

COME

VELOCIZZARE

IL TUO SITO



WORDPRESS

Brian Jackson

**KINSTA**

Hosting WordPress Gestito

# Come Velocizzare il Vostro Sito WordPress

(GUIDA DEFINITIVA)



Per la versione sempre aggiornata di questa guida, fai la scansione del QR code qui sopra, o vai all':

<https://kinsta.com/it/impara/velocizzare-wordpress/>

Pubblicato da Kinsta

Traduzione italiana di Carlo Daniele

Design di Maja Szakadat



## Sull'autore - Brian Jackson

Brian è *Chief Marketing Officer* di Kinsta. Si occupa di tutto, dalle nuove strategie di sviluppo online, alla creazione di contenuti, alla SEO e alla diffusione all'interno della community. Si diverte un sacco a scrivere lunghi articoli ed ha pubblicato oltre 2.500 post.

Una delle più grandi passioni di Brian è WordPress, che ha utilizzato per oltre un decennio. Dopo aver lavorato nel settore della distribuzione di contenuti e ora nel settore dell'hosting ad alte prestazioni, è specializzato nell'ottimizzazione delle performance web e nelle strategie di velocizzazione dei siti web.

Come Velocizzare il Vostro Sito WordPress

# Indice

**10**

Tipologia dei Siti  
WordPress:  
Statici e Dinamici

**13**

Scegliere un  
Hosting WordPress dalle  
Alte Prestazioni

**25**

Scegliere il Server  
Più Vicino ai Propri  
Visitatori

**34**

Un DNS Premium è  
Meglio di un DNS Gratuito

**38**

Il Vostro Tema  
WordPress È  
Importante

**47**

La Verità sui Plugin  
di WordPress

**52**

Impostazioni Ottimali  
di WordPress

**58**

Perché la Cache  
È Così Importante

**68**

L'Ottimizzazione  
delle Immagini  
È un Must

**77**

Ottimizzare  
il Database

**95**

Utilizzare un Content  
Delivery Network (CDN)

**105**

Scaricare Media e Email  
Quando È Necessario

**110**

Come Trovare Strozzature  
e Plugin Lenti

**129**

Raccomandazioni  
sull'Ottimizzazione  
del Back-End

**144**

Consigli  
sull'Ottimizzazione del  
Front-End e sui Servizi  
Esterni

**172**

Ottimizzare Sempre con  
il Mobile-First in Mente



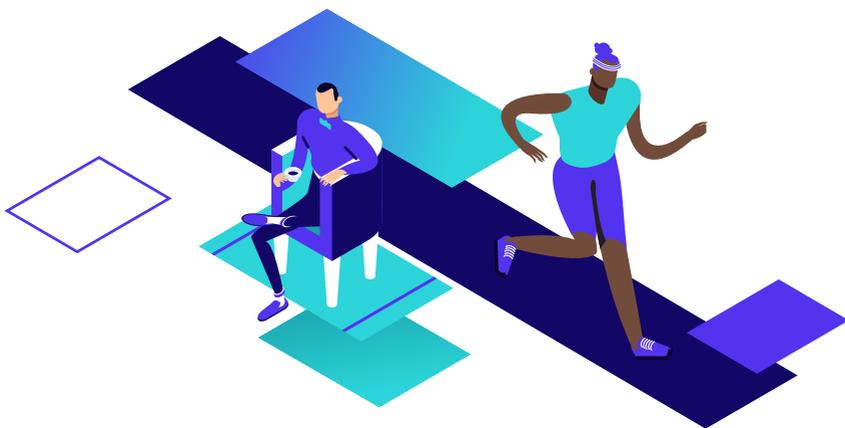
Nel corso degli anni abbiamo pubblicato molti tutorial in cui abbiamo illustrato un gran numero di metodi per ottimizzare e velocizzare WordPress. Ma a volte trovare tutto ciò che serve in un unico posto può essere difficile. Per questo oggi condivideremo con voi tutto ciò che sappiamo su come mettere il turbo a WordPress, dopo oltre 15 anni di esperienza e un gran numero di dure lezioni apprese: tutto in un'unica guida. Se avete appena iniziato a utilizzare WordPress o se siete sviluppatori esperti, vi promettiamo che troverete qualcosa di utile in questa guida!

**Oltre un terzo del web è ora basato su WordPress.** È fantastico, ma bisogna considerare che ci sono migliaia di temi, plugin e tecnologie diverse che devono coesistere. Per l'utente medio di WordPress, tutto questo può trasformarsi rapidamente in un incubo quando il sito inizia a bloccarsi e non si riesce a capire perché o dove cercare per risolvere il problema.

