Checklist For Improving Your Website Speed & Performance

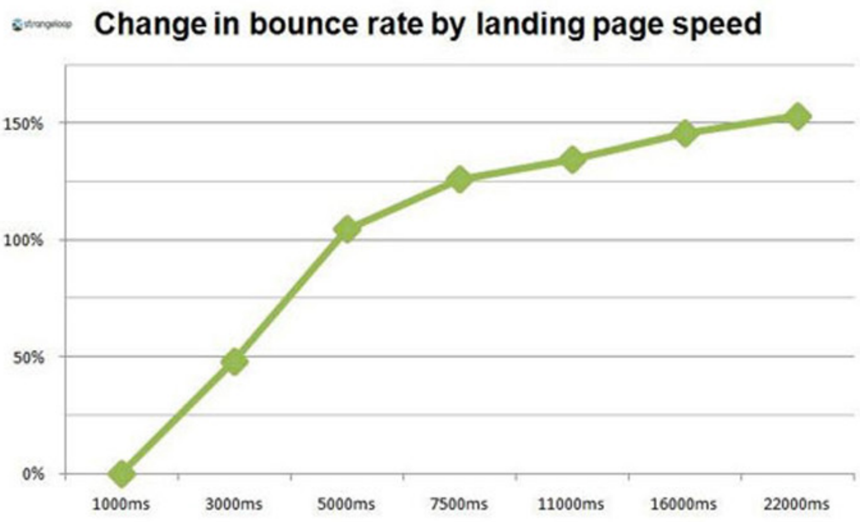
INDUS NET® TECHNOLOGIES

Contents

1 Why is website loading speed important?

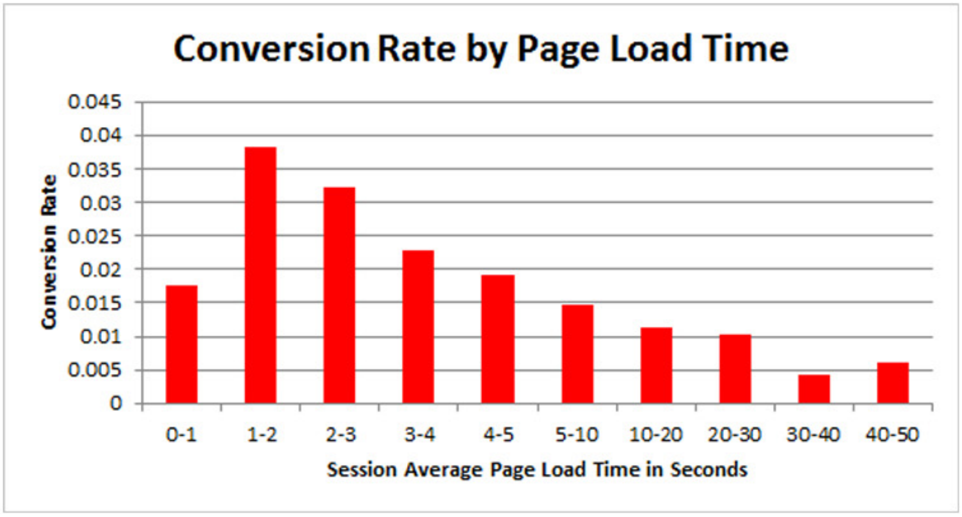
Website loading speed affects the ranking of your website on search engines. Website loading speed affects the speed of indexing pages and number of visitors. Slow websites lead to low conversion rates, which are crucial for the success of any online business.

Let us first look at the bounce rate depending on the loading speed.



Source: http://www.webperformancetoday.com/wpcontent/uploads/2010/06/change_in_bounce_rate_by_landing_page_speed.jpg

Not only does the bounce rate increases but also the chances of conversion



Source: https://assets.econsultancy.com/images/0002/4853/tagmma_image.png

2 How to speed up website?

Speeding up website requires detailed planning, knowledge of different technology areas, adequate resources, with the complete focus on testing and evaluation. In this segment cooperation from several teams: such as web designers, web developers, system administrators, network specialists and other professionals is highly important and required.

Below we present a checklist of things which you should pay attention before launching of a website:

2.1 HTML:

- Minify HTML: The HTML files need to be compressed in order to remove comments, whitespace between tags and unnecessary closing tags (like, <1i>) for reducing the overall file size and speed up the execution and parsing of the web page.

- Enable Gzip Compression: The HTML file should be compressed with gzip in order to reduce asset count and file size.

- Pre-Fetch Assets: The assets should be fetched and downloaded before they are requested by the users.

- Specify A Character Set: A character set in the HTTP response headers of the HTML documentation should be specified in order to allow the browser to begin parsing HTML and executing scripts immediately.

- Clean Code: The website code has to be perfect. Do not use unnecessary elements that will slow down the page.

