A/B Testing Tool Review

To A/B or Not to A/B

Michael Gooding | January 2015

Table of Contents

Overview

"A/B testing is a marketing tool to test the usability of a website and try and make it as user friendly as possible. By running simple experiments and monitoring the results of real users A/B testing can be used to find the most effective ways to lay out a web page in order to make the most revenue. There are many tools on the market that do this and some of the most popular use client side code to create the variations. By introducing this code performance suffers and as a result any gain from running the experiment can be offset by the performance degradation and lost revenue during that time. This report aims to quantify the performance degradation and offer some guidelines around when A/B testing."

Introduction to A/B testing

A/B testing tools are a great way to test the layout and content on your website and collect real user analytics to show what works best. The concept is to create 2 versions of the same page but with a slight variation, run an experiment for a period of time and split traffic to each variation during this period. Once the experiment is complete you can review the bounce rate or conversion rates of each variation and determine which works best for your audience. Some tools offer additional features that allow you to have more than 2 variations on a page, these are known as Multi Variant testing tools (MVT).

When implementing these experiments there are 2 main ways to do this. The first places code in the browser and will change the page as it is being built. This is known as a client side tool. The advantage to these is that they often don't require any technical input and can be used by a non-technical audience e.g. marketing. The tool will provide a "What you see is what you get" (WYSIWYG) interface allowing non-technical people to create variations of a webpage without getting anywhere near the source code. The additional advantage to this is that the development team can continue with their main job of building new features and functions. Often these tools provide the traffic splitting function and the analytics of results as well.

The second approach to creating these experiments is to create 2 versions of the same page and make the decision of which version they see when a user comes to the site. This decision is made by the server (or load balancer or proxy etc.) and these solutions are known as server side solutions. In order to create an experiment you would need access to the source code of the website and somewhere in the backend to apply the routing strategy, in short this requires the input of the technical team. Often these tools will not have the analytics included so any routed traffic to experiment A or B must be tracked.

This document focuses on the client side solutions as they seem to be the most popular due to the lack of technical input required. We will look at what impact these tools have on performance and try and quantify it, as well as provide some best practice advice.

The Key Players

The table below shows the tools used in this analysis. They represent some of the bigger players in this market.

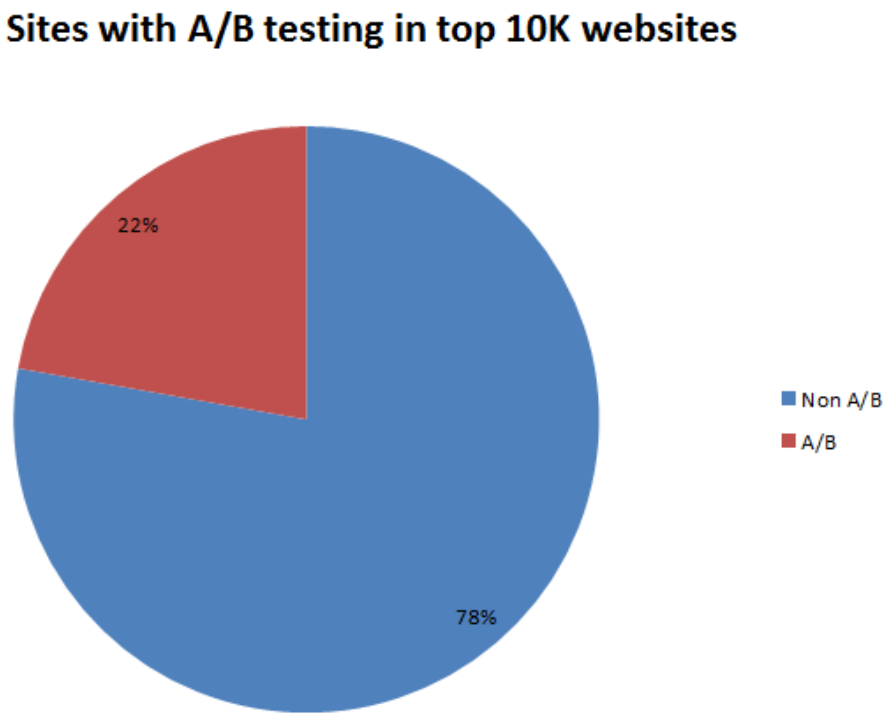
Where any tools were not purely A/B testing providers every effort was made to identify just the A/B testing part of the tools. The analysis was carried out by looking for the presence of the JavaScript code inserted to create the experiments. The data was taken from HTTP archive unless otherwise stated.

Where all data for A/B testing tools is used, all of the tools below were included, however when individual tool performance was reviewed we only looked at the top 6 provided based on the sample of data. These are highlighted at the top of the table below.

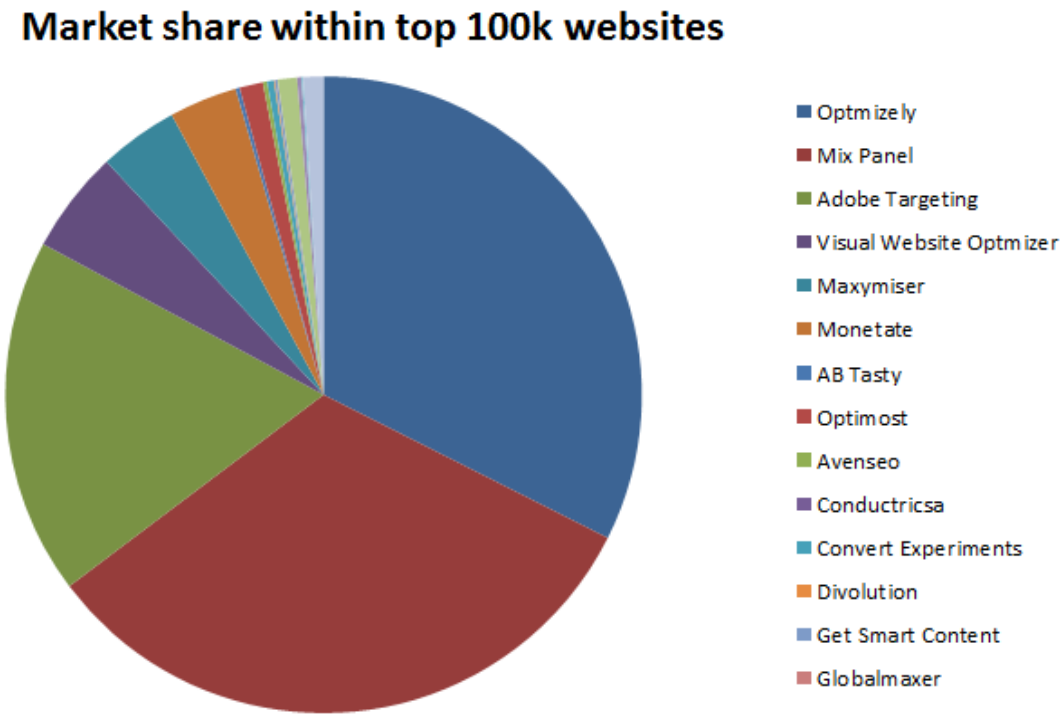
	URL	Initial set-up cost	Ongoing costs	Type	A/B or MVT
Optimizely	https://www.optimizely.com/	Free trial	$19-$399 /month	Client-side	Both
mixpanel	https://mixpanel.com/	None	0$ -$2000 / month or bigger depending on data points	Client side	A/B
Adobe	http://www.adobe.com/uk/solutions/testing-targeting.html	Contact Adobe	Usually a four-figure monthly fee	Client-side	Both
Visual Website Optimizer	https://vwo.com/?v=3	None	$49-$729 /month	Client-side	Both
maxymiser	https://www.maxymiser.com/	None	Usually a four-figure monthly fee	Client-side	Both
monetate	http://www.monetate.com/	Contact Monetate	Flat monthly fee based on previous 12 months' traffic	Client-side	Both
AB TASTY	https://en.abtasty.com/	Free Trial	From €40	Client-side	Both
Autonomy optimost	http://www.autonomy.com/products/optimost	Contact Optimost	Usually a four-figure monthly fee	Client-side	Both
AVENSEO	http://www.avenseo.com/en/	Free trial	From €300	Client-side	Both
CONDUCTRICS	http://www.conductrics.com/	Contact Conductrics for details.	Monthly fee (Contact Conductrics for details.)	Client & Server-side	Both
convert	http://www.convert.com/	None	From $9 per month	Client-side	Both
DIVOLUTION	http://www.divolution.com/de/	None	Usually a four-figure monthly fee	Client and Server-side	Both
GETSMARTCONTENT	http://www.getsmartcontent.com/	None	From $500 per month	Client-side	A/B
GlobalMaxer	http://www.globalmaxer.com/	From £2,000	Usually a four-figure monthly fee	Client-side	Both
Google	http://www.google.co.uk/analytics/	None	None	Client-side	A/B
hiCONVERSION	http://www.hiconversion.com/	Contact HiConversion	Contact HiConversion	Client-side	Both
Marketizator	http://www.marketizator.com/	None	From $19/month	Client-side	Both
SiteTuners	http://www.sitetuners.com/	None	$1k-$2.5k per test	Client-side	Both
webtrends Optimize	http://www.webtrends.com/products-solutions/remarketing/	Contact Webtrends	Usually a four-figure monthly fee	Client-side	Both

Data source: www.whichmvt.com

The use of A/B testing tools within the top 10K websites according to Alexa is approx. 22% so this is a popular market and these tools are used on 1 in 5 websites.



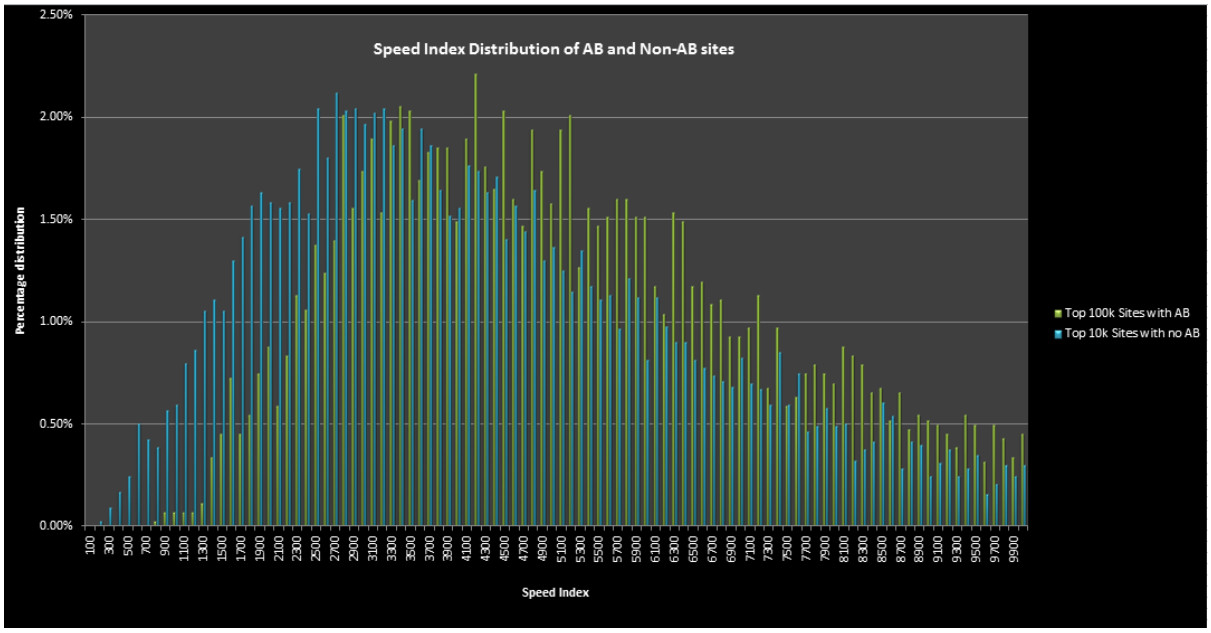
The Graph below shows the market share of these tools in the extended top 100k websites according to Alexa. The top 6 tools make up the lion share of the samples.



