

## Improving Search Rank by Optimizing TTFB

Shiv Kumar Goel  
Assistant Prof/Deputy HOD,  
MCA Department,  
VESIT,Chembur,  
Mumbai,India.  
shivkumar.goel@ves.ac.in

Nalawade Pranali .P .B  
MCA Department,  
VESIT,Chembur,  
Mumbai,India.  
pranali8085@gmail.com

**Abstract :** Now a days, everyone is in a great hurry and no one waits for turtles. Studies have shown that the search ranking are affected by how fast the web page loads. The simple logic behind this is “ A website which takes too much time to load provide poor user experience, today's user would quickly shift to next search result and you will end up losing one“. The big search engine giant, Google search uses large number of parameters for determining search rankings which are mostly related to the content on your website, URL , social metrics, quality, quantity, SSL certificates, etc. In 2010, Google added one more factor in search ranking which is websites speed. TTFB (Time To First Byte) provides a clear way to determine how fast/slow there web page loads.

Hence, here I'm focusing on improving website search engine ranking by applying various techniques to optimize TTFB

**Index Terms—** Optimization, Search-Ranking, TTFB.

\*\*\*\*\*

### I. INTRODUCTION

THE he TTFB (Time to First Byte) is the amount of time it takes after the client sends an HTTP request to receive the first byte of the requested resource from the server. In simple terms it's the time it takes for the browser to wait for the first byte of a requested resource to be received *after* the DNS lookup and connection time. The reason it is used as a ranking factor is its easy to measure. According to me it is the most appropriate metric because the browser cannot begin rendering content until it has some initial page markups which mostly consists of references to external resources such as javascripts, CSS, image files, etc. A slow response from the server can be killer. Yet 90% or more of web performance optimization issues relate to front-end architecture and code quality. But that 5-10% related to the back-end can be the most deadly.

Here I will be discussing the ways to optimize wordpress based sites.

### II. TIME TO FIRST BYTE - KEY FACTORS

Time To First Byte includes three key factors

1. Network latency of sending the request to the server
2. Time the web server takes to process the request and generate a response
3. Time the server takes to send the first byte of response to the browser

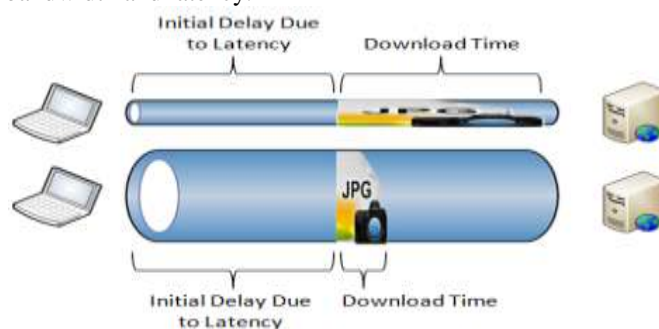
Let's discuss factors for slow server performance:

1. Latency
2. Multiple Processes or Multiple Connections
3. External Resources
4. Inefficient SQL Queries
5. Memory leaks
6. Slow database calls
7. Overloaded Shared Servers
8. Responsive Web Design
9. Demand Peaks

### A. Latency

Latency is the amount of time it takes to transmit a single data packet of data from one location to another. Contrary to what most people believe, having a high bandwidth internet connection does not guarantee low latencies. This is because latency measures how far a packet of data has to travel, regardless of the bandwidth of a user. A data packet will have to travel the same distance to a certain server, whether it travels over a high bandwidth connection or a low bandwidth one.

The figure below helps demonstrate the differences between bandwidth and latency.



Solution: one could use the CDN (Content Delivery Network) to improve latency. CDN helps you to save static data on your site locally on different locations of your choice, which in turn will reduce the latency.

### B. Multiple Processes or Multiple Connections

Programs that keep spawning processes or don't release server connections can bring down your server response time

Solution: rewrite the code to release resources upon completion.

### C. External Resources

Loading JavaScript or other resources from off-site can cause indeterminate delays for your own site. JQuery, advertising

scripts, Google Plus One, Facebook, and other external resources are typically loaded with a remote JavaScript call to Google, Facebook, or some other busy site. Loading infrequently changing resources (like a programming library) remotely introduces DNS and network overhead, and indeterminate delay from a server you don't control.

Solution: Periodically cache the resource locally on your server and serve locally. Use a CRON script to copy the resource.

#### D. *Inefficient SQL Queries*

SQL queries written poorly can slowdown site badly. Databases which are not indexed properly can also bring down the sites performance.

Solution: Always write an optimized queries and use indexing wherever necessary.

#### E. *Memory leaks*

Leaks can accumulate maxing out the memory of your server and make it too slow.

Solution: one way is to increase the memory size, but the best fix would be to debug and locate the memory leak then recode the offending script.

#### F. *Overloaded Shared Servers*

In shared servers, some hosting companies put too many sites on single shared server. In this scenario multiple sites compete with each other for shared resources such as bandwidth, memory, process, etc. If anyone of the shared site is heavily loaded it will shurly have bad effect on rest of the sites.

Solution: It is always possible to purchase dedicated server, but it's could be costly option.

#### G. *Responsive Web Designate*

Now a days, everyone uses mobiles to access sites on the go. For which todays web designs are made responsive to fit users need. But this good feature can in turn be our problem if not used correctly. In case, if media queries CSS is not enqueued properly may cause different variations of same image loaded in background, hence slowing down the load time.

