper, but also to the need for investment to attain performance improvements and stability. In the long run, investing in better infrastructure or architecture to achieve better performance is positive - but not if it's done too late or the wrong solutions are brought in. After troubleshooting initial performance problems, finding and implementing new solutions takes time and resources.

PERFORMANCE:

GOOD

BAD

12. <http://devops.com/2015/02/11/real-cost-downtime/>

13. <http://www.statuscast.com/application-downtime-according-to-idc-gartner-and-others/>

14. <http://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/>

15. <http://www.emersonnetworkpower.com/en-US/About/NewsRoom/NewsReleases/Pages/Emerson-Ponemon-Cost-Unplanned-Data-Center-Outages.aspx>

16. <http://info.appdynamics.com/rs/appdynamics/images/DevOps-metrics-Fortune1K.pdf>

17. <http://www.infonetics.com/pr/2014/Cost-Server-Application-Network-Downtime-Survey-Highlights.asp>

18. <https://www.buzzfeed.com/mattynley/the-high-cost-of-an-amazon-outage>

19. <http://www.ca.com/us/news/press-releases/na/2011/ca-technologies-survey-reveals-it-systems-failures-cost-businesses-127-million-lost-person.aspx>



RISK MANAGEMENT

Risk: Loss of trust and damaged reputation

Performance failures and problems may expose a company to risk on dual fronts.

The first, of course, is legal, if, for example, the performance failure violated an SLA. Even the threat of legal action has far-reaching implications, and actual legal actions will be costly, time consuming and damage a company's credibility and reputation. Legal considerations are generally well-understood in business, but it is important that businesses factor the legal ramifications, responsibilities and consequences of technology failure into their understanding of liability.

Perhaps somewhat less well-considered is the potentially far-reaching damage a business can suffer to its trustworthiness and reputation when performance is compromised or fails altogether. The aforementioned CA Technologies survey concluded that 50% of businesses felt that outages significantly damaged their reputation and brand⁹. While it can be difficult to quantify exactly what reputation and brand damage costs, there are estimates. The Ponemon Institute's 2014 report calculated that an outage cost an average of USD 474k/hour, of which 37% could be attributed to damaged reputation and customer churn¹².

This might be too simplistic and short-sighted a view on the possible cost and long-term damages. The effects are not limited to the immediate costs of lost business and employee time spent fixing the problem(s). There are also less tangible and less calculable damages: value of lost data and time, lost trust in the system(s) being used, damage to employee morale/trust - and then the even less predictable after-effects of brand/reputation damage. The costs here cannot be ignored, even if they cannot always be fully grasped. They include the aforementioned losses (including the possible lifetime value of a customer who leaves and does not return). But they also include the ripple effects of negative word of mouth, negative press coverage and the PR/marketing efforts/cost a company will have to invest to generate a turnaround. There are also potential financial effects, such as stock downturns, that are not necessarily immediately felt but which can lead to massive business shakeups, further eroding confidence in a company and possibly leading to legal and compensatory action²⁰, as a 2011 case worth AUD 20 million between Virgin Blue in Australia and its online reservation management company, Navitaire, illustrates.

Beyond risk: Unsupported open source - Arguments and counterarguments

Poor performance leads to potentially dire - and very real business consequences.

Knowing the risks and costs, there are still companies who will argue that unsupported open source solutions are good enough. We won't argue on half that point: open source is great. We happen to believe, though, that it's even better when it's offered with full support and expertise to ensure that companies do not run into the performance problems, and related costs highlighted above - and that they get the most from the software itself, which may help make significant contributions to their business. Each of the most frequent arguments about open source can easily be countered with a potential risk.

Legal considerations are generally well-understood in business, but it is important that businesses factor the legal ramifications, responsibilities and consequences of technology failure into their understanding of liability.

20. <http://www.theaustralian.com.au/business/navitaire-booking-glitch-earns-virgin-20m-in-compo/story-e6frg8zx-1226033624246>

Poor performance leads to potentially dire - and very real - business consequences.

Knowing the risks and costs, there are still companies who will argue that unsupported open source solutions are good enough. We won't argue on half that point: open source is great. We happen to believe, though, that it's even better when it's offered with full support and expertise to ensure that companies do not run into the performance problems, and related costs highlighted above - and that they get the most from the software itself, which may help make significant contributions to their business. Each of the most frequent arguments about open source can easily be countered with a potential risk.

Open source is (cost-)free

While self-development is perceived as cost-effective and efficient (because open source is ostensibly free of cost), the level of competence required to implement, configure, monitor and maintain involves exceptional capital and resource investment with no guaranteed or foreseeable quality outcome.

