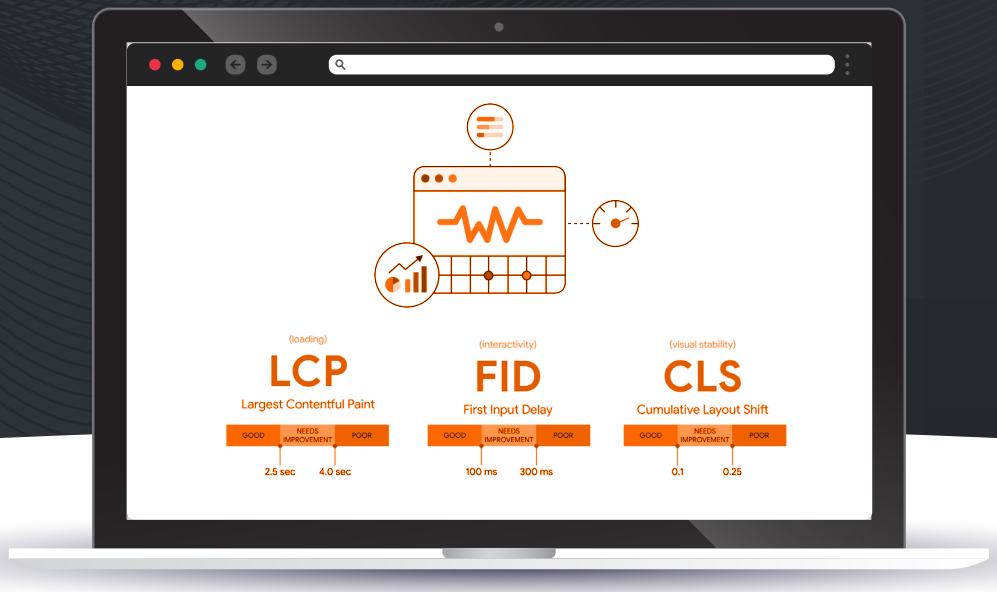
WEBSCALE

CORE WEB VITALS

"Simplified"



Core Web Vitals *"Simplified"*

Core Web Vitals are a hot topic right now, and no more so than in the world of ecommerce. And rightly so. Improving your storefront's Core Web Vitals scores translates into a better user experience for consumers, and Google Search will rank your site higher if you take the time to optimize around these new page experience signals.

However, there is a lot of information, and misinformation, floating around on this topic. Merchants are struggling to find a single source of truth that will guide them on where to focus their optimization efforts.

So, we did it for you. Core Web Vitals *"Simplified"* contains everything you need to know about the new standards, why they matter, how your site's scores can be improved – and how Webscale can help.

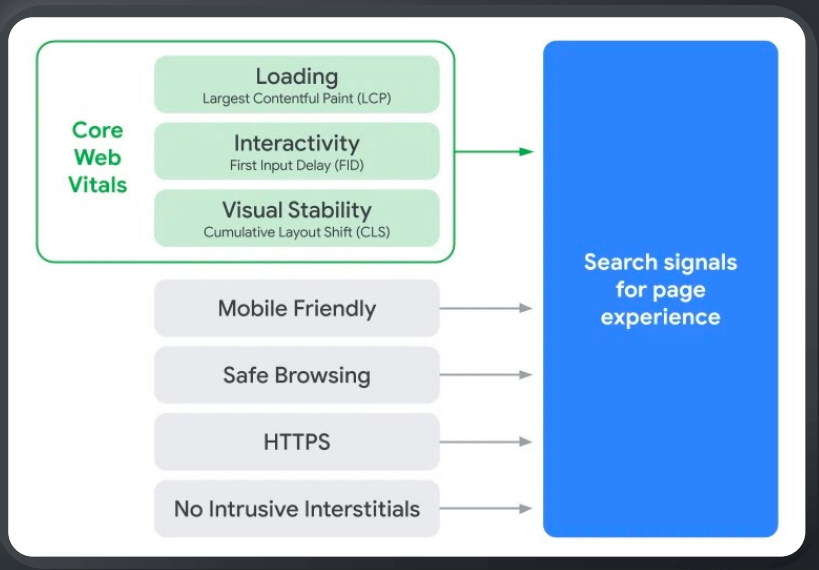
WEBSCALE

What are Core Web Vitals?