Performance Review

Speed Index

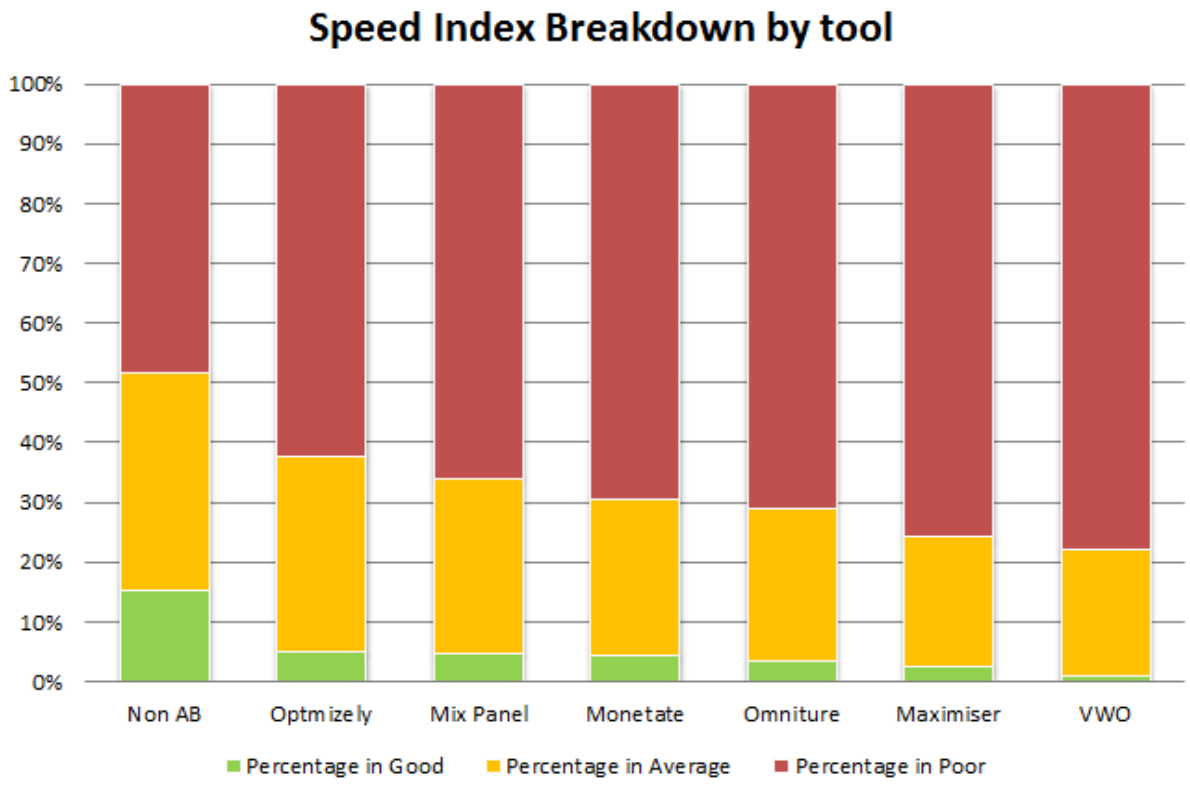
In a recent talk about all 3rd Party content The NCC Group found something similar to the below graph. In general sites using A/B testing tools have a worse Speed Index score than those that don't. Speed index is a value based the on the area of a graph that shows a visually complete page, for more information read the Web Page Test documentation.



This graph clearly shows the distribution of the sites with A/B testing in green is shifted to the right which represents a higher Speed Index score.

The data above is based on all of the tools we looked at. The appendix contains the results for the most popular 6 tools against sites without any A/B testing, however reviewing the results it became clear that when viewing the Speed Index that all companies follow a similar pattern.

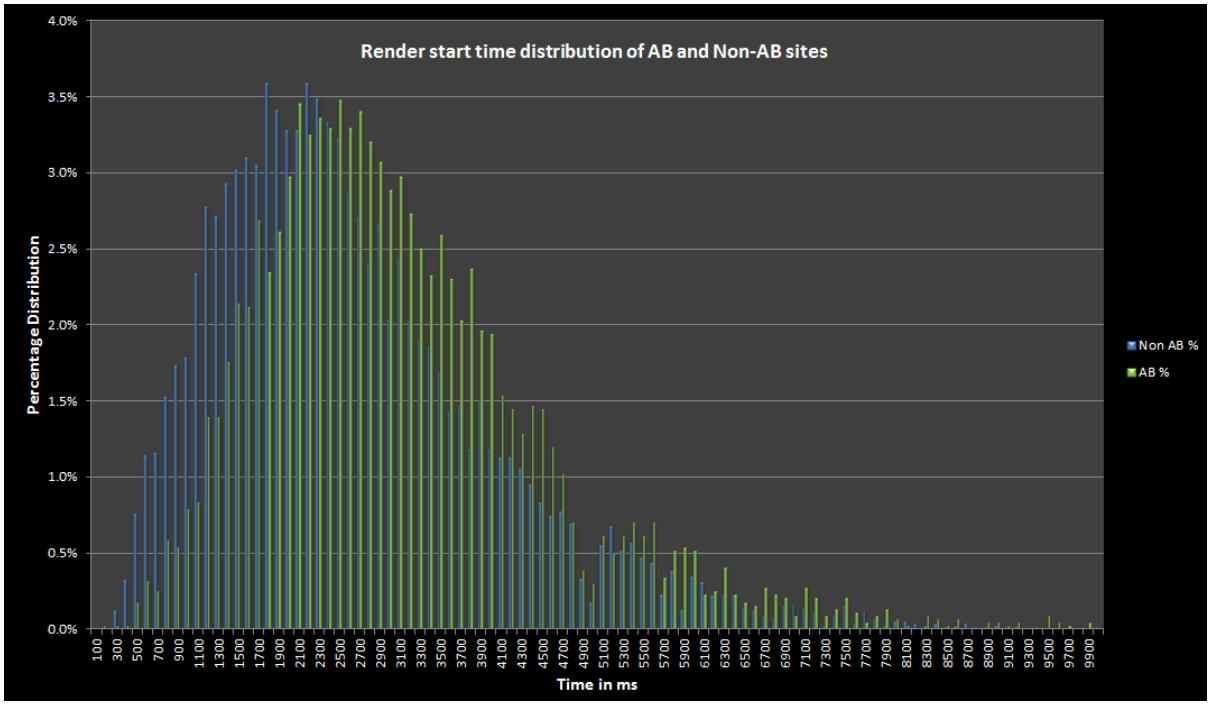
Looking at the distribution above we could draw a conclusion that a good Speed Index is something less than 2000 and a poorer performing site may have a Speed Index in excess of 4000. When we then look at the distribution across the top 6 tools we see the picture becoming a lot clearer.


Speed Index Breakdown by tool
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Non AB
Optmizely
Mix Panel
Monetate
Omniture
Maximiser
VWO
Percentage in Good
Percentage in Average
Percentage in Poor

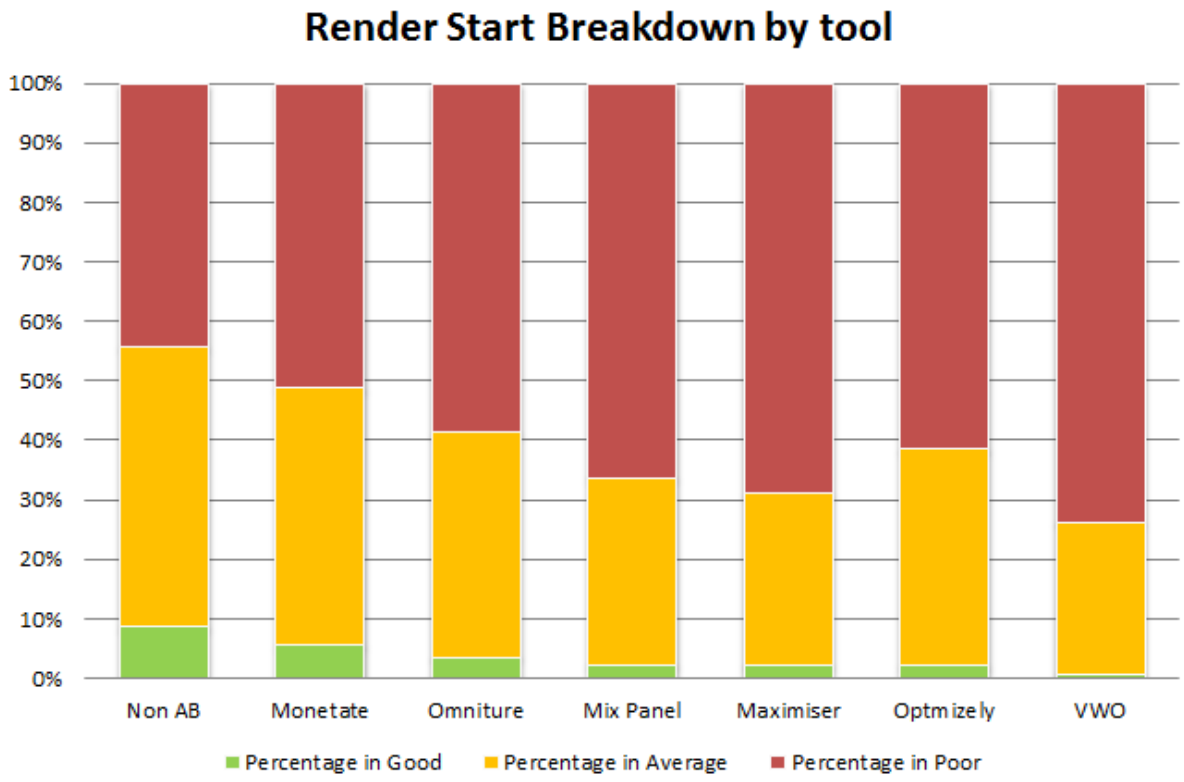
Render Start

While this metric is linked to the Speed Index it also contains insight into when something becomes visible to the user, providing feedback that the site is loading. Often a good render start time can result in perceived better performance against sites that take the same time to load completely.

The graph below shows a similar story to the Speed Index that the distribution of render start time (green) is shifted to the right.

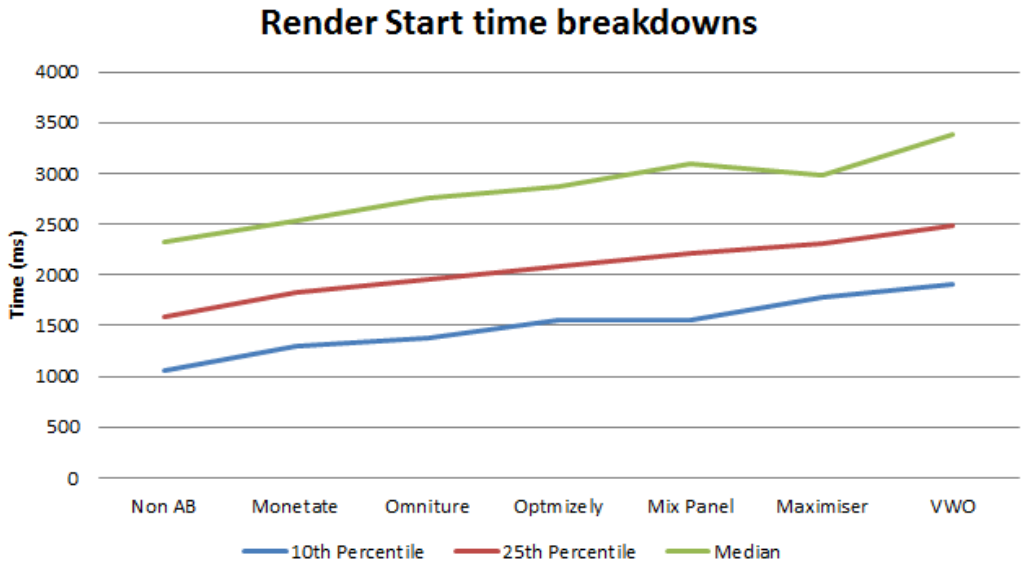


A good render start time should be around the 1 second mark and for a poor render we have used a quite generous 2.5 seconds. The graph again shows very different results between tools with an A/B testing solution and those without. It also shows that although similar the tools also vary in their performance.



While this graph is quite similar in terms of the overall results it is interesting that some of the tools have changed position, notably Optimizely has gone from being one of the best to one of the worst meaning that it will get you to a visually complete state faster, but it will be slow to start that process.