A 2011 London School of Economics paper looked at the total cost of ownership, and indeed found that, while many drivers prompted the adoption of open source software, the number one concern was the basic, upfront cost of the software itself (as well as the easiest to estimate). Much harder to estimate were the costs of maintenance, upgrades and support as well as major organizational change, exit costs or setting up in-house services²¹. Nevertheless, these drivers are important considerations for anyone looking at open source solutions and what kind of support is needed. Key question: It's free on the surface – but does your company want to invest in becoming an expert in, for example, Varnish and in caching structures and strategies overall, to the point that you commit resources going forward? Adopting open source without support requires developing a certain level of expertise in all the specific layers you put into the stack.

80% of outages due to lack of expertise

This is one of the major issues: configuration and implementation. Once you've put multiple applications into one physical server, one improperly configured application can also affect other applications¹⁰.

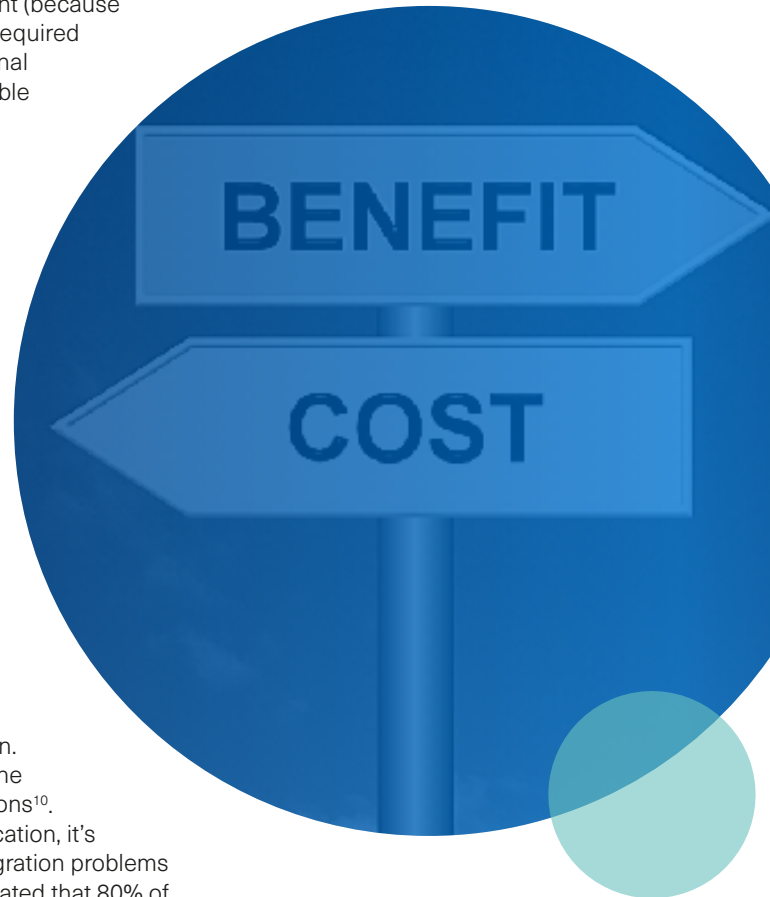
If developers working on this stack are not experts in each application, it's almost a foregone conclusion that configuration, change or integration problems are inevitable: IT Process Institute's Visible Ops Handbook estimated that 80% of unplanned outages are due to ill-planned changes by administrators or developers²². A Gartner

study backed up these numbers: **"Through 2015, 80% of outages impacting mission-critical services will be caused by people and process issues, and more than 50% of those outages will be caused by change/configuration/release integration and hand-off issues."**²³ These kinds of manual errors and issues can cost companies up to USD 72,000 per hour in downtime²¹.

What is the true cost in monetary and resource terms? That is, what is the **total cost of ownership** and how is that managed and paid for throughout the lifecycle of the software? Can you afford **even one** of these kinds of configuration errors in your open source commercial deployment?

Open source is freedom

The aforementioned LSE paper posits: "the softer benefits of open source adoption are also widely appreciated – those of flexibility, openness, ability to tweak and customize, and support for open standards and open data."²²



The effects of poor performance can vary widely and have quite a broad scope.

21. [http://eprints.lse.ac.uk/39826/1/Total_cost_of_ownership_of_open_source_software_\(LSERO\).pdf](http://eprints.lse.ac.uk/39826/1/Total_cost_of_ownership_of_open_source_software_(LSERO).pdf)
22. <http://www.evolve.com/blog/downtime-outages-and-failures-understanding-their-true-costs.html>
23. https://img2.insight.com/graphics/no/info2/insight_art6.pdf

At least nominally, open source offers the promise of these freedoms: no barriers to getting started, technical and creative control (for example, with Varnish Cache you have the Varnish Configuration Language (VCL)) and community resources at your disposal. Flexibility and customizability hold great appeal - but also require a deeper level of expertise. (Will you have that expertise in-house/on-staff?). Open source may create the perception that a use case will be feasible and have longevity - and happen quickly. In fact, these outcomes depend on what you put into them. You've undoubtedly heard analogies: buying an already-built house (commercial software) or buying the tools, materials and blueprints (open source)²⁴. Your need will depend on what you want and how much risk and time you are willing to assume and invest.