Until early 2020, Google measured user experience using four parameters: whether a site is mobile-friendly, offers safe browsing, has enabled HTTPS, and is free of intrusive interstitials (any pop up that significantly obscures the web page). Now, they have added a fifth – Core Web Vitals.

Google launched Web Vitals in May 2020 as an initiative to provide unified guidance for quality signals that are essential to delivering a great user experience on the web.

Core Web Vitals are the subset of Web Vitals, each representing a specific aspect of your users' real world experience when visiting your site. They include the page loading experience, interactivity, and visual stability of page content.



Web Vitals are split into Core Web Vitals and non-Core Web Vitals.

Core Web Vitals

- Largest Contentful Paint (LCP)
- First Input Delay (FID)
- Cumulative Layout Shift (CLS)

Non-Core Web Vitals

- Total Blocking Time (TBT)
- First Contentful Paint (FCP)
- Speed Index (SI)
- Time to Interactive (TTI)

Each metric measures a critical piece of the user experience. The illustration below shows how Web Vitals come into play when a web page loads.

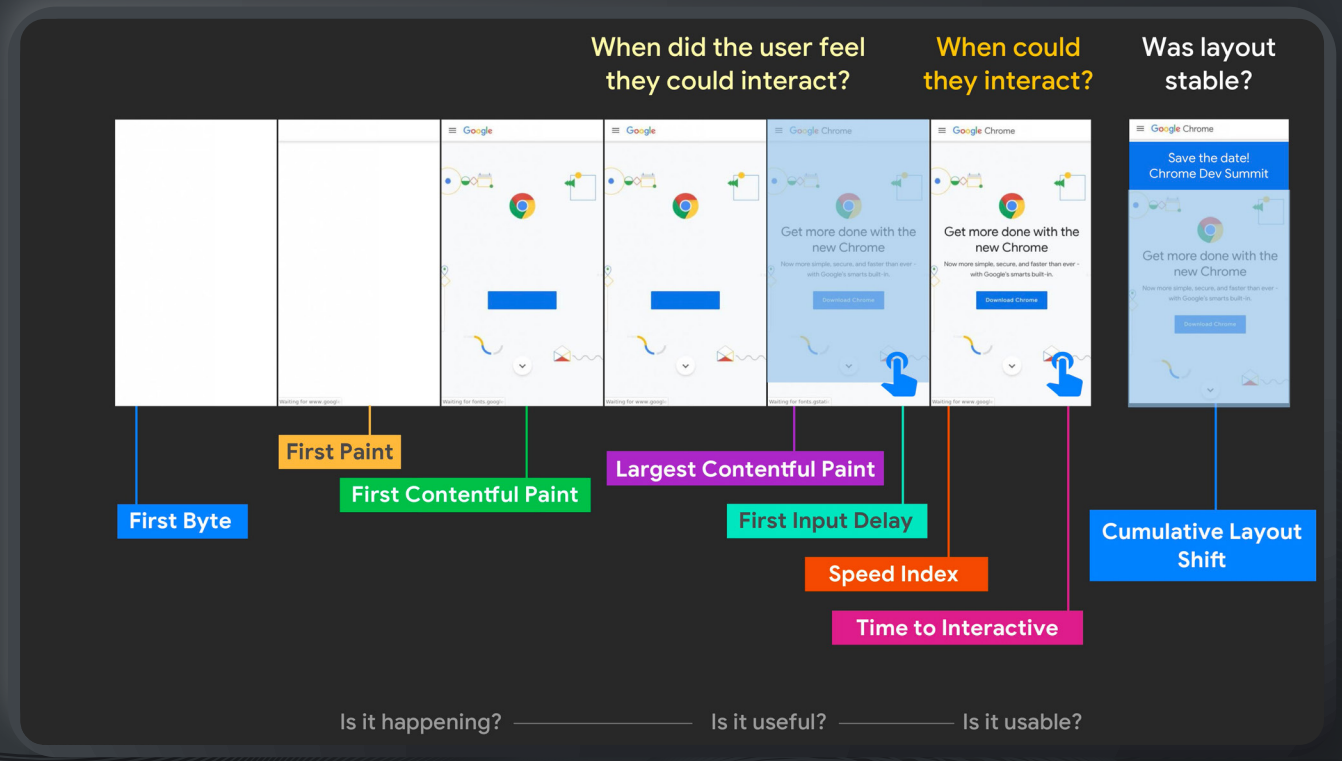


Image credit: Google's Addy Osmani's visualization of Web Vitals

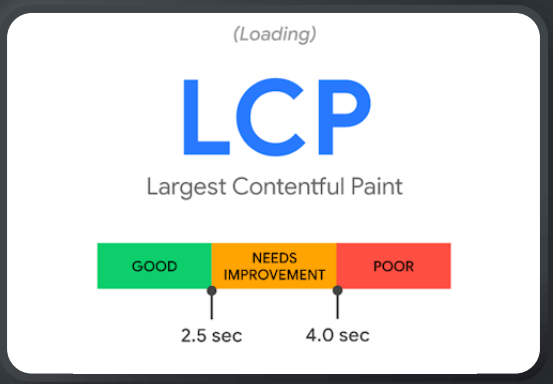
Core Web Vitals

Largest Contentful Paint (LCP)

