niteco
Powering Your Ambition

info@niteco.com
niteco.com

E-GUIDE / part 3 of 3

Improving Web Performance in Episerver

July 2018

How to Improve Web Performance in Episerver

When it comes to the appeal of a website, nothing beats good performance. You can have the fanciest presentation with great visuals, perfect structure, and amazing content – if your site takes too long to load, nobody will stick around to see the fruits of your hard work.

Performance is key, and as an Episerver MVP, I know how to tease all the power out of an Episerver website. In this three-part E-Guide, I will show you how to do it. I will explain simple, common-sense approaches to speeding up your site, tell you where your site can lose considerable weight, and reveal the little secrets of Episerver's platform.

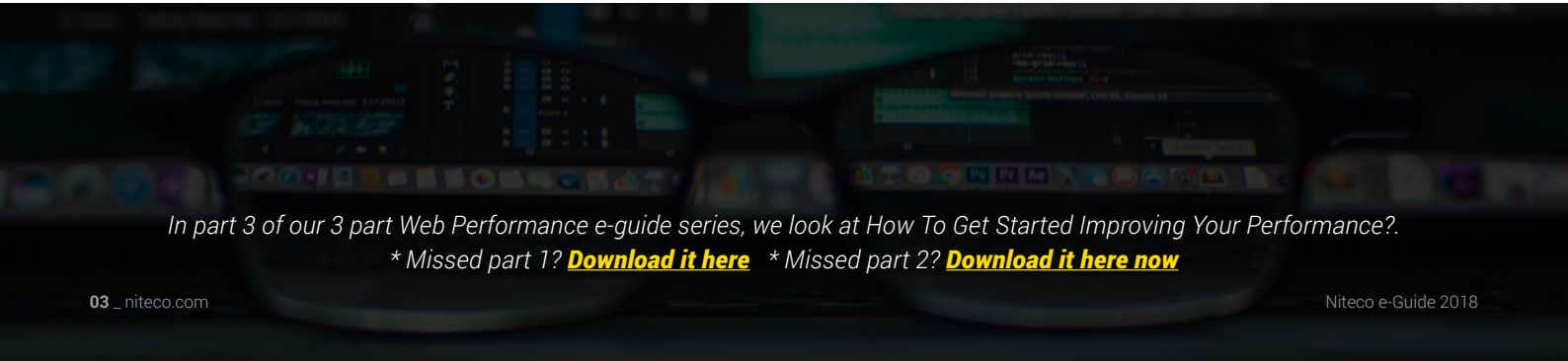
EMVP - Linus Ekström, Chief Technology Officer
The 3rd developer of the original team that created Episerver, Linus is a prominent figure amongst the industry's professionals. His greatest strength lies in the combination of technical knowledge and an eye for the big picture.

How to Get Started Improving Your Performance?

The best way to start performance testing of your site is to run a tool like Web Speed Test from the location that most of your users come from. If you are targeting several markets, for instance the US and Europe, you might want to run this twice with different locations. This gives you a benchmark for how your application is currently performing. Even better, turn on an active tool like SpeedCurve to be able to track any changes you make over time.

Once you have a benchmark in place, you want to go through the results and prioritize what changes would benefit your solution the most. Perhaps you do not currently have a CDN in place and would benefit from it? Or perhaps you have a lot of scripts that are blocking rendering? Usually, it is good to start small, with a single change, and try to implement it before moving on to the next one.

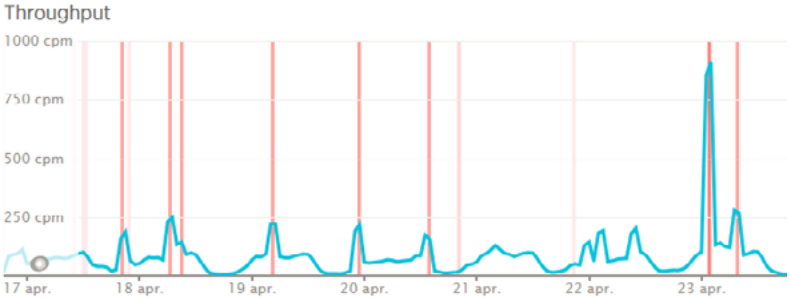
For most changes to the code, I use a tool like Google Lighthouse to be able to run tests before and after a code change with single user–single page testing. An individual user in your development team needs to test the build of your site on a single computer, without all the added complications that latency can bring to the equation. This is an especially good way to find front-end bottlenecks. The idea here is to find the most time-consuming and easy-to-fix changes and apply them one by one while continuously testing the performance. This can also find obvious server-side code that takes too long to execute in a single-user environment. Once it looks good on a local environment, you can use an automatic build to push it to a shared environment where you have a public URL, so that you can verify the improvements with a tool running externally.