- Define Character Set: Always define character set in HTML documents which allow browsers to start parsing HTML.

- Remove HTML Comments From The Code: If it is not important or must require then remove the comments. Comments are useful for easier maintenance of websites but they affect speed of loading pages.

2.2 CSS:

- Combine CSS Files: The CSS files should be combines or "concentrate" the CSS files together in order to decrease the number of requests and round trips of the browser has to make, which subsequently decreases the page load time.
- Minify CSS: The CSS files should be compressed or "minify" in order to remove all whitespace and semicolons for the last property of a CSS declaration, which reduces overall file size and decreases the page load time.
- Put CSS In <head>: We should put CSS in the <head>, remove inline style blocks and use <link> CSS files in the <head> section in order to improve browser execution and display load times.
- Load CSS Before JavaScript: We should load external CSS files first before inline JavaScript in the <head>.
- Remove Unused CSS: We should clean code and remove any unused CSS in order to decrease the overall file size and improve browser execution & page load times.
- Use Efficient CSS Selectors: We should avoid using a universal key selector (*) and the use of descendant selectors. We should use class and ID selectors over tag selectors and remove redundant qualifiers. We need to be specific with CSS.
- Enable Gzip Compression: The CSS files should be compressed with gzip in order to reduce asset count and file size.
- Avoid CSS Aimport: We should avoid using CSS 0 import, which increases the number of requests and prevents the browser from being able to download the CSS files in parallel. We should user the traditional <link> tag to ensure parallelized downloads.
- Avoid CSS Expressions: The CSS expressions slows down browser execution. We should use standard CSS properties or JavaScript in Order to improve rendering for IE users.
- CSS: The CSS should be kept in the head part of the document, this helps to achieve that styles is loaded first before rest of the webpage.

2.3 JavaScript:

- Minify JavaScript: The JavaScript files need to be compressed or “minify” in order to remove unnecessary line breaks, extra spaces and indentation, which reduces the overall size of the JavaScript files and increase page load time.
- Load 3rd Party Assets Asynchronously: We should load 3rd party assets asynchronously, so they don’t block important resources from loading, these assets are basically social share widgets and analytics tacking.
- Load JavaScript After CSS: We should load external and inline JavaScript files after CSS in the <head>, JavaScript files are generally larger than the CSS files, which can delay the browser execution and download of other assets.
- Defer Loading Of JavaScript: The defer loaded JavaScript which are not executes onload will reduce the initial download size, allow other resources to be downloaded in parallel and speed up execution and rendering time.
- Defer Parsing Of JavaScript: We should clean code and remove any unused CSS in order to decrease the overall file size and improve browser execution & page load times.
- Enable Gzip Compression: The JavaScript files should be compressed with gzip in order to reduce asset count and file size.
- Use Intelligent Script Loaders For Parallel Processing: We should use script loaders to load assets asynchronously this will bypass the problem of scripts blocking behaviour with parallel processing.
- Avoid Using document.write(): We should avoid using document.write() early in the document in order to prevent slow display and page load times.

2.4 Images:

- Combine Images Using CSS Sprites: We should use CSS sprites to combine many image files into one, this will reduce the number of image requests and overall total page size, this techniques also allows for faster parallelization of assets downloads.

- Compress Images: We should compress the file size of the images. We should manually tune save settings for images in design software or use free online tools that automatically compress the images.

- Specify Image Dimensions: We should specify the image dimensions in the HTML or CSS, this will prevent any reflow the browser has to make by determining image on the fly.

- Serve Scaled Images: We should save and resize an image, when the dimensions are known to us, we should also use scaled images for responsive web designs to serve the smallest file possible.

- Select The Appropriate Image Format: Use JPEG for photographs and PNG format when it is necessary.

2.5 URLs:

- Avoid Bad Requests: We should remove any broken links, missing images or other asset request that result in 404 errors, failing to do so may create a higher number of requests for non-existent assets that slows down the loading of the page.
- Use Domain Sharding For Parallel Processing: We should serve resources from two different hostnames to increase parallel processing, which facilitates more simultaneous downloads than the browser would previously allow.
- Serve Assets From A Single URL: We should serve all the assets and files from a single URL to eliminate any extra overhead with duplicate downloads and extra round trips by the browser.
- Serve Static Content From A Cookieless Domain: We should serve static resources from a cookieless domain in order to reduce the total size of requests made for a page.
- Minimize DNS Lookups: We should reduce the number of unique hostnames from which assets are served to cut down on the number of DNS resolutions.
- High Quality Servers: Speed and server response time are the first parameter on which you should pay attention before launching a website. Explore, test and select the best server option that perfectly fits the needs of your website.
- Optimize Database: A large number of sites uses database, and if we think of complex sites, the database can significantly affect the performance of the site. For better performance it is important to add the index to the database tables. This helps for a faster database response, which leads to the higher website loading speed.
- Use Data URLs: We should use data URLs for smaller images to cut down requests by inlining images into HTML or CSS files. The image will be immediately available when its host document is downloaded.
- Broken Links: Incorrect and broken links will always affect the page loading. Also, broken links are bad for any website.