Largest Contentful Paint (LCP) measures loading performance – the time in seconds from when the page starts loading to when the largest text block, or image element, is rendered on the screen. Its aim is to measure when the page's main contents have finished loading.

The lower the LCP, the better. To provide a good user experience, LCP should occur within 2.5 seconds of when the page first starts loading. LCP is available in both field data and lab data.

Here's how to interpret your LCP score:

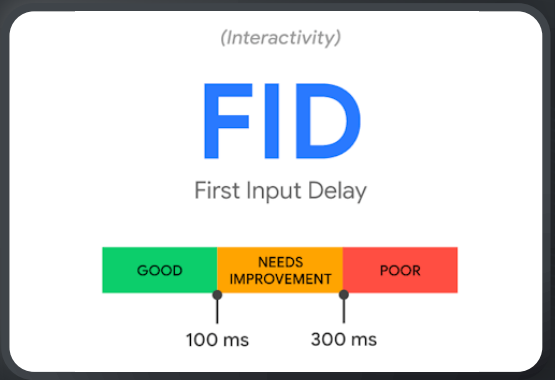


First Input Delay (FID)

First Input Delay (FID) measures interactivity and responsiveness – the time in milliseconds from when a user first interacts with your site (i.e. when they click a link, tap a button, or press a key) to when the browser is able to respond to that interaction.

The lower the FID, the better. To provide a good user experience, pages should have a FID of 100 milliseconds or less. FID is only available in field data. For testing within the lab, the Total Blocking Time metric is used, as it closely correlates with First Input Delay.

Here's how to interpret your FID score:



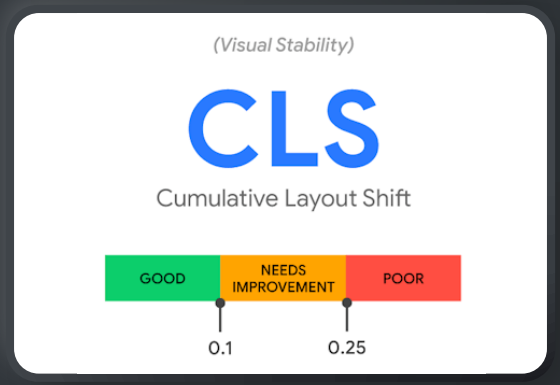
Core Web Vitals

Cumulative Layout Shift (CLS)

Cumulative Layout Shift (CLS) measures visual stability – the cumulative score of all unexpected layout shifts that occur as the page loads making it difficult for users to engage with elements on your site like buttons and links.

