In this world of instant gratification, the rate at which your blog delivers content is incredibly important. User satisfaction significantly decreases when sites load slowly and search engines now use page speed as a consideration in search rankings. If your blog takes a long time to load, many users may leave before getting a chance to see your content. Do yourself a favor and use these steps below to attract and keep visitors on your site.

<h2>1. Use browser caching</h2>

Browser caching is a relatively quick and easy way to boost the speed of your site. It works by saving some of your site resources such as CSS or JavaScript files. When the user navigates to another page or returns to your site, they no longer have to download every file. The browser refers to the cached version instead to more quickly display the page instead of downloading it again.

To enable browser caching, set an expiry date in the HTTP headers that instructs to load previously downloaded items from memory. It is recommended that you use browser caching for all static resources, such as images, CSS, Javascript, PDFs, Flash files and other binary object files.

A great benefit of using browser caching is a decrease on the load of your server, which in turn results in decreased bandwidth and hosting costs.

<h2>2. Use a CDN</h2>

Content Delivery Networks function by placing your site (or parts of your site) on super-fast servers. These servers are strategically located around the world. The server closest to the user will deliver the files and content, which results in a faster experience. This means that content is delivered more efficiently because it reduces the number of "hops" before loading content to the visitor.

You can elect to serve up your entire site or specific items such as pictures, videos or downloads on a CDN. Readily available for blogs of all sizes, <a href="http://www.metacdn.com/integrations/wordpress-cdn" target="_blank">WordPress CDN services</a> is a reasonably priced option.

Again, other benefits of using a CDN include reduced hosting and bandwidth costs. Specific to using a CDN, you can protect your blog more effectively against DoS attacks and increased up-time of your site.

<h2>3. Use Google Analytics Page Speed Tool</h2>

While there are many page speed options available, the Google Analytics Page Speed Tool integrates with your current analytics to provide a holistic view of your site. Now, you can correlate page speed and traffic which means that what once were abstract numbers can now be seen as drivers of traffic. It's one thing to see that your site is a bit slow, but when it can be correlated against percentage of exits or bounces, provides extra incentive to take actions to increase site speed.

Use this a href="http://www.simplybusiness.co.uk/microsites/google-analytics-guide/" target="_blank">guide to analytics</a> if don't already have Google Analytics set up. From there, you just need to update the JavaScript code snippet to enable the Page Speed Tool. After your site is technically enabled, you'll need about a few hundred page views in order for analytics to tell an accurate story.

The Page Speed Tool will rank pages from highest latency to lowest latency (slowest pages to fastest pages) and display the following information

- Average Page Load Time
- Pageviews
- Page Load Sample
- Bounce Rate
- Percent Exit

<h2>4. Compress Images</h2>

Pictures take up most of the space of a page file size. By reducing the file size for images, you reduce the amount of information that visitors must download, which can dramatically increase site speed.

Images should first be sized appropriately for their space on your blog post. Then, you compress the image by reducing the quality of the picture. You can use lossy (saving color rich photographs as JPEGs) or loseless (saving text and limited color images as a PNG) compression as well as use other tools to further optimize compression.

<h2>5. Defer Parsing of JavaScript</h2>

When JavaScript is located near the top of your code, browsers must download and process all of the code before loading the rest of the page elements. This means a lot of users see a white screen until the page is loaded completely.

Since JavaScript is mainly used for user triggered events (such as drag and drop, hidden elements expansion, form entry) it is best to defer the parsing until after the page is loaded. You can do this by either putting the code at the bottom or by using a defer attribute in HTML code.