Open source is also attractive because it helps companies avoid "vendor lock-in"²⁵, which has ranked consistently as the number one and two reasons companies opt for open source, according to Future of Open Source surveys²⁵. Proprietary solutions from vendors that ultimately cannot integrate or be compatible with other solutions create both expense and hassle. The constant need to and expense of upgrading commercial software (and possibly disrupting day-to-day operations) also tempts companies to look at open source²⁶. The industry as a whole, particularly in open source and open standards development, has aimed to work from a common set of standards and protocols that can work with, be built on or replaced regardless of vendor. Sticking with open source has generally been an antidote for this kind of vendor dependence.

But even with open source, strings can be attached – adopt the open source solution without support, and the knowledge within your internal teams suddenly becomes not just extremely valuable but make-or-break to your business. But what if the internal Varnish expert walks out the door? What if you don't have a bona fide Varnish expert on staff, and many different people know different things, and you lose some of them – or no one has an ownership stake in Varnish Cache and never updates to newer versions? You are vulnerable on many fronts in this scenario: your in-house expertise and knowledge can leave or be fragmented; your implementation of Varnish Cache might be stuck on an old version for which there is no longer community support, which could lead to performance or legal problems²⁷.

Are you really getting total freedom and flexibility by going "free"?

Open source is "done"

In many cases, the perception exists that all the work is done already with an open source solution, but quite often it's just a foundation to build on. When customization and bespoke configuration come into play, it can become more complex. No one will argue that Varnish Cache is not complete – you can adopt and use it without a lot of development work. However, if your Varnish Cache implementations are business-critical and you rely on the software to achieve high-performance content delivery and want key modules to "out of the box", this is where competence and development effort come into play, and where access to Varnish Software's engineers will save time, money and effort. You care about your performance – and that involves a lot of moving parts. That means Varnish Cache is not necessarily your priority – but it is ours.

Do you have the internal resources to build and develop the extras you need to take full advantage of your caching strategies and to deliver content at scale? Particularly in leaner economic times when cutting back on staff or in hiring freeze conditions, this should be a top consideration.

24. <http://www.forbes.com/sites/rajsabhlok/2013/07/18/open-source-software-the-hidden-cost-of-free/>

25. http://www.slideshare.net/North_Bridge/2015-future-of-open-source-study

26. <http://www.computerweekly.com/feature/Finding-out-the-hidden-costs-of-open-source>

27. <https://opensource.com/law/14/12/gplv2-court-decisions-versata>

28. <https://medium.com/@nayafia/how-i-stumbled-upon-the-internet-s-biggest-blind-spot-b9aa23618c58#>

Open source is well-funded and/or supported and updated by the community

Theoretically, yes, open source communities build, use and contribute back to the community they drew from. But it is becoming increasingly common that more people and entities use open source ("free rider problem") but do not contribute back to the source²⁸. You do not always know what you are going to get. Thus, without supported software (open source or otherwise), you may escape vendor lock-in, but your trade-off is that you almost become your own software vendor: "It's up to you to provide ongoing maintenance, upgrades and troubleshooting, as well as any needed end-user support. Congratulations! You're now a software vendor. The high switching costs of commercial apps are now replaced by the high costs of supporting open source apps.^{25"} This does not have to be the case, but very often is.

Are you really getting total freedom and flexibility by going "free"?

Estimating the risk and implications of unsupported open source projects

Before you can accurately evaluate the risk of introducing an unsupported open source solution to your infrastructure, it's important to look carefully at what is at stake and ask yourself critical questions.

Some big picture considerations:

- How important is performance and scalability to your business? Ignoring, underestimating or neglecting performance as key to your web and overall business strategy can be perilous.
- What are your legal responsibilities/liabilities and potential repercussions? This is becoming a more common and tangled issue all the time²⁹.

More specific questions for evaluating the risk of developing on your own:

- How much time will you need to dedicate to development for deployment?
- Is there considerable development/integration/configuration work involved? How much time will that take? How many staff will be involved and how much do they cost?
- How much time and money do you lose if the specialized internal staff leave with all their accumulated knowledge?
- How much will downtime cost will you face if your internal staff cannot easily solve a problem/fix an outage?
- How much will you lose if your internal staff fail to keep Varnish and other auxiliary service software updated? (Performance, failure, security?)

The advantage of Varnish

Inexpensive insurance – proven support

Why is a supported subscription the right choice? The previously cited LSE paper explains: "Open source adopters do note that they needed to hire experts and look for support to meet their organisation's ambitions including for control of code and configuration"²².