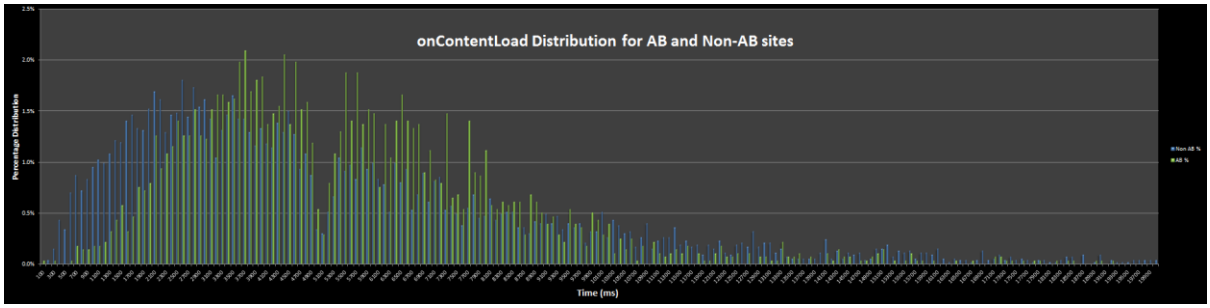
If we then take a look at the 10th Percentile, 25th Percentile and median data we can visually see the impact these tools have on render start time. We can see that the render start time goes up by ~0.2 to ~0.9 seconds depending on which tool you choose.

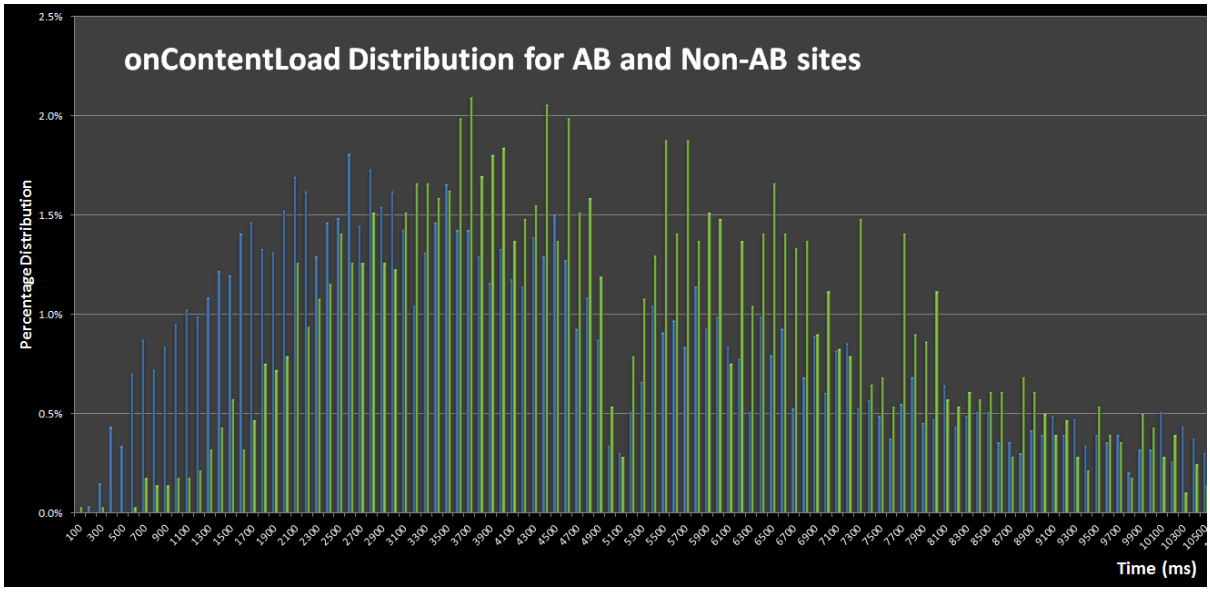


On Content Load

The data from HTTP Archive is taken from running tests in Web Page Test, which uses an IE9 browser to test. The metric for onContentLoad is the same as *document.readyState == interactive.* If a page has JavaScript that has been deferred this will run on this event and once that is done the DOM will be complete. If there is no deferred JavaScript then this measurement is very similar to saying the DOM is complete.

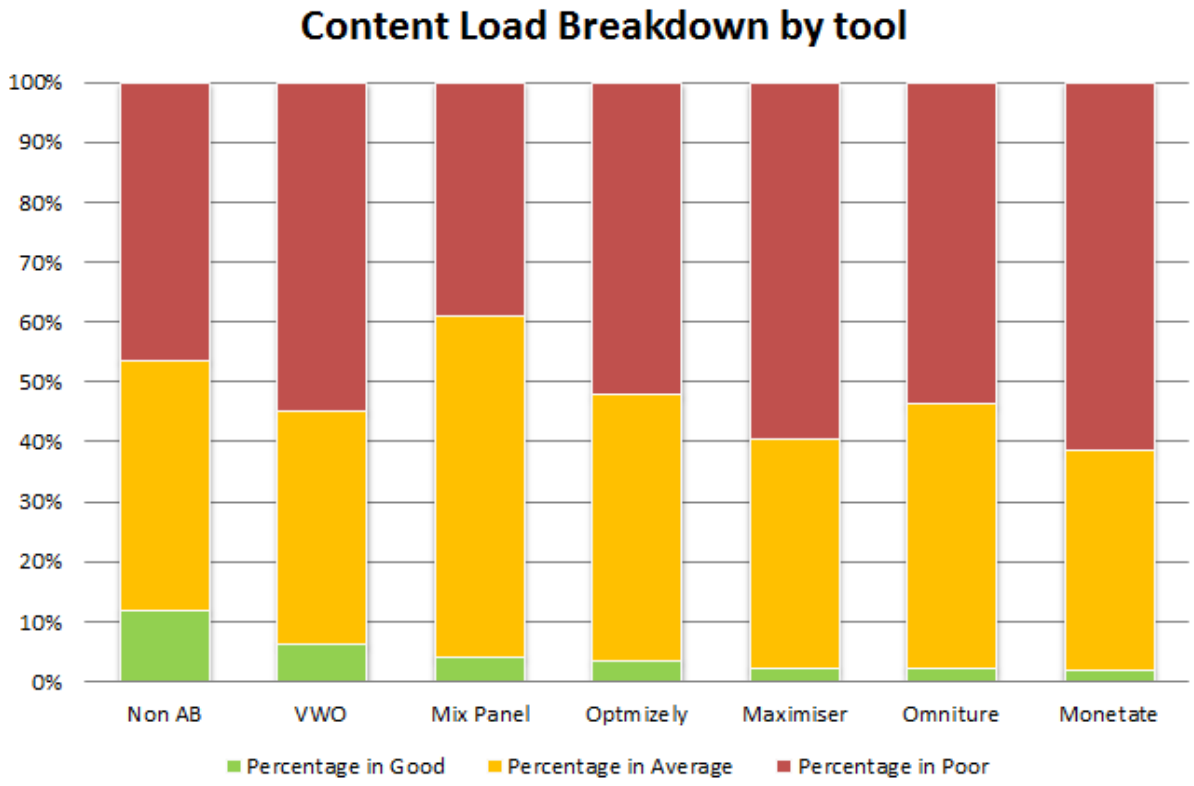
Due to the long tail of slow results this graph gets extended somewhat but we see the same pattern as we saw before.



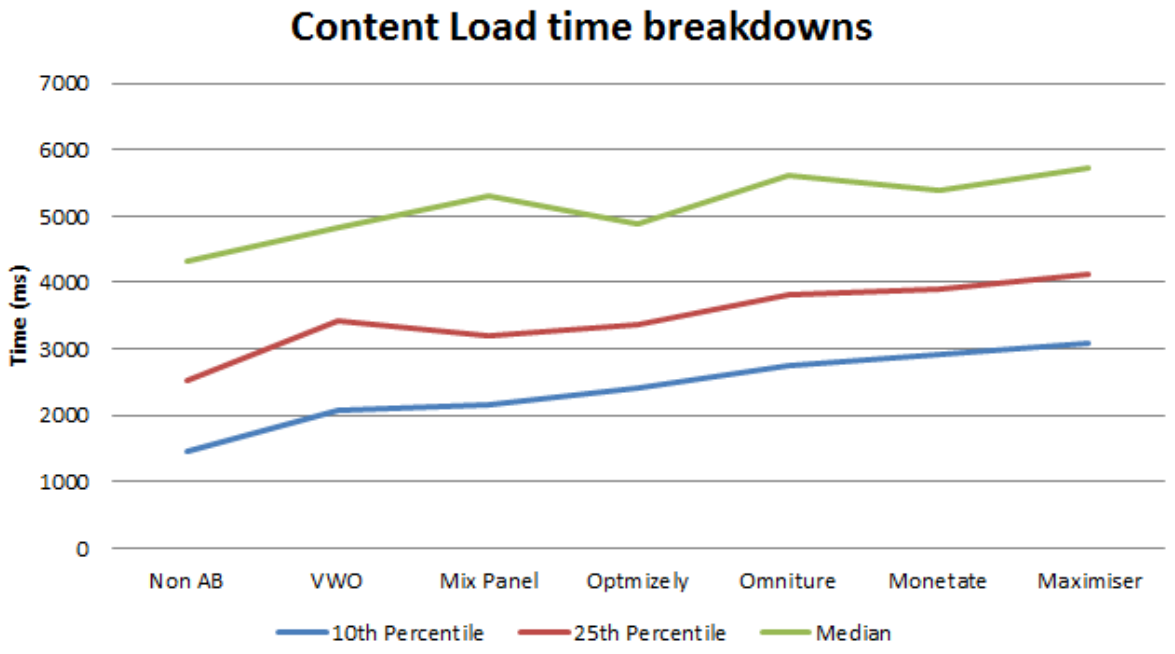


Zooming in a little on the graph allows us to set a good target of 2 seconds and a poor threshold of greater than 3 seconds to see how the individual tools compare.

The graph below shows that once gain the non-A/B sites have over 10% of samples in the good category and this slowly tails off between the tools. Visual Website Optimizer (VWO) does quite well with this metric compared to the others. Interestingly the difference between the non-A/B sites falling into the poor category and those with A/B tools on the page is quite consistent.



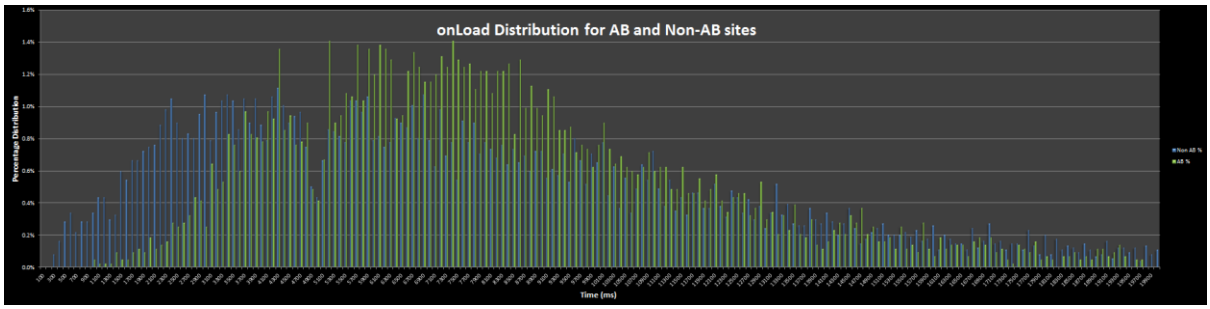
If we then look at the performance numbers for the tools we can see that using an A/B tool will add between ~0.8 seconds and ~1.5 seconds to you content load times depending on the tool you use.