The lower the CLS, the better. To provide a good user experience, pages should maintain a CLS of 0.1. or less. CLS is available in both field data and lab data.

Here's how to interpret your CLS score:



To ensure you're hitting the recommended target for most of your users, across each of the above metrics, a good threshold to measure is the 75th percentile of page loads, segmented across mobile and desktop devices.

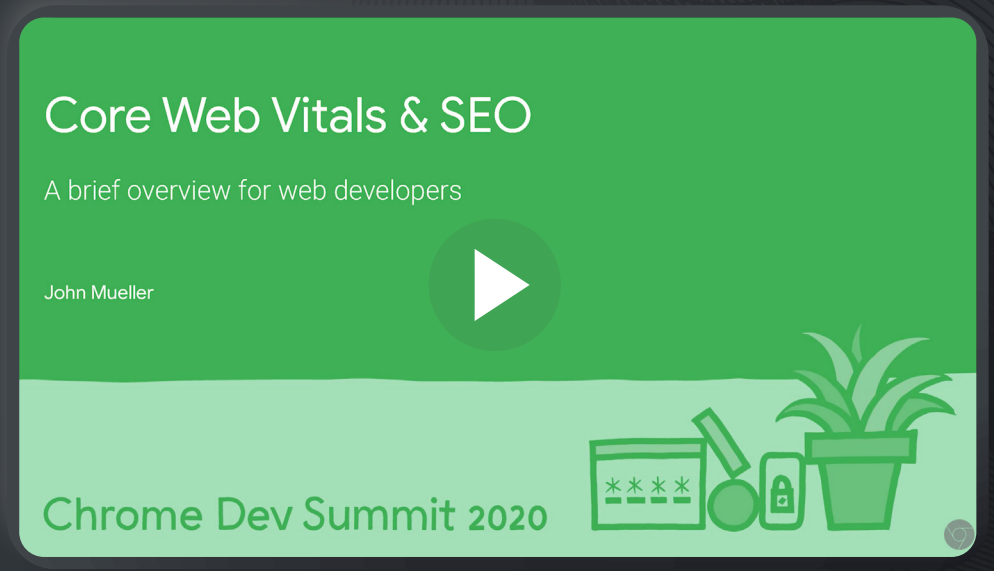
Why do Core Web Vitals Matter?

Core Web Vitals provide you with data on how your website is performing across a wide range of users, and subsequently, a wide range of devices. A faster site delivers a better user experience, and there is plenty of **research** on the direct correlation between user experience, engagement, average cart size and revenue.

Poor Core Web Vitals scores mean low visibility on Google SERP which will heavily impact organic traffic to your website.

Google has also hinted that they may, in the future, start displaying a **"Good Page Experience"** badge for each website in their SERPs, which can also influence brand perception and traffic.

Core Web Vitals and SEO



WEBSCALE

How to Measure Core Web Vitals?

In the Lab

Features under production can only be performance tested in the lab. This is the best way to avoid performance regressions.

In the Field

To truly assess how a site performs for users, the site's performance as those users are loading and interacting with it has to be measured. This type of real-time performance measurement is called Real User Monitoring or RUM. The Chrome User Experience Report (CrUX) provides field data.

In order to pass the Core Web Vitals assessment, the site needs to score "good" (denoted in green color code) for all three Core Web Vitals – Largest Contentful Paint (LCP), First Input Delay (FID) and Cumulative Layout Shift (CLS) – based on field data.