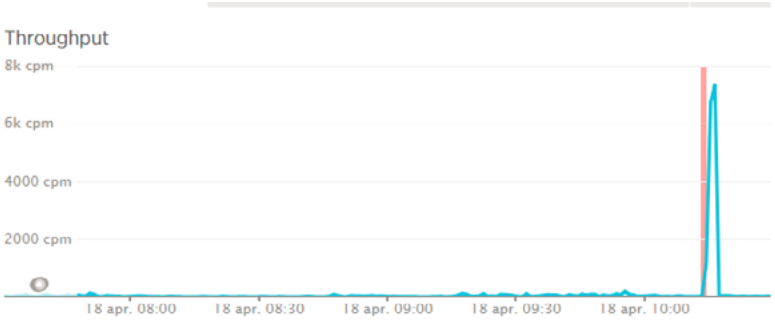
In part 3 of our 3 part Web Performance e-guide series, we look at How To Get Started Improving Your Performance?.
** Missed part 1?* ***Download it here*** ** Missed part 2?* ***Download it here now***

Example: Web Performance Issues

Let's look at a real-life example of an E-Commerce website that is suffering from slow performance and regular server downtime. This graphic presented by performance testing tool New Relic shows the parts server rendering time (purple), network time (brown), DOM parsing (yellow), and render times (blue) play in loading the site.

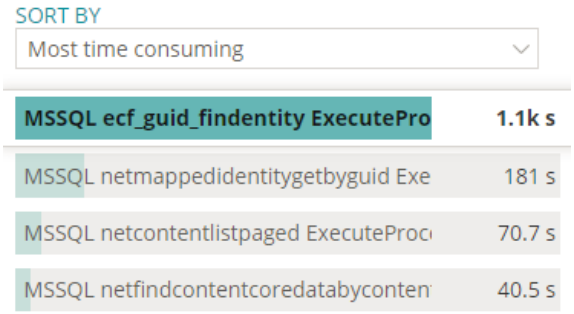


Looking at the results presented in a New Relic review shows that the site becomes unresponsive several times across a small number of days.



In this case, the blue line represents database calls per minute, while the red areas represent the server being unresponsive – 12 times in six days. From a first look, it becomes clear that there seems to be a correlation between the spike in database calls and the site becoming unresponsive.

A closer look at one of those instances shows that the unresponsiveness does not occur at the spike of database calls per minute (cpm), but rather when that number starts to rise.

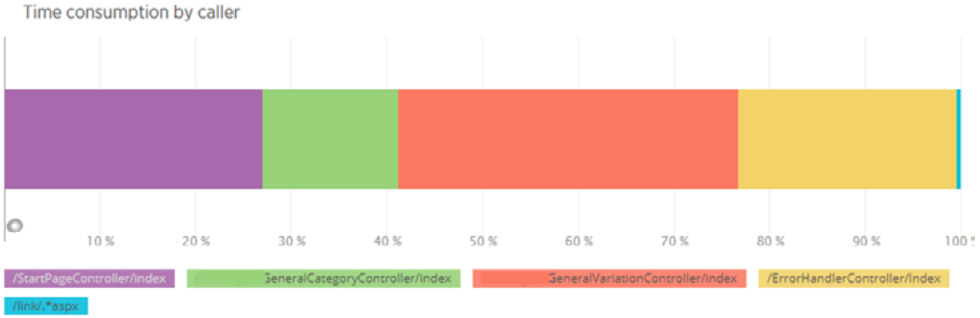


This could indicate that cache has been invalidated by an update, but could also be an effect of catalog data being imported from an external system. This would require a more in-depth investigation.

If we look more closely at one of the times when the site and database have issues, we can see that the stored procedure MSSQL ecf_guid_findentity ExecuteProcedure takes a much longer time than usual to execute during the spikes.