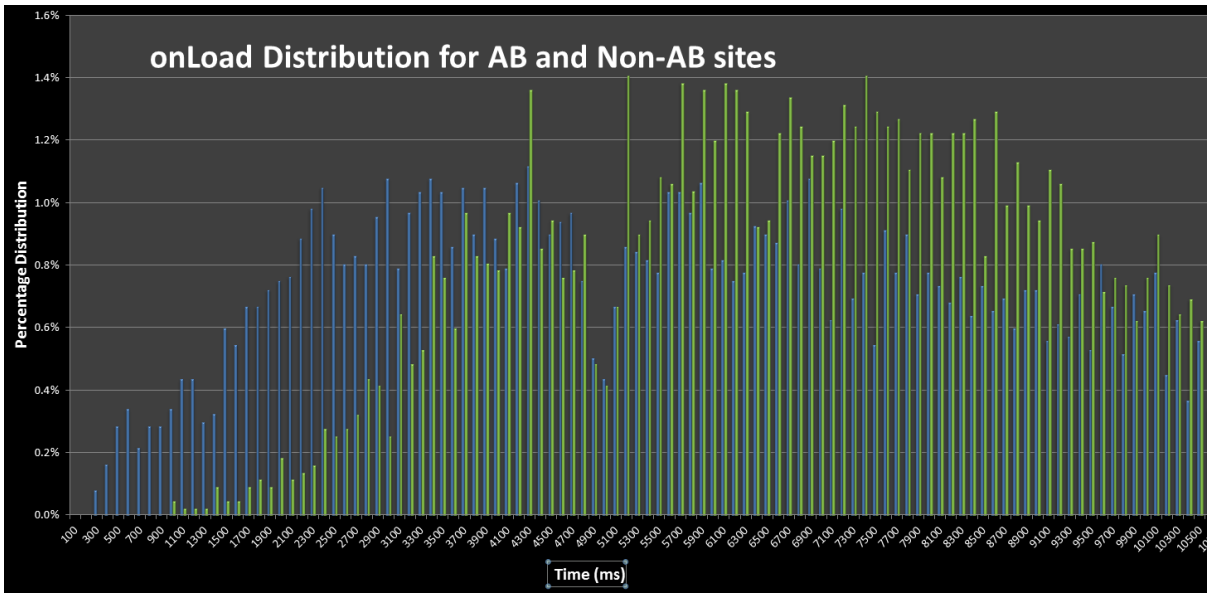


onLoad Event

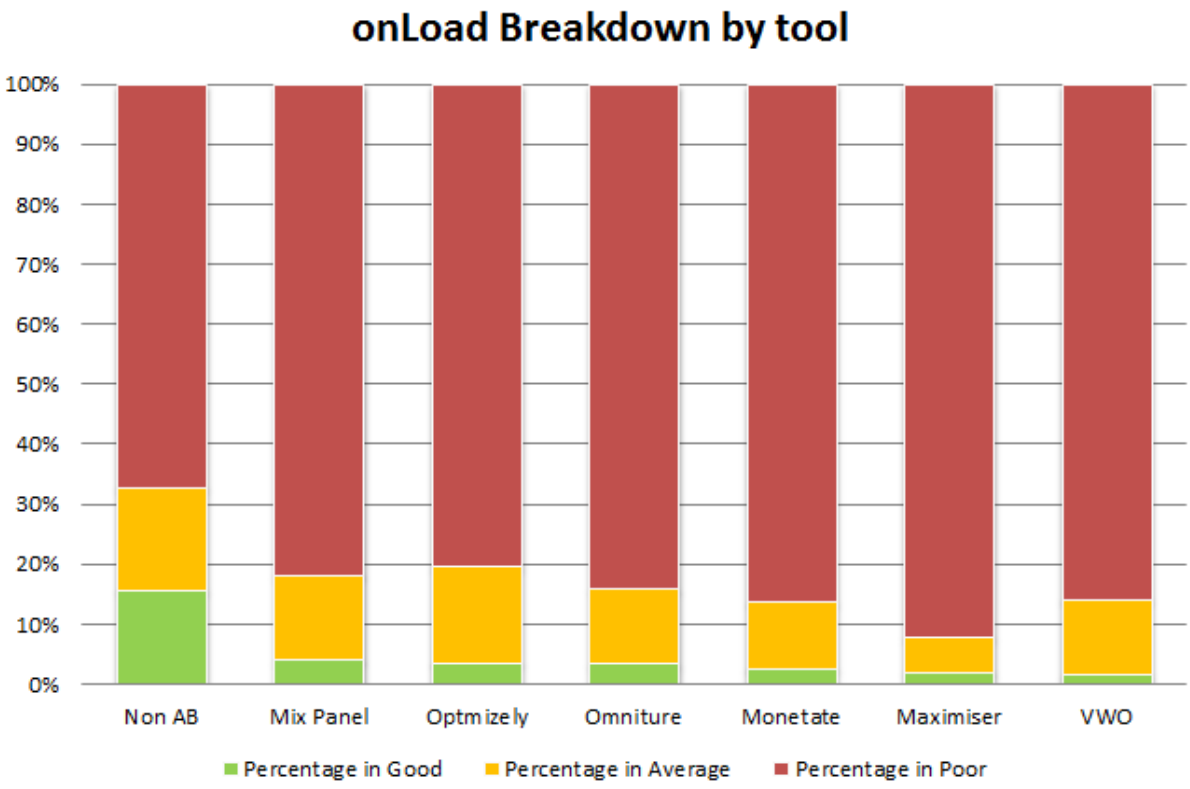
This represents one of the best metrics available today to gauge when a page is ready for your end user to use. This is not always the case but as a rule of thumb this represents a good indication of readiness.

As we have seen before with the long tail of really slow results the graph becomes quite condensed, however if we zoom in on the results up to 10 seconds (after all no webpage should take more than 10 seconds … EVER!!) we see similar results. The green A/B sites have all shifted to the right of the graph and have much higher distributions in the 6-9 second area.

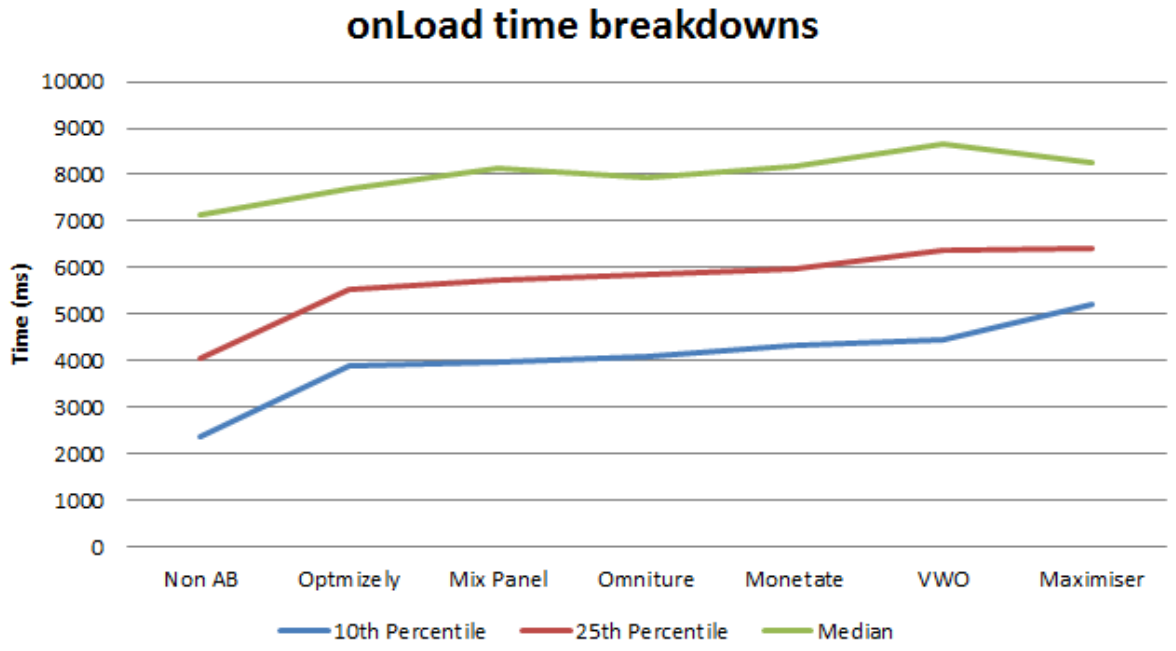




Using 3 seconds as a good target and 5 seconds as a poor threshold we see similar results as before. Strangely, using these targets even the non-AB sites have a lower set of Good & Average examples, with Poor sites equaling about 65% compared to around 45% in the previous metrics. However, what doesn't change is the pattern of the AB tools showing much lower numbers of sites in the good category. At these targets 80% or more of all the sites using AB tools fell into the Poor category regardless of the tool being used. For Maximizer this was over 90%.

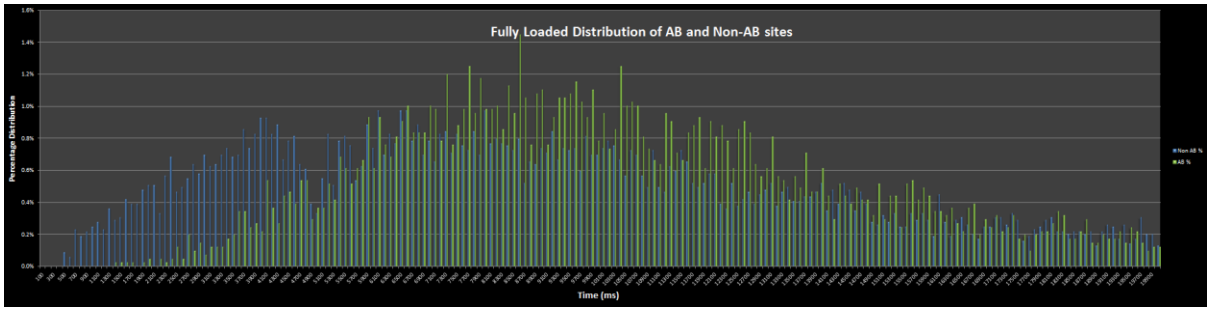


Reviewing the performance numbers we see here that the content load time increase by ~1 second for the median, ~2 seconds for the 25th Percentile and ~3 seconds for the 10th Percentile.

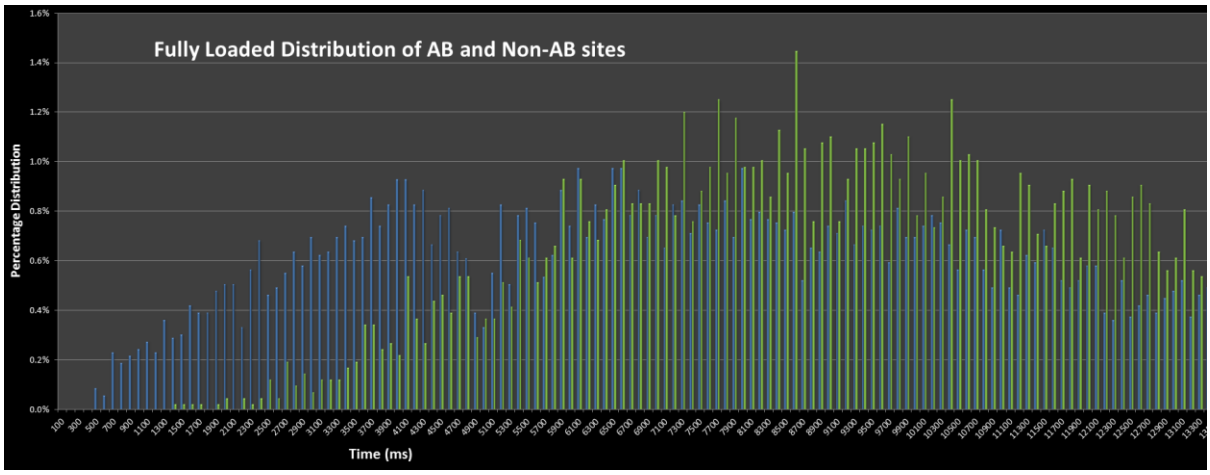


Fully Loaded

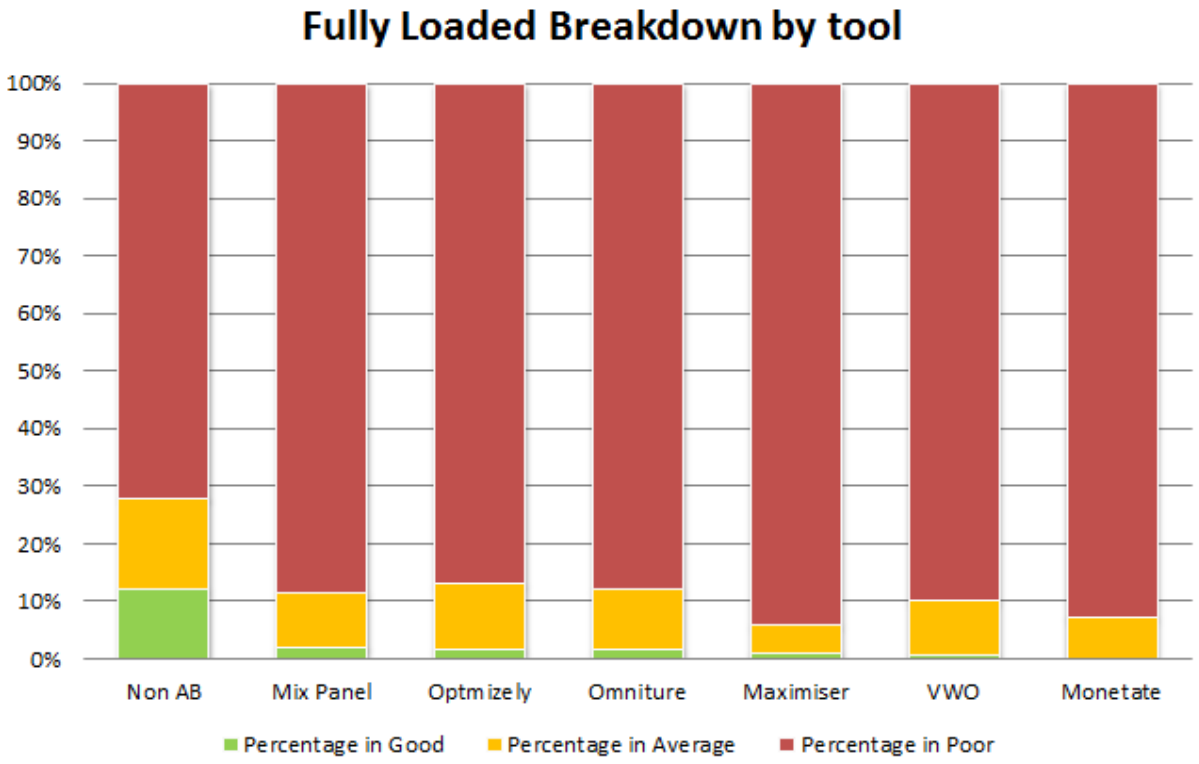
The fully loaded metric represents the time it takes to Load and execute everything on the page. As with the previous metrics there is a very long tail of poor results and the pattern seen here is the same.