Tools to Measure Core Web Vitals

Google Search Console (GSC)

Core Web Vitals can be measured using Google Search Console, which is the most easy-to-use and reliable tool. However, to use the field data provided by GSC, you need to have a verified property in your Google Search Console account. Alternatively, a RUM tag on your website can collect and publish your scores through a visibility portal, enabling merchants to track their progress with real-time reports so they can take timely action, but we'll share more on this in a **later section.**

Google PageSpeed Insights (PSI)

Significant enhancements have been made to PageSpeed Insights, making it capable of measuring Core Web Vitals. It is also now possible to use the data provided by the Chrome UX Report without any technical or coding expertise.

Google Lighthouse

Lighthouse is an open-source, automated tool for measuring the quality of web pages. It can be run against any web page, public or requiring authentication. Lighthouse audits performance, accessibility and SEO of web pages. The latest version of Lighthouse (6.0, released mid-May 2020) includes additional audits, new metrics, and a newly composed performance score.

Chrome DevTools

The Chrome DevTools Performance panel has a new Experience section that helps webmasters find and resolve visual instabilities that may contribute to Cumulative Layout Shift (CLS).

Chrome UX Report (CrUX)

The Chrome User Experience Report provides user experience metrics for how real-world Chrome users experience popular destinations on the web.

Web Vitals Chrome Extension

This extension measures Core Web Vitals, providing instant feedback on loading, interactivity and layout shift metrics.

Best Practices to Optimize Core Web Vitals

Optimizing Largest Contentful Paint (LCP)

Render your main content faster

One factor contributing to a poor user experience is how long it takes a user to see any content rendered to the screen.

The most common causes of a poor LCP and how to resolve them are detailed below:

Slow server response times

The longer it takes a browser to receive content from the server, the longer it takes to render anything on the screen. A faster server response time directly improves every single page-load metric, including LCP.

Before anything else, improve how and where your server handles your content.

- Optimize your server
- Route users to a nearby CDN
- Cache assets
- Serve HTML pages cache-first
- Establish third-party connections early
- Use signed exchanges

"Why does it take forever to load?"

Render-blocking JavaScript and CSS

Scripts and stylesheets are both render blocking resources which delay FCP, and consequently LCP. Defer any non-critical JavaScript and CSS to speed up the loading of content of your web page.

Reduce CSS blocking time

Ensure that only the minimal amount of necessary CSS is blocking render on your site with the following:

- Minify CSS
- Defer non-critical CSS
- Inline critical CSS

Reduce JavaScript blocking time

Download and serve the minimal amount of necessary JavaScript to users. Reducing the amount of blocking JavaScript results in a faster render, and consequently a better LCP.
This can be accomplished by optimizing your scripts in a few different ways:

- Minify and compress JavaScript files
- Defer unused JavaScript
- Minimize unused polyfills

Slow resource load times

While an increase in CSS or JavaScript blocking time will directly result in worse performance, the time it takes to load many other types of resources can also affect paint times. The types of elements that affect LCP are:

- <img> elements
- <image> elements inside an <svg> element
- <video> elements (the poster image is used to measure LCP)
- An element with a background image loaded via the url ( ) function (as opposed to a CSS gradient)
- Block-level elements containing text nodes or other inline-level text elements

The time it takes to load these elements, if rendered above-the-fold, will have a direct effect on LCP. There are a few ways to ensure these files are loaded as fast as possible:

- Optimize and compress images
- Preload important resources
- Compress text files
- Deliver different assets based on network connection (adaptive serving)
- Cache assets using a service worker

Client-side rendering

Many sites use client-side JavaScript logic to render pages directly in the browser. Frameworks and libraries, like React, Angular, and Vue, have made it easier to build single-page applications that handle different facets of a web page entirely on the client, rather than on the server.

Hence, while building a client-side rendered site, consider the following optimizations:

- Minimize critical JavaScript
- Use pre-rendering

Optimizing First Input Delay (FID)

Respond faster to user interactions

The main cause of a poor FID is heavy JavaScript execution. Optimizing how JavaScript parses, compiles, and executes on your web page will directly reduce FID.