This stored procedure is part of the Episerver Commerce Core and is frequently called, but something seems to happen from time to time that makes it take a lot longer. In an initial review, there is no way for us to see what is causing this spike, we would need to investigate further to find out if a specific event prior to these spikes causes this behavior.

When we look at what parts of the web application are calling the database, it seems pretty spread out between the most common controllers, so it is probably not specific activity in the web that is causing the slowdowns. However, in this case the ErrorHandlerController takes up more than 20 percent of calls, which is something that needs to be fixed since exception handling is time consuming.



The GeneralVariationController, which is used to handle product pages, seems to be taking up the most time, with two major subtasks having to do with fetching and loading carts. The aforementioned serializable carts could help to reduce the number of calls this Controller has to make.

When we look at another performance testing tool, WebPageTest, we can start to see some issues that might cause drops in performance for the user.

The list shows us that the site uses Keep-Alive as well as GZip, which is positive. The two main issues lie in the two columns on the right. As we can see, only 10 percent of static content is cached, resulting in a lot of unnecessary calls to the database. All those red X signs should indeed be a red flag.

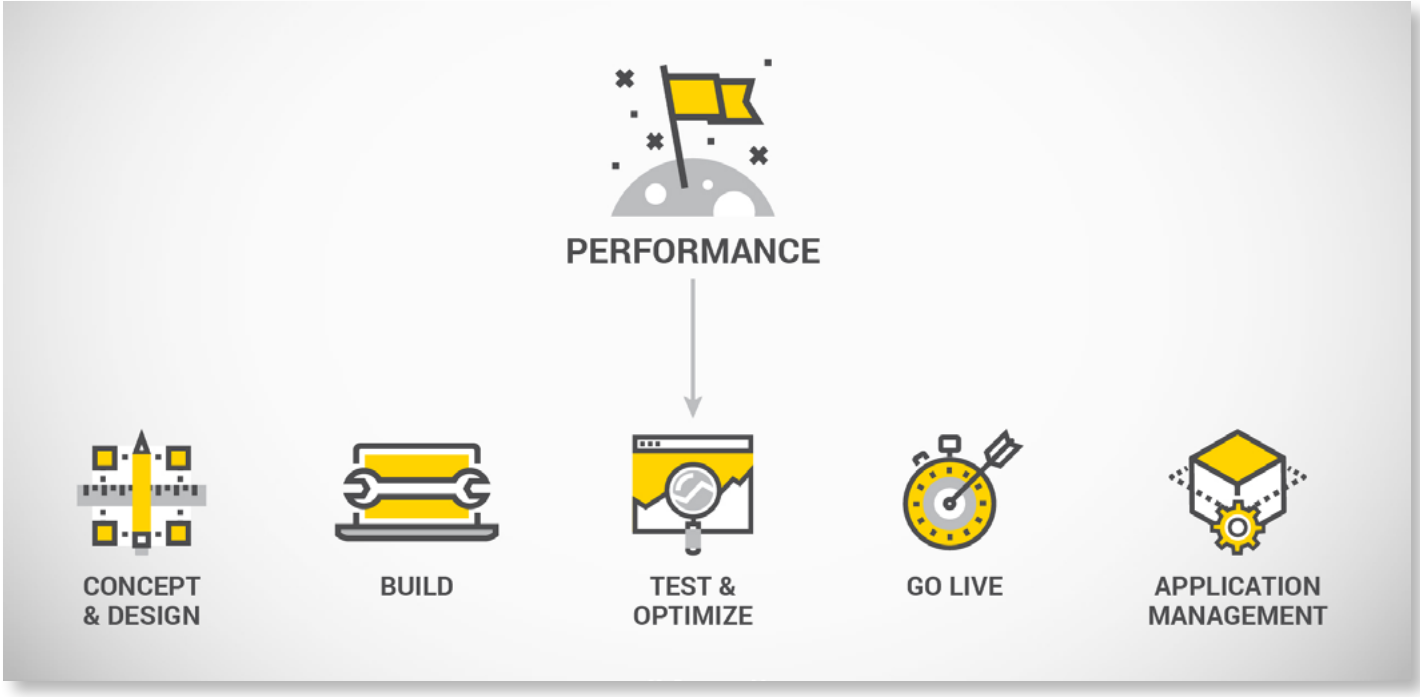
The report also states that the site uses a CDN, which is a good choice for all the reasons indicated earlier. However, the CDN has been implemented poorly. Only 76 percent of the items that could potentially be stored in the CDN are actually cached there. In this case, you need to check Episerver's configuration files to ensure that all static content, both the one stored in Episerver as well as application files like script and CSS, has proper cache settings. You can find information in Episerver's online documentation **here**.