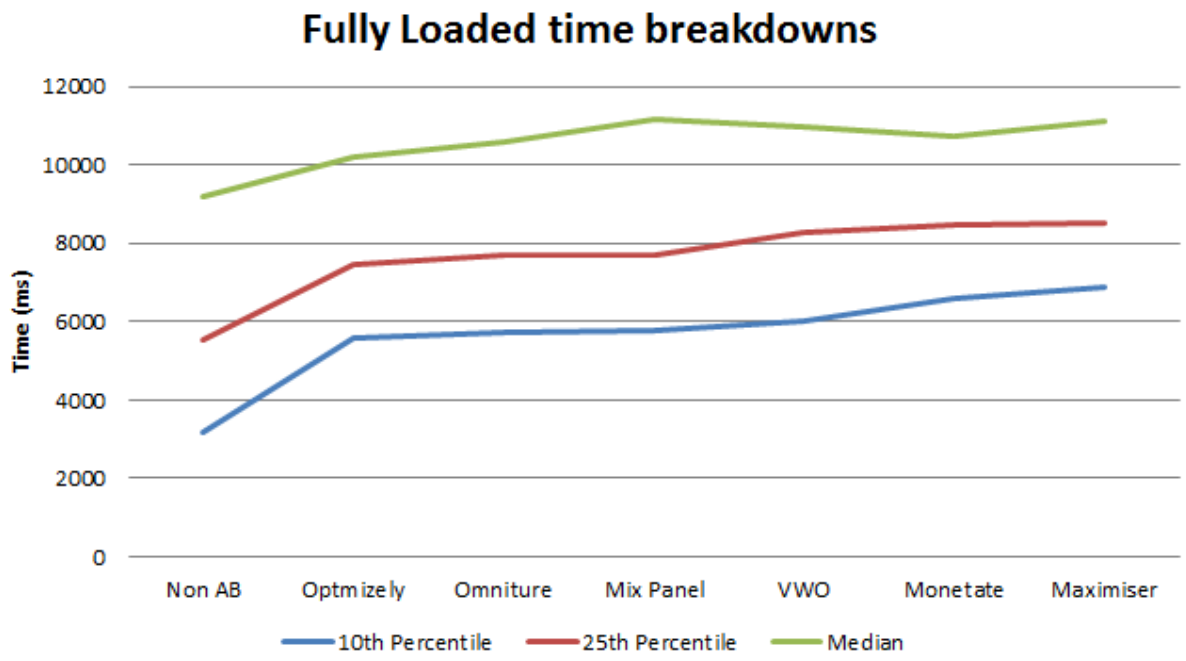
Zooming in on the results for samples under 10 seconds we can see the distribution of sites with A/B testing only starting around the 3 second mark compared to the 1 second mark of the non-A/B sites.



By applying a 3.5 second boundary for a good site and a 6 second threshold for a poor site, as with onLoad we see a really poor showing from all sites (not just those with A/B tools on the page), however we also see the same pattern of reduction in good sites compared to those with no A/B tools.

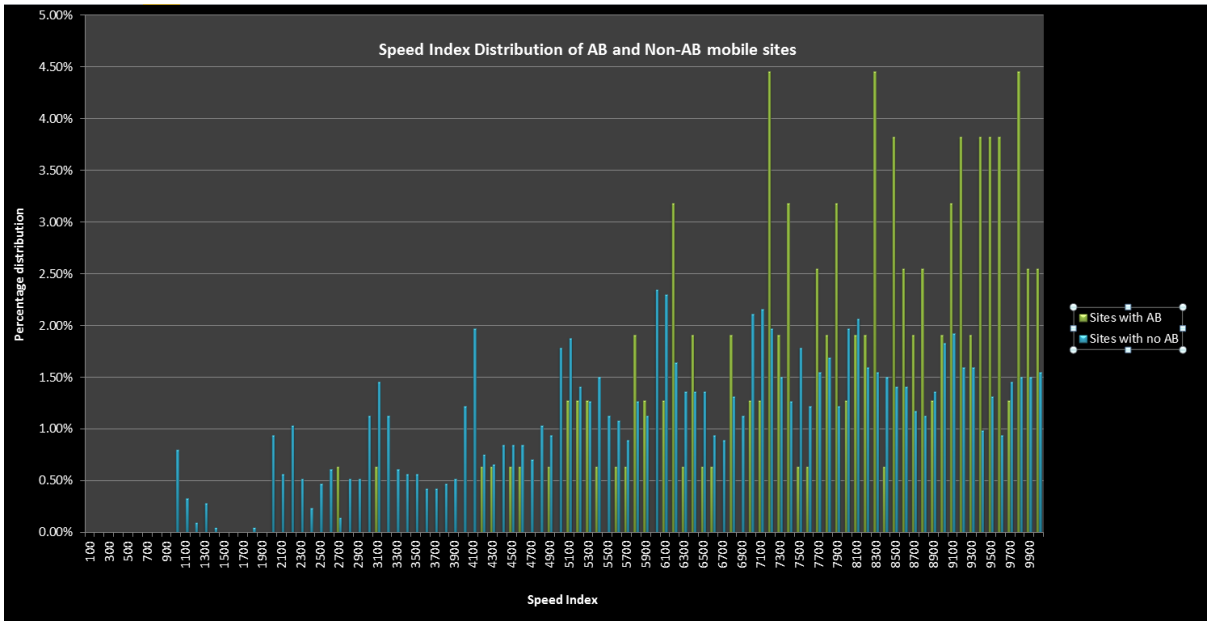


Reviewing the performance times and the median, using A/B tools can add between 1 and 2 seconds to the fully loaded page time. This gap is larger in the 10th & 25th Percentiles.



Mobile

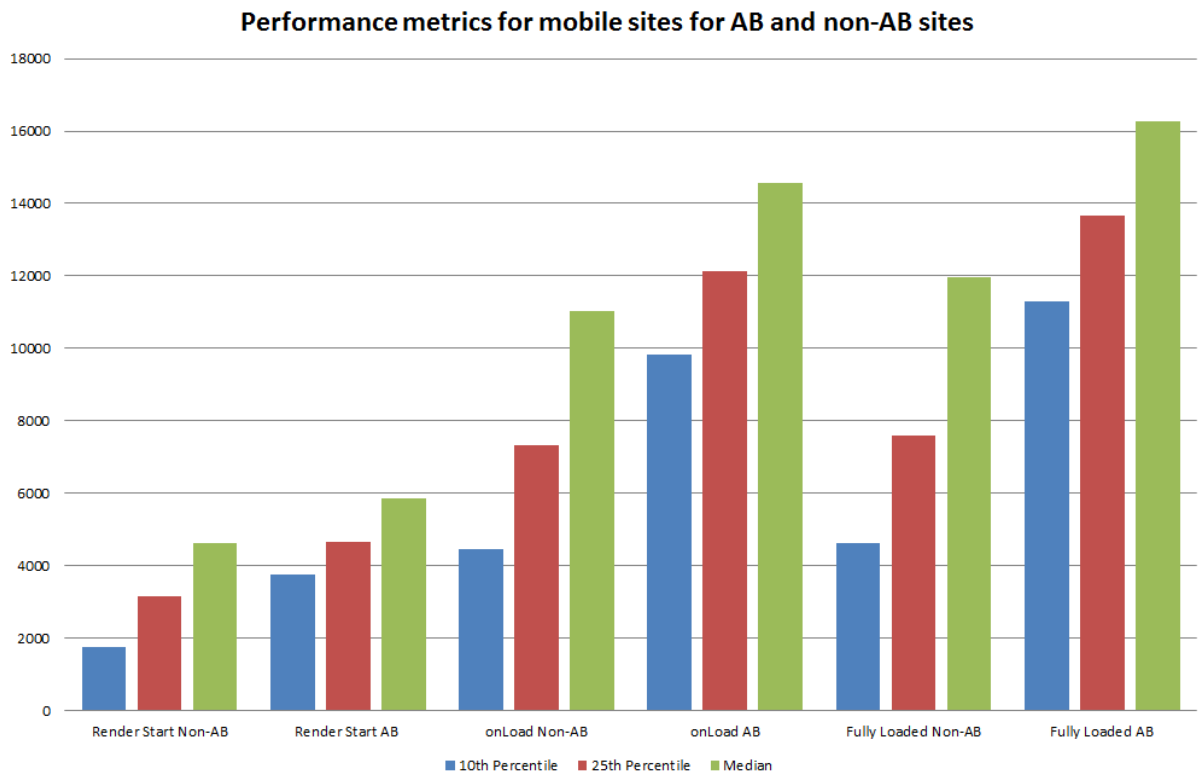
The previous section has shown applying a client side A/B testing tool decreases performance and this can be by as much as 4 seconds. These results were all taken from Desktop sites. What happens on Mobile?



Due to the lower sample sizes there is less of a clear distribution but you can still see fewer sites with AB tools on the left hand side that is also coupled with the high green bars on the higher end of the scale. This is also not a great advert for mobile performance in general.