- External JavaScript & CSS Files: It is advisable to have JavaScript and CSS in external files which can be cached by the browser. Do not use inline styles as it loaded every time you open a webpage.
- Put The Script At The Bottom: The aim of putting scripts at the bottom helps the web components are being loaded before the scripts.
- DNS Lookups: Careful of using different host names in your scripts, objects, images and other components. This makes the websites load longer.

2.6 Caching:

- Use A Content Delivery Network (CDN): We should leverage a CDN cache static assets in data centers around the world and deliver a faster website.
- Use Browser Caching: We should set up an expiry date or a maximum age in the HTTP headers for static resources to direct the browser to load previously downloaded resources from the local disk rather than over the network.
- Make Redirects Cacheable: We should redirect the cache by a user's browser when possible in order to decrease page load times for repeat visitors.
- User Proxy Caching: We should enable public caching in the HTTP headers for static assets in order to allow the browser to download resources from a nearby proxy server rather than from a remote origin server.
- Reduce HTTP Requests: Most of the page load time is depends on downloading individual components. In the time of development we should minimize number of HTTP requests. The following points are techniques you can use to reduce the number of HTTP requests:
  - Combine CSS & JavaScript
  - Empty Image SRC Elements
  - iFrame Elements
  - Photos On A Website
- Enable Caching: Expire header defines how long a particular component can be hold in cache. This helps the return visitors who return to the site will enjoy much higher loading speed.

3 Tools For Analysing Your Website Speed and Performance

- PageSpeed Module: Google's open source module for the web servers which recommend rules for web performance of web pages and related web assets.
  URL: https://developers.google.com/speed/pagespeed/module/

- PageSpeed Insights (by Google): Browser extension, available for Chrome and Firefox that allows you to get performance data of your site and suggestions for how to improve them. Works as part of the developer tools.
  URL: https://developers.google.com/speed/pagespeed/insights/

- GTmetrix: An excellent website to analyze web pages.It offers great insights and covers all the aspect from which a website should be analysed.
  URL: https://gtmetrix.com/

- KeyCDN Website Speed Test: It allows users to test its website speed from 14 locations with an option to make the test public or private.
  URL: https://www.keycdn.com

- Pigndom: It provides analysis of about size analysis, size per domain (compare your CDN assets size vs your domain), the number of requests per domain, and what type of content had the most requests.
  URL: https://tools.pingdom.com/

- WebPageTest: It allows users to check their website on more than 25 browsers (including mobile) and help to diagnose what maybe be a 1st time DNS lookup delay. URL: http://www.webpagetest.org/

- Varvy Pagespeed Optimization: It shares a report with 5 different sections including a resource diagram, css delivery, javascript usage, page speed issues found, and services used. URL: https://varvy.com/pagespeed/

- Uptrends: It offers detailed report on 1st party, statistics, CDN, social, ads, first party overall, and third party overall.
  URL: https://www.uptrends.com/tools/website-speed-test

- Dotcom- Monitor: It allows users to run all geographical tests simultaneously. This saves a lot of time as we generally have to run them individually per location.
  URL: https://www.dotcom-monitor.com/

3 Tools For Analysing Your Website Speed and Performance

- Page Scoring: They have designed a performance report with a minimalist design where they check on Connection Time, Redirection Time, Page Size and Download Time.
  URL: http://www.pagescoring.com/

- Yellow lab tools: It is a tool which provides a view of when JavaScript interactions with the DOM during the loading of the page and other code validation issues.
  URL: http://yellowlab.tools/

- Sucuri Load Time Tester: It shares an interesting insight on "time to first byte", which gives you how long it took for the content to be sent back to the browser to start processing the page.
  URL: https://performance.sucuri.net/

- Pagelocity: The pagelocity analyses on factors like social, SEO, resources, and code. The tool also offers the ability to track your competitors too.
  URL: http://pagelocity.com/

- YSlow: YSlow is an open source project and tool that analyzes web pages and helps you figure out why they are slow based on Yahoo!'s rules for high performance websites.
  URL: http://yslow.org/

- PerfTool: PerfTool is an open source client side performance tool project, hosted on Github. It collects various information about your website and displays it in an easy-to-digest manner on a reports page.
  URL: http://performance-tool.devbridge.com/

Want a Free Website Speed Audit?
Mail us at info@indusnet.co.in

INDUS NET TECHNOLOGIES