Full Optimization Checklist

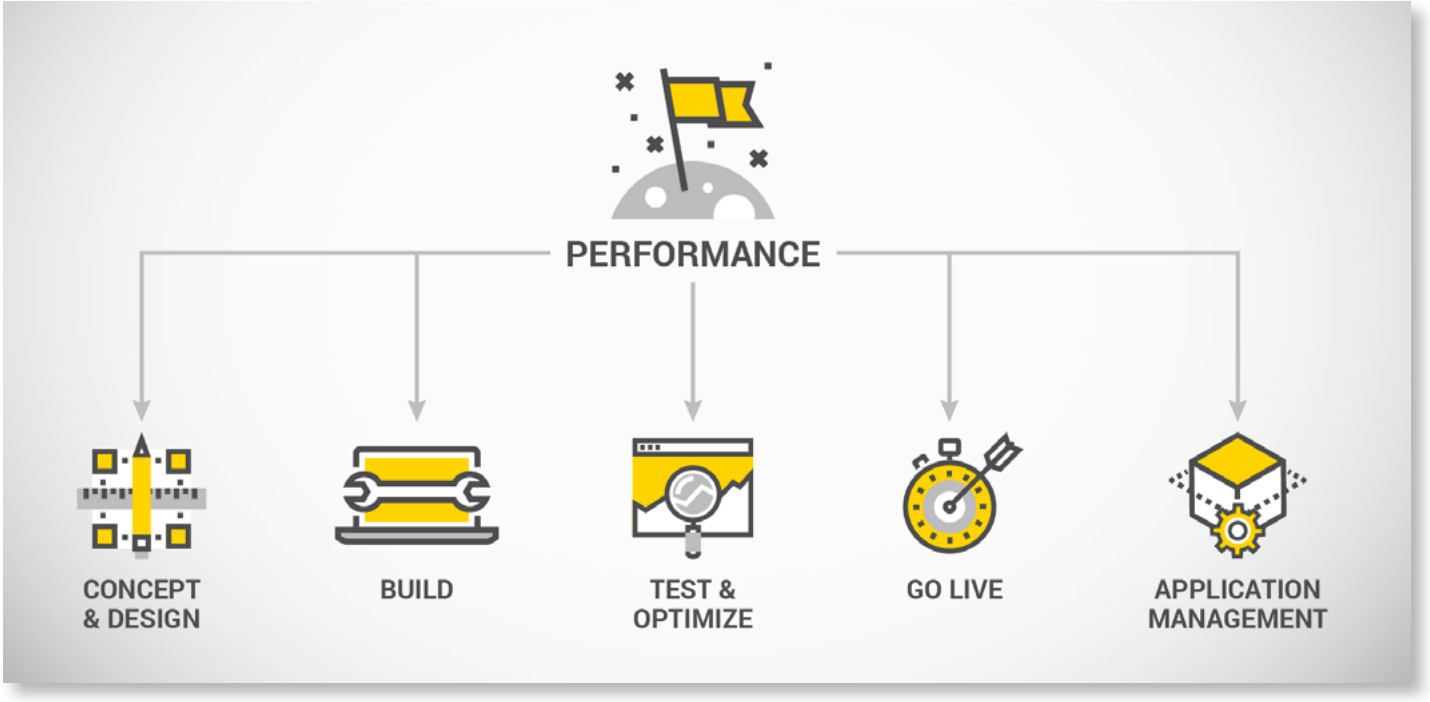
	Keep-Alive 100%	GZip 93%	Compress Img 87%	Progressive 87%	Cache Static 10%	CDN Detected 76%
[redacted]	✓	✓				
[redacted]	✓	✓			⚠	✗
[redacted]	✓	✓			✗	✗
[redacted]	✓	✓			✗	✗
[redacted]	✓	✓			✗	✗
[redacted]	✓	✓			✗	✗
[redacted]	✓	✓			✓	✓
[redacted]	✓	⚠			✗	✗
[redacted]	✓	⚠			⚠	✗
[redacted]	✓				⚠	✗
[redacted]	✓	✓			⚠	✓
[redacted]	✓	✓			✗	✓
[redacted]	✓				⚠	✗
[redacted]	✓				⚠	✗
[redacted]	✓		✓		✗	✗
[redacted]						
[redacted]	✓				⚠	✗
[redacted]	✓		✓		✗	✗
[redacted]	✓				✗	✗
[redacted]	✓	⚠			⚠	✓
[redacted]	✓		✓		✗	✓
[redacted]	✓		⚠	✓	✗	✓
[redacted]	✓	⚠			✗	✓
[redacted]	✓				✗	✓
[redacted]	✓				✗	✓
[redacted]	✓		✓		⚠	✓
[redacted]	✓		✓		⚠	✓
[redacted]	✓		✓		✗	✓
[redacted]	✓		✓		✗	✓
[redacted]	✓	✓			✗	✓
[redacted]						
[redacted]	✓				✗	✓