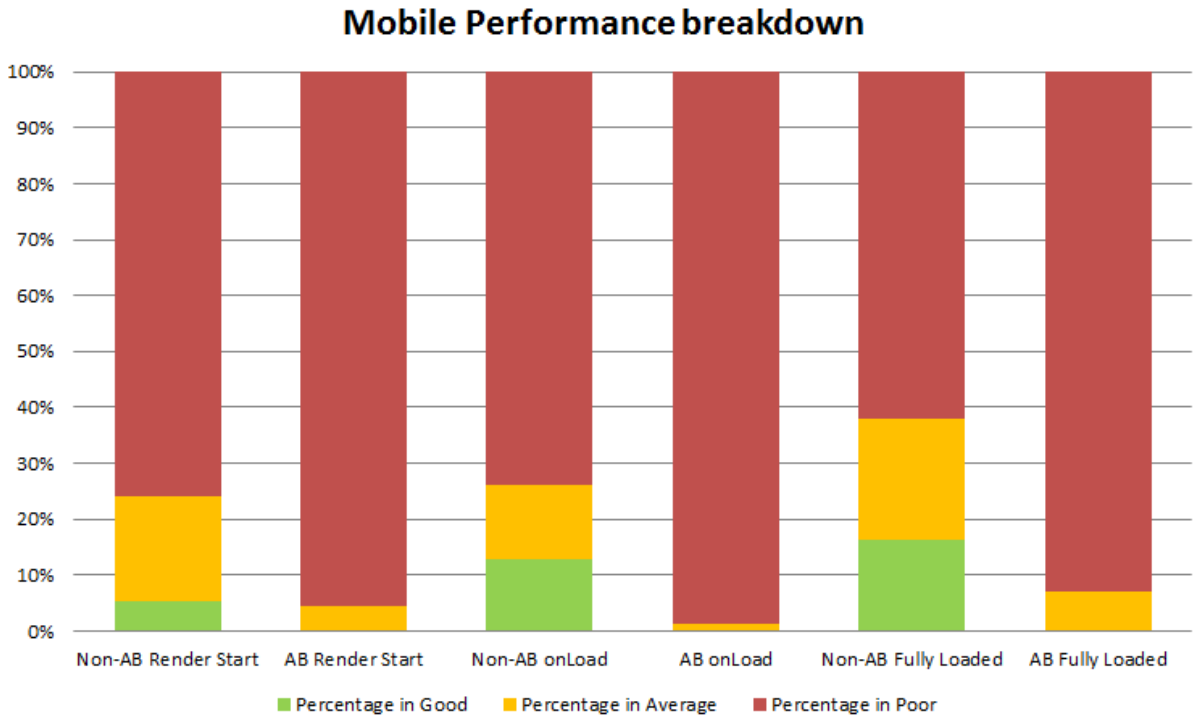
Looking at the performance metrics below and focusing on the mean results we can see:

- Render Start increases by 1.2 seconds (worst case desktop was 1 second)
- onLoad increases by 3.5 seconds (worst case desktop was 1.5 seconds)
- Fully loaded increases by 4.2 seconds (worst case desktop was 2 seconds)



If we introduce some targets for mobile as:

- **Good** render start 2 seconds and below
- **Poor** render start above 3 seconds and above
- **Good** onLoad 5 seconds and below
- **Poor** onLoad 7.5 seconds and above
- **Good** fully loaded 6 seconds and below
- **Poor** fully loaded 10 seconds and above



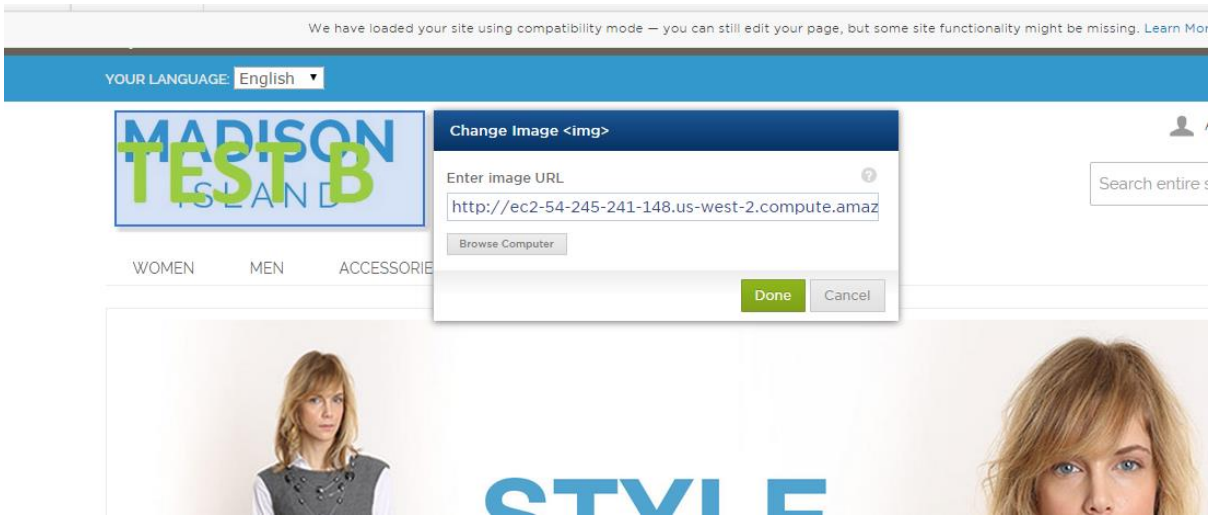
Against these targets most mobile sites with A/B testing tools on them perform poorly. The sites without them are much worse than their desktop counterparts but as before we see a significant decrease in performance of sites with A/B testing installed. With these thresholds none of the sites with A/B testing tools installed performs in the good category for render start, onLoad or fully loaded metrics.

Deep Dive

Taking Optimizely as an example as they have the biggest market share we opened an account and applied some simple changes to the logo on a test e-commerce site.

The first impression was that it was really simple to use, we added the url for our test site (a sample Magento responsive e-commerce site) and started to make changes. 5 minutes later we had added the JavaScript tag and were ready to go.

The screen below shows how easy it is to click on an element and make a change, in this example we were changing the main logo.

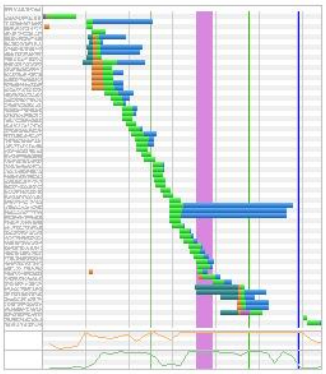


Once you have saved your variations you are told the JavaScript to include on the page.

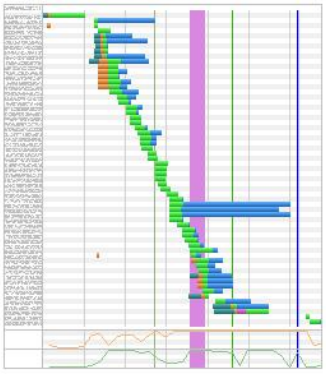
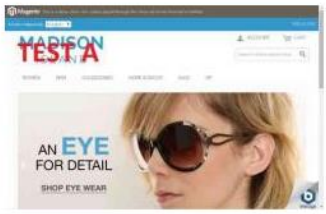
```
44 <script src="//cdn.optimizely.com/js/2425160074.js"></script>
```

We then ran this page through Web Page Test with a 9 test run. Within the results we can see Test A and Test B results as expected.

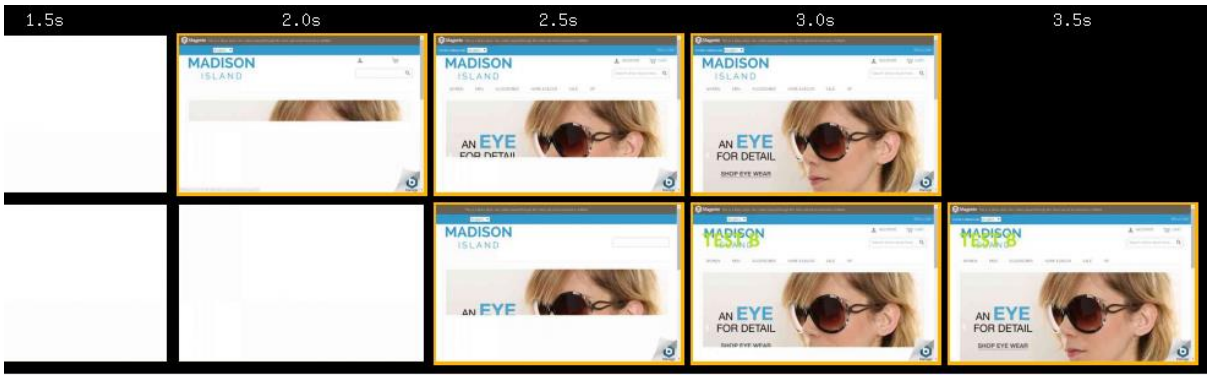
Run 3:

	Waterfall	Screen Shot	Video
First View (2.964s)			Filmstrip View - Watch Video

Run 4:

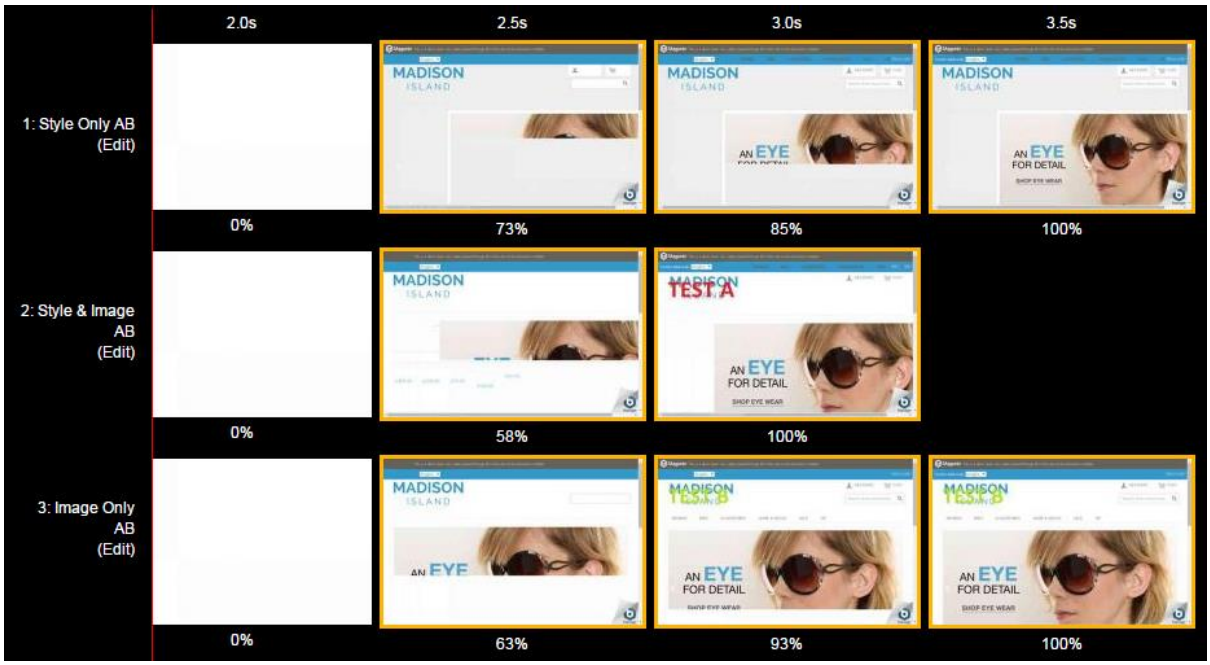
	Waterfall	Screen Shot	Video
First View (3.085s)			Filmstrip View - Watch Video

The filmstrip at the top shows the site with no A/B testing and the filmstrip at the bottom is our simple test where the logo was swapped out for one of the same size.

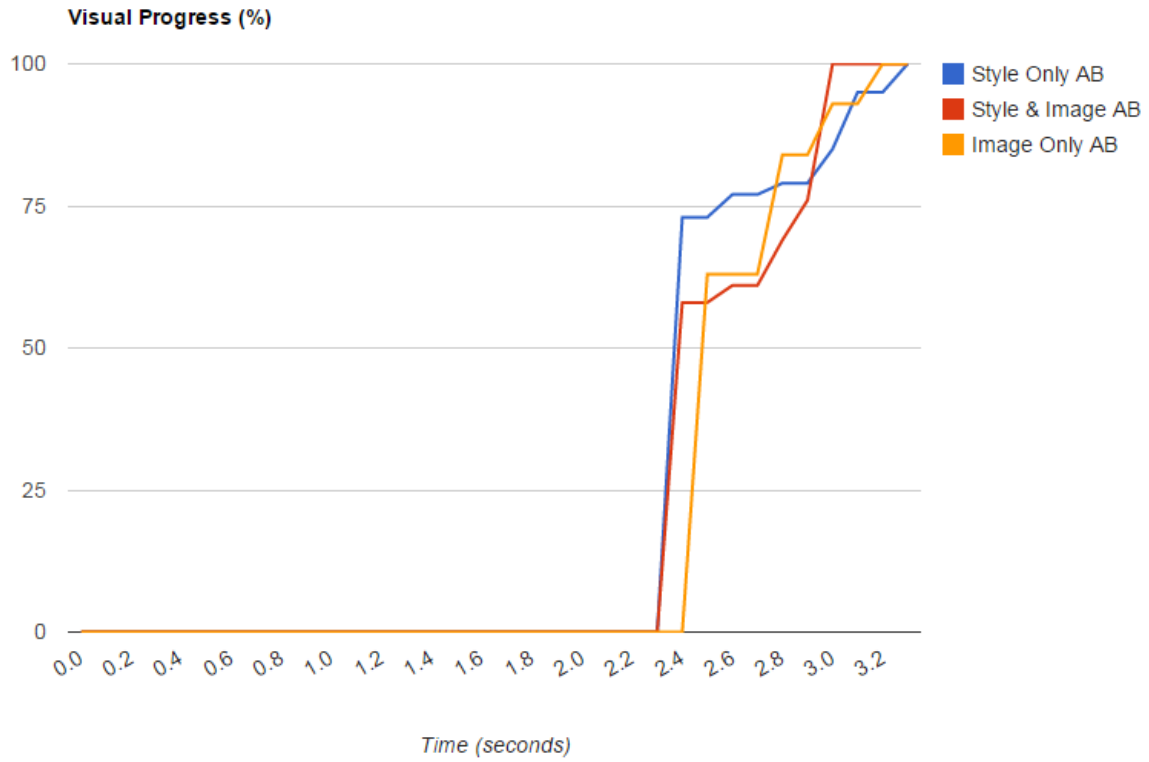


Taking the time to first byte (TTFB) into consideration the render start and onLoad of the page were delayed by about half a second for this page.