Solution: One should make sure that images and other such resources are managed properly.

#### H. *Demand Peaks*

Sometimes your site might be down only on some specific time during a day or on some hours or so. It may be due to inconsistent website response time caused due to increased amount of load on server slowing down response time.

Solution: Find out the root cause for this high utilization and also which resource in particular gets loaded and then take action accordingly.

## II. OPTIMIZATION TECHNIQUES FOR TTFB

In general TTFB can be improved using following techniques:

### A. *Improving Latency*

Latency can be improved using various approaches one of which is using CDN (already explained). But on should avoid using CDN if you already know that your users are always going to be local to your area.

### B. *Dynamic Content*

Long wait for TTFB is usually caused due to dynamic content on web pages. For generating dynamic contents server has to perform several steps before it can return the page requested by user, leading to delay. On the other hand, when a browser requests a static file, the server just immediately responds with the file. As WordPress pages are dynamic, every time browser makes a request, the pages are built by getting PHP files and interacting with a database. This makes TTFB unreasonably slow. If your latency is WordPress related, you can fix it by providing cached versions of the pages. You can achieve this either by installing a caching plugin or using a WordPress hosting that provides caching.

Using server-side caching is a good way to generate dynamic pages quickly, but if your content does not change frequently, you can utilize local cache. This will greatly improve the page load time.

### C. *Moving Static Content to CDN*

You can use Content Delivery Network (CDN) to reduce the virtual distance between your content and visitors. You can move your static content into a CDN, and they will automatically replicate the content to multiple locations across the world, bringing it geographically closer to the users. CDN can help reduce the TTFB considerably, however, you must ensure that the CDN you employ should be able to cache the static HTML of your home page and not just images, CSS and JavaScript.

### D. *Back-end Infrastructure*

Another factor that can cause a delay in TTFB is the time the web server takes to process the request and generate a response. This processing time is determined by a number of factors that include the computer hardware, an operating system which runs your website, the application code that is running on that hardware, and database queries that the application makes to build the page and the amount of data returned. All these components and their configuration play a critical role in determining TTFB.

You may not have a control over all these issues, and it may not be practically possible to optimize all of them to improve TTFB, but there are certain things you can easily do. Firstly, make sure that you have the required infrastructure to run your website efficiently. Two things that can adversely affect your site performance are shared hosting and virtual hosting. From a cost point of view, shared hosting seems a great idea, but if your goal is to have a fast loading and high ranking site then shared hosting might not work very well for you.

When you share the server resources with other websites, it affects the performance of your site because the processing speed of the server depends on the performance and load of other sites you are sharing the resources with. Alternatively, on-demand hosting systems also increase the TTFB because the server goes onto pause mode when the site has not received any traffic for some time, and it has to run some processes to

resume activity when a user visits the site after a pause. This can have an adverse effect on TTFB.

TTFB Improving techniques for WordPress:

A. *Minimizing Plugins*

The easiest way to lower the load caused due to wordpress plugins by deactivating or deleting the unnecessary plugins. One can use selective deactivation approach to find which plugin is causing higher load and then make decisions to optimize it's code.

B. *Themes*

Just like plugins, themes also needs to be optimized. Themes provides the look and feel to websites so they mostly deal with images, javascript, jQuery, CSS files. These files must be enqueued only where necessary.

It is always a good practice to enqueue js files in footer as they may take time to load and delay the response time for first byte.

C. *Compression*

As mentioned above images, jQuery, CSS, etc are type of static content to your site. It is always good to combine multiple .js and .css files in one for each which reduce the number of requests required.

Compressing this scripts and stylesheets using plugins for minifying or any online minifying site. It helps in redusing the file size.

D. *Offloading*

Using a CDN can greatly reduce the load on your website. Offloading the searching and delivery of images, javascript, css and theme files to a CDN is not only faster but takes great load off your WordPress server's own app stack. A CDN is most effective if used in conjunction with a caching

E. *Caching*

There will always be high amount of static content on any site compared to dynamic. These pages with static content can be cached periodically and users will be served with this cached copies resulting in improved performance.

F. *Optimize Software*

Always use latest versions of softwares on your system because, with every new update they provide some kind of bug fixes and code optimization in it. Running latest versions of softwares will always be helpful.

G. *Use Third Party Hosting*

Using a Managed Hosting solution such as WordPress.com or WPEngine takes advantage of the hard work and expertise of service providers to do the heavy lifting of optimization for you.

### III. CONCLUSION

Most of the site may take high load time due some complex coding, frequent and improperly formatted database queries, geographical location of users, unwanted content, etc on your site.

Keeping only relevant and necessary data and deleting the old data taking regular backups will help in reducing the load. Keeping check on resource utilization and peak hours and

other such factors will help you to optimize your site and according to your need.

### REFERENCES

- [1] [https://en.wikipedia.org/wiki/Time\\_To\\_First\\_Byte](https://en.wikipedia.org/wiki/Time_To_First_Byte)
- [2] <http://www.webpagete>
- [3] <https://www.searchenginejournal.com>
- [4] <https://sites.google.com/a/webpagetest.org/docs/usingwebpagetest/quickstartguide>
- [5] <http://www.webpagetest.org/forums/showthread.php?tid=13520>