"Why is the page not responding to me?"

The browser cannot respond to most user input while it's executing JavaScript on the main thread. In other words, the browser can't respond to user interactions while the main thread is busy. To improve this:

Break up long tasks

Long tasks are JavaScript execution periods where users may find your UI unresponsive. Any piece of code that blocks the main thread for 50 ms or more can be characterized as a long task. FID should improve noticeably as you adopt best practices like code-splitting and breaking up your long tasks.

Optimize your page for interaction readiness

There are a number of common causes for poor FID scores in web apps that rely heavily on JavaScript, all of which can be fixed.

- First-party script execution can delay interaction readiness
  - Progressive loading of code and features can help spread this work out and improve interaction readiness. Also, consider shifting more logic to the server-side or generating more content statically during build time.
- Data-fetching can impact many aspects of interaction readiness
  - Minimize the reliance on cascading data fetches and how much data needs to be post-processed on the client-side.
- Third-party script execution can delay interaction latency too
  - Explore on-demand loading of third-party code (e.g. maybe don't load those below-the-fold ads until they're scrolled closer to the viewport). Also, prioritize loading what you believe offers the greatest value to users first.

Use a web worker

A blocked main thread is one of the primary causes of input delay. Web workers make it possible to run JavaScript on a background thread. Moving non-UI operations to a separate worker thread can cut down main thread blocking time, and consequently improve FID.

Reduce JavaScript execution time

Limiting the amount of JavaScript on your page reduces the amount of time that the browser needs to spend executing JavaScript code. This speeds up how fast the browser can begin to respond to any user interactions.

To reduce the amount of JavaScript executed on your page:

- Defer unused JavaScript
- Minimize unused polyfills

Optimizing Cumulative Layout Shift (CLS)

Avoid sudden layout shifts to improve user-experience

Imagine you've started reading an article when all of a sudden elements shift around the page, throwing you off and requiring you to find your place again. This is very common on ecommerce sites when trying to click a 'Search' or 'Add to Cart' button, and especially when shopping on a mobile device and using a website that has been optimized to be responsive. Such experiences are visually jarring and frustrating. They're often caused when visible elements are forced to move because another element was suddenly added to the page or resized.

The most common causes of a poor CLS are:

Images without dimensions

Always include width and height size attributes on your images and video elements. Alternatively, reserve the required space with CSS aspect ratio boxes. This approach ensures that the browser can allocate the correct amount of space in the document while the image is loading. Modern browsers now set the default aspect ratio of images based on an image's width and height attributes so it's valuable to set them to prevent layout shifts.

"Why are the page elements dancing?"

Ads, embeds, and iframes without dimensions

Ads are one of the largest contributors to layout shifts on the web. The good news is that it's possible for sites to follow best practices to reduce ad shift. Sites can mitigate these layout shifts by following a few best practices:

- Statically reserve space for the ad slot
  - In other words, style the element before the ad tag library loads
  - If placing ads in the content flow, ensure shifts are eliminated by reserving the slot size. These ads shouldn't cause layout shifts if loaded off-screen
- Take care when placing non-sticky ads near the top of the viewport
  - Position the ad below the main banner and make sure to reserve enough space for the slot
- Avoid collapsing the reserved space if there is no ad returned when the ad slot is visible by showing a placeholder
- Eliminate shifts by reserving the largest possible size for the ad slot
  - This works, but it risks having a blank space if a smaller ad creative fills the slot
- Choose the most likely size for the ad slot based on historical data

Dynamically injected content

Avoid inserting new content above existing content, unless in response to a user interaction. This ensures any layout shifts that occur are expected. If you need to display these types of UI affordances, reserve sufficient space in the viewport for it in advance (for example, using a placeholder or skeleton UI) so that when it loads, it does not cause content in the page to surprisingly shift around.