Even if you were to compare the cost of one internal employee against the cost of a Varnish Software support subscription, you're way ahead of the game. A subscription costs considerably less than the salary of a dedicated employee, and you get the guarantee of continuity and no loss of institutional memory if your dedicated employee decides to leave.

Varnish Software customer, sheego.de, a German retail e-commerce platform, turned to Varnish Cache when they moved to a new e-commerce platform that did not have a built-in caching component. Without in-depth caching expertise in-house, sheego wanted to ensure the stability, reliability and performance of their shop and subscribed to Varnish. The 24/7 professional support gave them the expertise they needed, even in the middle of the night.

When we went live **with Varnish**, the professional support from Varnish Software was excellent. I can really recommend it. It was there when we needed it. Our shop is the center of our marketing and business. We go live with our marketing activities during the night to ensure minimal disruption to business, and the support from the Varnish team was there when we needed it, even in the middle of the night."

– **Steven Dagadu**,
vsheego.de Head of
E-Commerce Development.

Hard-won expertise through development and testing - technical innovation and user experience

The engineers who provide support to Varnish subscribers are the same core engineers who have built Varnish Cache. They know it inside and out. It's their experience and expertise you get when you buy a support subscription. It is also this "foundation of invention" that helps our customers to develop solutions that enable them to deliver exceptional user experiences, unique tools and implementations and take full advantage of the flexibility Varnish offers.

Varnish customer, CacheFly, a CDN network built specifically for high throughput, experienced rapid growth that required a caching solution that supported scalability at CacheFly's performance level. CacheFly discovered that they could deliver massive amounts of content with no performance disruption, enjoy more granular control over caching, drastically improve their cache hit rate and deploy new features quickly with the flexibility of VCL. Essentially, with Varnish engineering help and Varnish, CacheFly was able to use Varnish in a new way to develop exactly what they needed.

Varnish development is driven by technical innovation, looking toward technology trends and what people are doing on and with the web and how they are using it. We constantly explore how Varnish contributes to customers/community success and by extension provides better user experiences, whether those experiences are within the media industry, the financial service industries, the healthcare industry, in e-commerce or any number of other online/digital aspects of business.

Savings

When Varnish experts help to optimize your Varnish instances, you often gain further efficiencies and thus can reduce investment in infrastructure/servers. That's not to say that your in-house experts cannot find ways to achieve this, but Varnish experts work with Varnish every day and can identify efficiencies that can pay dividends in terms of performance and scalability - all while working with the infrastructure you already have (or even reducing it).

Varnish Software customer, SFR, a French telecommunications giant providing voice, video, data and internet services, not only used Varnish to create its own CDN network to reduce costs - they used their new Varnish-powered DIY in-house CDN solutions as a new revenue stream, selling the CDN services on to their customers. With Varnish, SFR improved their online store response time while also eliminating the costs associated with their Akamai CDNs. As a result, SFR ended up providing a full-featured, best-performing CDN to external clients, such as leading e-commerce sites, video providers and TV channels. SFR's DIY CDN solution has not only delivered ROI for themselves but also, by extension, to many of their customers.

Using Varnish we are able to deliver hundreds of terabytes of content seamlessly as well as deliver on our 100 percent SLA to our customers,

-- *Matt Levine,*
CacheFly founder and CTO.



Conclusion

We love open source, obviously. The powerful core of the Varnish Plus solution suite is the open source Varnish Cache. Varnish Cache is a fantastic, flexible piece of software, and we are not shy about saying so. We know that Varnish Cache, like many other open source platforms and tools, delivers.

But we also recognize that introducing and maintaining an open source solution into a high-stakes, business-critical commercial environment comes with potential risks and costs, which can be more than offset by opting for a commercial subscription and support from the makers of an open source product.

Bottom line: there is no such thing as completely cost-free software development³⁰. The upfront “free” cost of open source software is rarely the steal you imagine it to be. A supported version of open source software, though, can bridge the gap: lower costs than fully proprietary commercial solutions without the vendor lock-in or lack of flexibility such solutions often mean.

You can develop your solutions yourself using unsupported open source software, but it can be both costly and risky. In commercial deployment, though, it is well worth considering a supported solution.

With Varnish Cache, you’re doing something great already. With a Varnish Plus subscription, you can optimize, enhance and build your performance to perfection with the help of the experts behind the open source solution while still maintaining the flexibility and control you need.

Isn’t it too risky to leave your online, digital business to chance?

30. <http://www.forbes.com/sites/rajsabhlok/2013/07/18/open-source-software-the-hidden-cost-of-free/>

www.varnish-software.com

Los Angeles - Paris - London
Stockholm - Singapore - Karlstad
Dusseldorf - Oslo - Tokyo



www.varnish-software.com