When Should You Think About Web Performance?

Conventional wisdom dictates that performance is part of one of the basic steps in project management, namely testing and optimization.



This approach may have been good enough in the past, but doesn't cut it anymore nowadays. What we should be doing is this:



Performance must be part of every step in the design and development process. In the conceptualization and design phase, ideas that could potentially lower performance should be weeded out in favor of designs that allow for quicker response times. During development of a site, programmers should already think about performance instead of leaving those concerns to the testers, or even worse, the end customer or users. Adding a new feature that you think will increase sales? Don´t forget to consider how this might potentially affect performance, since that can have a negative impact on sales, making the new feature pointless.

And of course, during launch and maintenance, performance optimization leading to increased customer satisfaction should be a priority. The quest for performance should be a cycle that never ends. As you measure and prioritize your site's performance, you find new ways to improve it. After you have finished said improvements, you should measure again and find new ways to increase performance.

Conclusion

Improving your website's performance can be a long and arduous process. It's also one that never really ends, since new issues can always arise, requiring regular maintenance. However, it can make an impact on your bottom line. Whether that impact is positive or negative depends on the work you put into web performance.

As we established earlier, a slow and laggy site can be the death of your business, leading your visitors to leave your offerings for greener, high-performance pastures. A slick, highly responsive site, on the other hand, can keep users coming back for more.

So in order to unlock the full potential of your website, focusing on top-notch performance is absolutely essential. With the information in this E-guide, you have a good starting point for your own journey toward high performance. So don't delay, tune your website now to make it the best it can be.

POWERING YOUR EPISERVER AMBITION

Niteco AB _ **STOCKHOLM**
Norrtullsgatan 6, 5tr, SE-113 29
Stockholm, Sweden
+46 (0) 700 355 830 | sweden.info@niteco.se

LONDON _ Niteco Group Ltd.
3 More London Riverside, London, SE1 2RE
United Kingdom
+44 (0) 746 012 2355 | uk.info@niteco.co.uk

Niteco Group Ltd. _ **SYDNEY**
PO Box 868 Rozelle NSW
Australia 2039
+61 (0) 405 208 629 | australia.info@niteco.com

SAN FRANCISCO _ Niteco Group Ltd.
38505 Bautista Canyon Way, Palm Desert
CA 92260, USA
+1 (0) 415 871 2455 | usa.info@niteco.com

Niteco Group Ltd. _ **HONG KONG**
36/F, Tower Two Times Square
1 Matheson Street, Causeway Bay, Hong Kong
+84 (0) 128 801 2674 | hk.info@niteco.com

HANOI _ Niteco Vietnam Co. Ltd.
C'Land Tower, 14th Floor, 156 Xa Dan II Street
Dong Da District, Hanoi, Vietnam
+84 (0) 243 573 9623 | info@niteco.com

Niteco Vietnam Co. Ltd. _ **HO CHI MINH CITY**
E.Town Building 1, 2nd Floor, 364 Cong Hoa Street
Ward 13, Tan Binh District, HCM City, Vietnam
+84 (0) 286 297 1215 | info@niteco.com

niteco.com

Niteco E-guide / NITECO1807