Web Fonts causing FOIT/FOUT

Downloading and rendering web fonts can cause layout shifts in two ways:

- The fallback font is swapped with a new font (FOUT – flash of unstyled text)
- "Invisible" text is displayed until a new font is rendered (FOIT – flash of invisible text)

There are tools to help minimize these by modifying the rendering behavior of custom fonts, reducing the time it takes to get necessary fonts, or even preloading the fonts giving you a higher chance to meet the first paint, in which case there's no layout shifting.

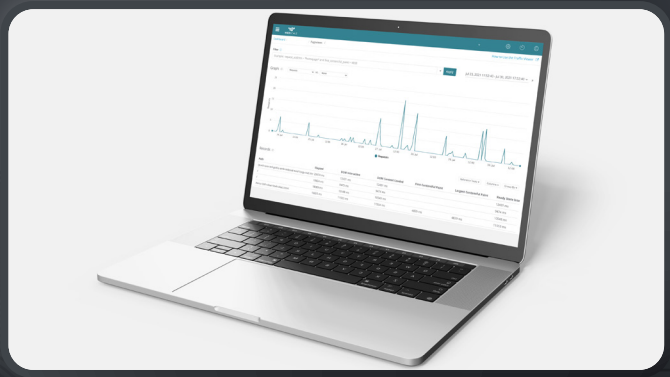
How can Webscale Help?

Webscale's Real User Monitoring (RUM) helps ecommerce merchants get ahead of Core Web Vitals. Our RUM tag measures important page load metrics like LCP, FCP, TTFB (Time To First Byte), Domain Object Model (DOM) Interactive, DOM Content Loaded, Page Load Time, and Ready State Interactive. These metrics are collected and published through the Webscale Portal so merchants can track their progress with real-time reports and take timely action.

Beyond visibility, Webscale technology and products resolve performance challenges and help improve Core Web Vitals scores.

Webscale delivers lightning fast page loads, accelerating page views through end-end performance optimizations across its entire stack, from the internet edge to the Webscale layer and beyond, to the application infrastructure.

We are constantly monitoring and improving all facets of your web application, including intelligent caching to CDNs. **Webscale CloudEDGE CDN** uses intelligent caching, leveraging cloud resources to deliver content from nodes closest to a user, reducing the time it takes for your content to reach their browser.



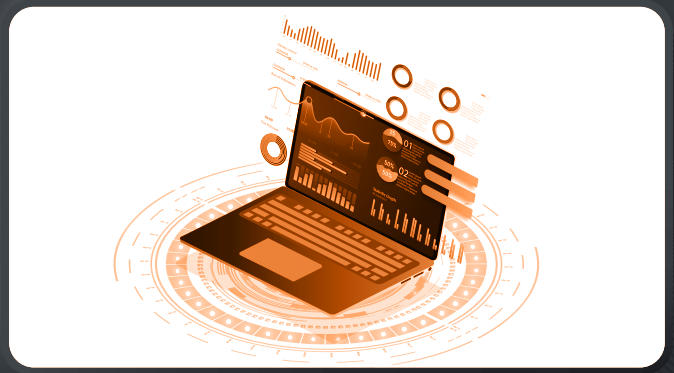
With in-built intelligent caching in its data plane, Webscale uses advanced page and content optimization techniques, in real-time, to optimize web page asset delivery, reduce round trips and page size.

Webscale CloudEDGE Image Manager automates image optimization and management for merchants, ensuring online buyers receive the right image for their specific device, every time, and ideally always from the cache closest to the end user.