We then went and tried some different combinations of features to see if they affected performance. We added some style changes, resizing changes and moving changes, together with combinations of swapping an image out and not.



The performance differences were negligible, even when we tried to rewrite most of the elements on the page, so there doesn't seem to be too much danger of introducing further delays by making the experiment more complex. What is altered though and should also be considered is the rendering pattern. The Speed index graph below shows subtle differences between each test.



In addition to this it is important to make sure if you are making changes that the performance of A and B are similar, especially if you are swapping out images, make sure they are of a similar size.

Conclusions

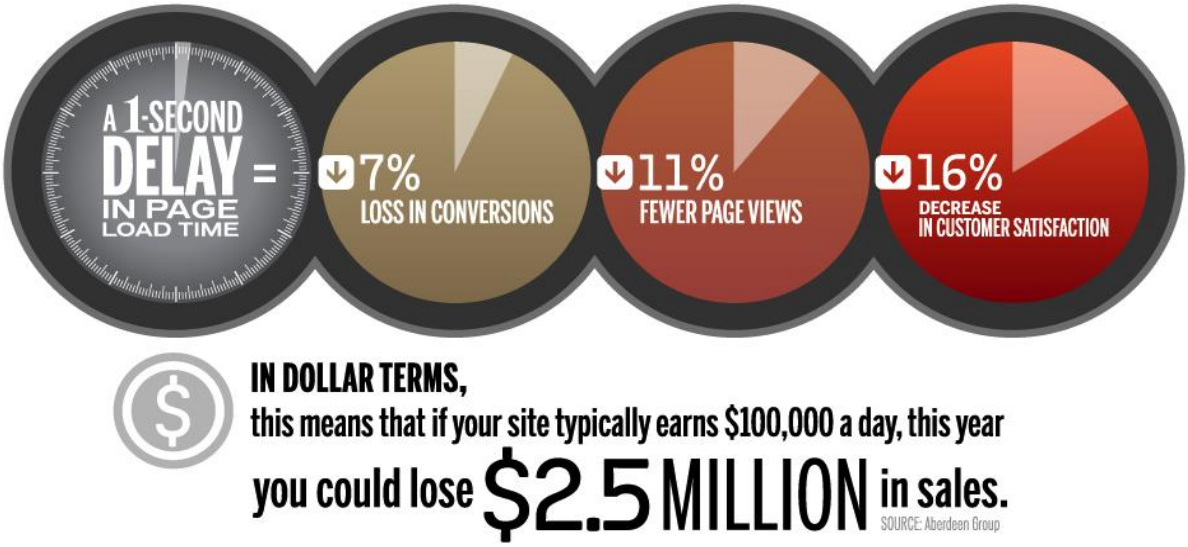
If you do decide to use an A/B testing tool then there are some key considerations to bear in mind when selecting a tool and implementing the solution (known as creating the experiments). After using one of these client side tools it is easy to see why they are popular, they make it very easy for non-technical people to make changes to a webpage without having to involve a potentially busy technical team. The only involvement you will need from your developers is to place the tag on the page, however, trying to ease even this small burden can lead to an issue highlighted below.

A/B testing v's Performance

One of the biggest problems faced today is trying to put a value on performance. Despite many studies showing direct correlations between performance and conversion rates many web sites choose not to take on board performance best practice. Yet many of those same sites are willing to use A/B experiments to see how a website can give the best user experience and increase conversions and revenue.

Using a client side A/B testing tool will decrease site performance, so if you manage to determine that a repositioned button increases conversion by 1%, what has the cost been in decreased performance while running that experiment?

The Aberdeen group ran a study that showed 1 second delay in page load resulted in a 7% decrease in conversion, so unless your experiment yields better results you are potentially in the red before you even start. The data shown earlier shows that sites with AB testing tools on them added at least 1 second to all metrics, so this should be taken into account.



Data source: The Aberdeen Group

To try and understand the business impact of performance on your own site you could add a 1 second delay using a JavaScript timer and run an experiment on this, using the normal tracking to correlate conversion. This would give you a clear indication of what the value of performance is for your web site.

Server side AB testing

If the decision is made that A/B testing has to be used and you don't want to introduce the performance delay then the alternative is to split traffic at the server, proxy or load balancer. This does require more development activity upfront, however if you decide to use result A rather than B the good news is the code or change already exists. With client side testing you will need to get the change made and fit it into a release after the experiment, so the net technical cost may end up being the same.

To try and calculate value and return the development cost should be incorporated as well as any potential lost revenue caused by a decreased performance during the experiment.

At Akamai we are currently working on a solution that allows you to split traffic near the end user at the edge of our network, giving you even more performance gains compared to making that decision at the origin server, which can potentially be a long distance from your end user.

Turn off when not using

If you are not running any live experiments then best practice would be to remove the tag. Having the tag on a page introduces a delay, so if no experiments are running you will be hurting performance unnecessarily without any gain. There could be a tendency to save developer resource to leave the same tag on the site permanently and change experiments within the A/B interface. It really shouldn't take that long to add a JavaScript tag to a page so this benefit is far outweighed by the performance impact.

Keep experiments short

Because of the performance delay it is important to make experiments as short as possible. Collecting data for a whole week will allow for day to day trends to be addressed. Running experiments for anything more than this shouldn't be necessary.

Keep experiments fair

Check the performance of each variation when creating experiments. If they differ then it may not be a fair test as the performance differentiation may outweigh any experiment variation. This is particularly relevant when testing different images as it is very easy to include one version that is a lot larger than another and this will impact performance. During the testing we tried to compare the difference between running simple experiments and complex experiments but the results seemed to be negligible. The only change we saw was the rendering patterns of the page, however, page load times should be measured for all variations as a best practice before launching the experiment.

Only use one A/B tool

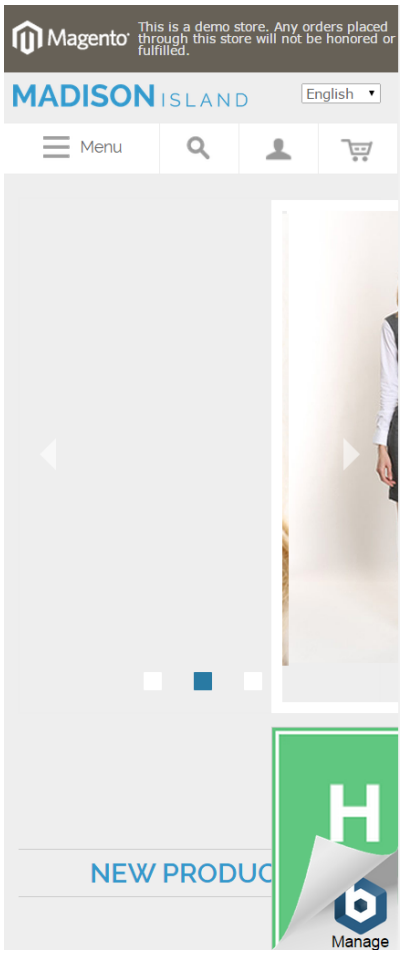
When looking at the data several sites used more than one A/B testing tool on the same page. Whether all the tools had active experiments is not clear but the performance delay introduced by the tools where compounded by having more than one on the same page.

Try not to use on mobile

Mobile data in this report showed the introduction of AB tools on mobile has even bigger effects. Best practice advice would be to try and avoid using on mobile at all.

Beware what else you break

Client side A/B tools use JavaScript to rebuild your page and this can cause breakage of other functions on your page depending on the elements you are applying the experiment to. In our example we used a responsive web site, this was broken by applying an experiment as the resulting A/B experiment used fixed positioning to a design that was predominantly relative. The screenshot of the broken view is shown below:



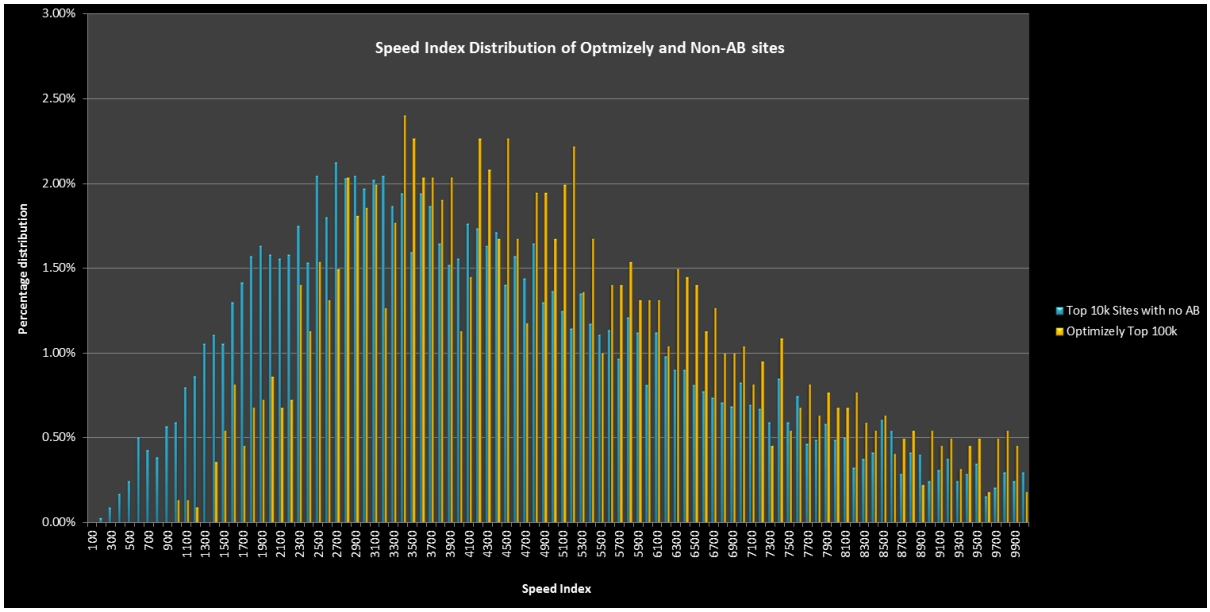
In addition to the layout, there was also a plugin that allowed for zooming of product images that broke after the experiment was applied.

Select your tool wisely

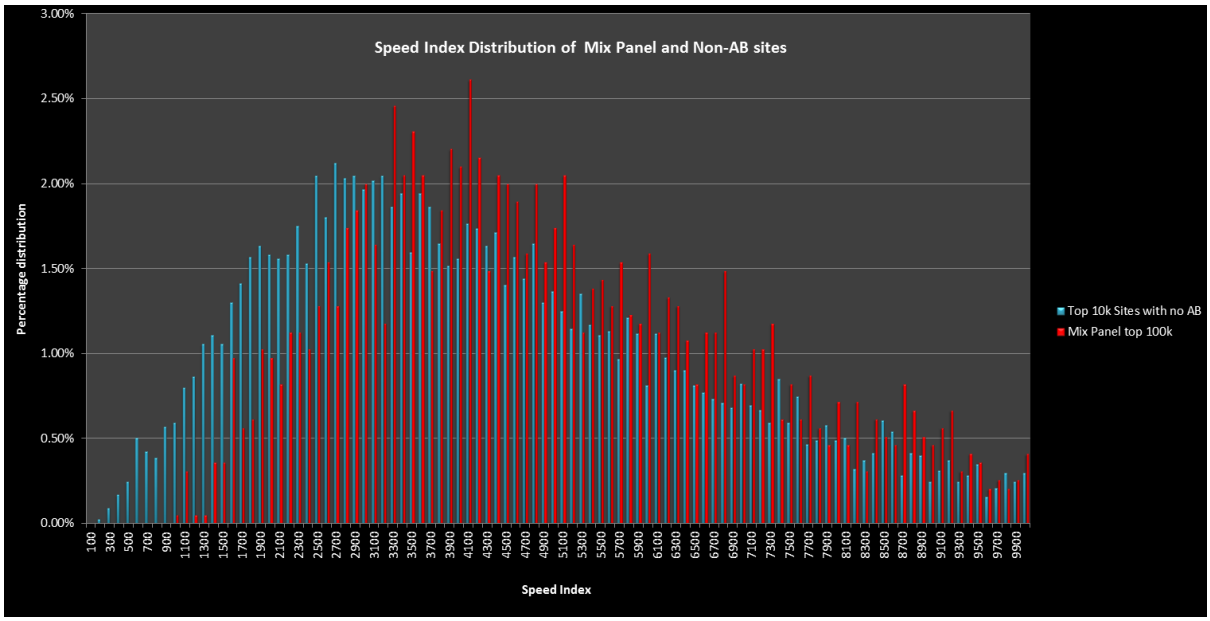
With everything else considered when selecting a tool it is important to find one that is easy to use, however also review the performance of your page after an experiment is applied as it will vary from tool to tool. In addition to this, select a tool that offers high availability, these tags are likely to be hosted by the 3rd party so the performance of the tag itself will be out of your control. Finally make sure the provider uses a CDN because if they don't and they try serving the tag to users a long way away from their server, performance of the tag alone will suffer.