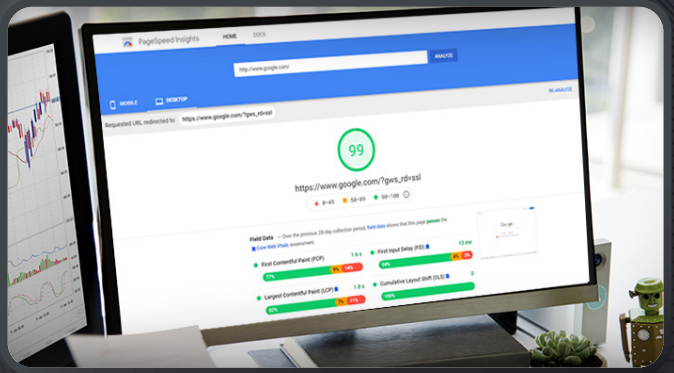
If all the web page assets are as close to the user as possible, then it will reduce their network transfer time. This is the reason why Webscale's 2021 product line up has seen "cloud edge" products addressing application testing, security, bot management and image optimization.

Related Webscale blogs

Webscale's top 4 proven strategies for great Core Web Vitals



Get ahead of Core Web Vitals with (Webscale) Real User Monitoring



Your entire business needs RUM, not just developers

WEBSCALE

Parting Thought

Is going headless a viable path to passing Core Web Vitals?

A popular topic of debate is whether or not switching to a headless commerce infrastructure will give you great Core Web Vitals.

One has to start with the simple fact that brittle, monolithic ecommerce applications are no longer cutting it – the world has moved on and the ecommerce application is breaking up into two parts – the front end and the back end. In this new world, marketing and engineering teams can work at their own pace, with the back end kept more stable with less frequent updates, while the front end, or the user experience piece of your storefront can continually improve and adapt to customer needs. In short, headless commerce infrastructure puts the user experience front and center, and that is a good thing for Core Web Vitals.

Headless architectures, especially when coupled with a PWA deployment for your mobile presence, are redefining how HTML is built, and this change is delivering significant performance improvements. The front end can be deployed anywhere, close to the end user, and if developers are using statically generated pages as an example, your Core Web Vitals will reap the benefits.

But it's not a silver bullet. Projects can take months, especially if you don't have the right team, so choosing the right partners is key. These architectures don't come cheap either, but while the high cost of developing and managing a headless environment, or a one-off PWA build, has historically slowed down their adoption, with your search visibility, organic traffic and sustained business growth at risk, the ROI is now significantly more justified.

Webscale can help here too as we offer tailor-made cloud delivery infrastructure designed around the needs of **headless storefronts and PWAs**, with complete visibility into how the performance and availability are impacting the user experience across multiple devices. Our decoupled architecture has been designed to offer extreme flexibility, up and down the technology stack, and ensure that merchants can maintain seamless user experiences, regardless of their choice of ecommerce platform, cloud provider or toolchain

Useful Resources

- Core Web Vitals & Page Experience FAQs
- Tools to measure Core Web Vitals
- Largest Contentful Paint (LCP) explained
- Optimizing LCP scores
- First Input Delay (FID) explained
- Optimizing FID scores
- Cumulative Layout Shift (CLS) explained
- Optimizing CLS scores

With your search rankings at risk, scoring well in Core Web Vitals is no longer a **recommendation** – it is a critical business **requirement**.

About Webscale

Webscale is the world's only cloud platform for the successful delivery of modern commerce applications. Offering enterprise-grade security, predictive scalability and blazing-fast performance, the Webscale SaaS platform leverages automation and DevOps protocols to simplify the deployment, management and maintenance of infrastructure. The platform supports omni-channel use cases across a variety of ecommerce platforms and architectures, including headless, progressive web applications, self-hosted and fully hosted commerce clouds. Deployed in multi-cloud environments, including Amazon Web Services, Google Cloud Platform, and Microsoft Azure, Webscale powers Fortune 1000 brands including Dollar General, Unilever, Swarovski, Olympus, Regal Cinemas and thousands of other B2C, B2B, and B2E ecommerce storefronts across 12 countries. Webscale has offices in Santa Clara, CA, Boulder, CO, San Antonio, TX, Bangalore, India and London, UK.

Need help?

Reach our ecommerce performance experts at **info@webscale.com**

Published by Webscale Networks, Inc. August 2021.

www.webscale.com

WEBSCALE