Appendix

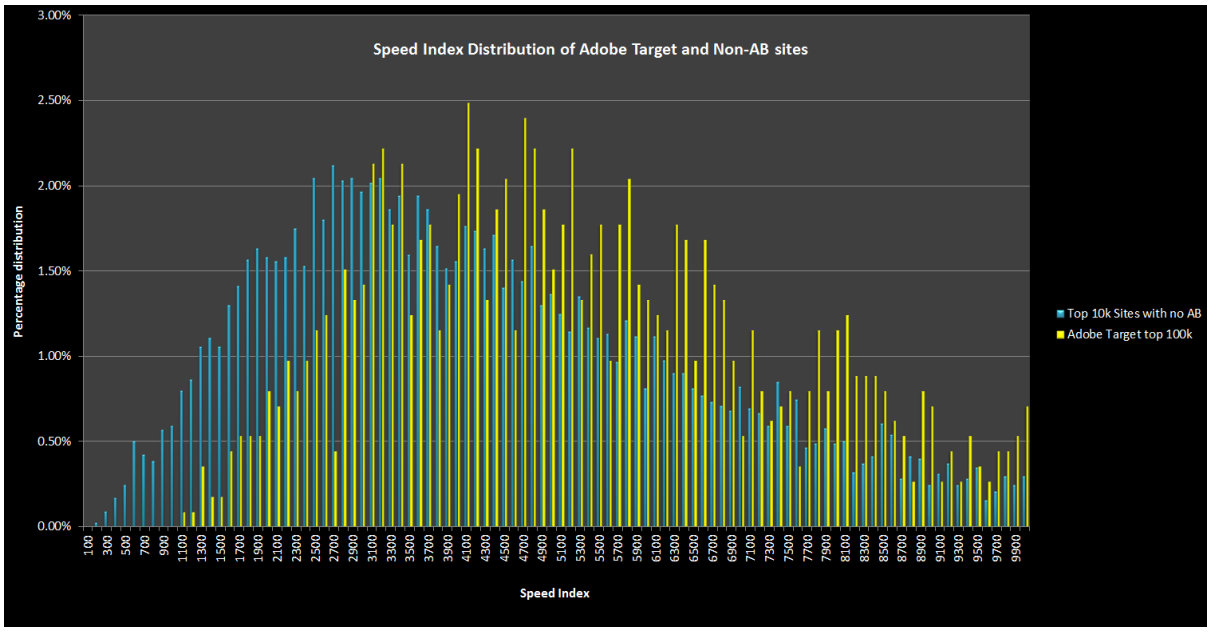
Optmizely Speed Index Distribution



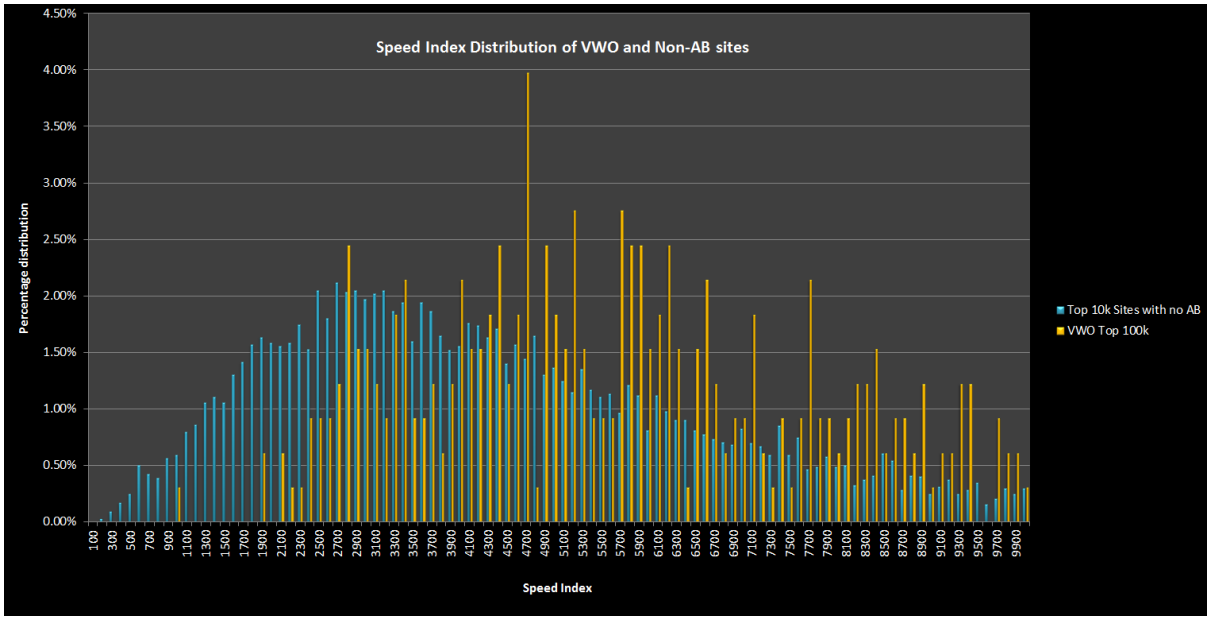
Mix Panel Speed Index Distribution



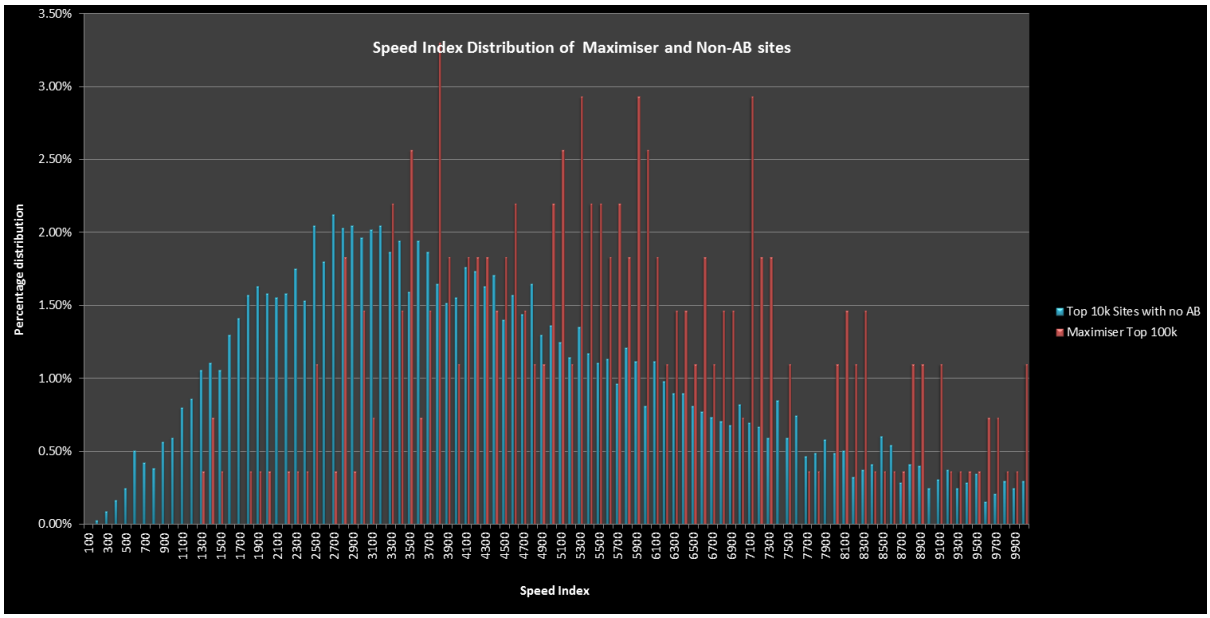
Adobe Target (Omniture) Speed Index Distribution



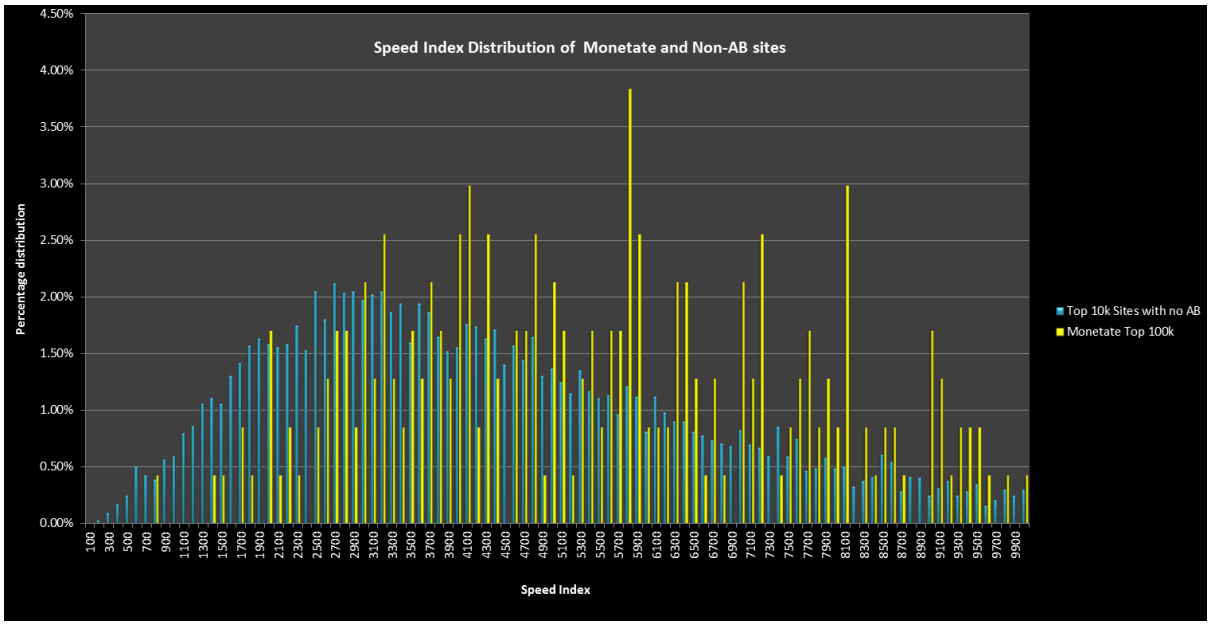
Visual Website Optimizer Speed Index Distribution



Maximiser Speed Index Distribution



Monetate Speed Index Distribution



Akamai® is the leading provider of cloud services for delivering, optimizing and securing online content and business applications. At the core of the company's solutions is the Akamai Intelligent Platform™ providing extensive reach, coupled with unmatched reliability, security, visibility and expertise. Akamai removes the complexities of connecting the increasingly mobile world, supporting 24/7 consumer demand, and enabling enterprises to securely leverage the cloud. To learn more about how Akamai is accelerating the pace of innovation in a hyperconnected world, please visit www.akamai.com and follow @Akamai on Twitter.

Akamai is headquartered in Cambridge, Massachusetts in the United States with operations in more than 40 offices around the world. Our services and renowned customer care are designed to enable businesses to provide an unparalleled Internet experience for their customers worldwide. Addresses, phone numbers and contact information for all locations are listed on www.akamai.com/locations.