Oggi analizzeremo in profondità le azioni da intraprendere anche subito per ottenere importanti miglioramenti nei vostri siti WordPress. Condivideremo con voi anche alcune risorse che per noi sono state preziose.

CAPITOLO 01:

# Tipologia dei Siti WordPress: Statici e Dinamici



**Prima di analizzare le ottimizzazioni, è importante innanzitutto capire che non tutti i siti WordPress sono uguali. Questo è il motivo per cui molti utenti si trovano nei guai, dal momento che non è possibile affrontare tutti i problemi allo stesso modo. Classifichiamo sempre i siti WordPress in una delle due categorie: statici o dinamici. Quindi, la prima cosa che faremo sarà quella di analizzare le differenze tra questi due tipi di siti.**

## **Siti Prevalentemente Statici**

Sono statici, in genere, siti come blog, siti di piccole attività economiche, siti di notizie a volume ridotto, siti personali, di fotografia, ecc. Per statico intendiamo che i dati presenti in questi siti WordPress **non cambiano molto spesso** (forse un paio di volte al giorno). Anche la gran parte del nostro sito Kinsta potrebbe essere considerato un sito web statico.

Questo è incredibilmente importante in quanto molte delle richieste possono essere servite direttamente dalla cache sul server a velocità incredibili! Non preoccupatevi: approfondiremo l'argomento del caching nel dettaglio più avanti. Per adesso diciamo solo che questo significa che ci saranno meno chiamate al database e non saranno necessarie tante risorse per raggiungere le performance di Google.

## Siti Altamente Dinamici

Dall'altro lato abbiamo siti altamente dinamici. Questi comprendono siti come gli eCommerce (WooCommerce o Easy Digital Downloads), community, siti a iscrizione, forum (bbPress o BuddyPress) e sistemi di gestione dell'apprendimento (LMS o Learning Management Systems). Con il termine dinamico intendiamo che i dati su questi siti WordPress **cambiano frequentemente** (le transazioni dei server avvengono con una frequenza di pochi minuti o anche ogni secondo). Questo significa che non tutte le richieste che arrivano al server possono essere servite direttamente dalla cache e richiedono risorse aggiuntive del server e un maggior numero di query sul database.

Questi siti hanno in genere **un numero elevato di visitatori contemporanei** e di sessioni simultanee. Su un sito WordPress a carattere informativo o aziendale, che è per lo più statico, un visitatore potrebbe rimanere per cinque o dieci minuti finché non trova quello di cui ha bisogno (è una stima approssimativa, di solito le frequenze di rimbalzo sono molto più alte). Su siti dinamici accade il contrario. I visitatori di solito arrivano sul sito per interagire con qualcosa o qualcuno. Se stanno seguendo un corso online, non è insolito che ci restino per delle ore.

Potete intuire questo dove ci porta. I visitatori simultanei connessi al vostro host WordPress aumentano velocemente. Per complicare le cose, avete un gran numero di visitatori contemporanei con un problema di “contenuto non memorizzabile nella cache”.

*Non è possibile trattare tutti i siti WordPress allo stesso modo per quel che riguarda le prestazioni. I siti statici e i siti altamente dinamici sono due animali molto diversi.*

CAPITOLO 02:

# Scegliere un Hosting WordPress dalle Alte Prestazioni



Un host WordPress è un'azienda che tiene in archivio tutti i dati del vostro sito web. Sottoscrivete un piano e tutte le vostre immagini, i contenuti, i video, ecc. vengono memorizzati su un server che si trova nel data center dell'host. L'host di WordPress vi offre un modo semplice per accedere ai dati, gestirli e indirizzarli verso i vostri visitatori. Abbastanza semplice, vero? Beh, non proprio.

Nel Web incontrerete tre tipi molto diversi di host WordPress. Analizziamo i pro e i contro di ciascuno di questi tipi. È importante scegliere quello giusto fin dall'inizio, altrimenti vi troverete solo con dei gran mal di testa e molto tempo perso.

## 1. Hosting WordPress Condiviso

Il primo e più popolare tipo di hosting WordPress è quello che chiamiamo "hosting condiviso". In questa categoria sono compresi i più grandi host del settore, come le società EIG tipo Bluehost e HostGator, nonché provider come Siteground, GoDaddy e InMotion Hosting. Generalmente utilizzano cPanel e il cliente medio paga tra i 3 e i 25 dollari al mese.

Chiunque utilizzi questo tipo di hosting a un certo punto sperimenterà lentezza nel sito, è solo una questione di tempo. Perché? Perché **gli host condivisi tendono a sovraccaricare i server**, il che può influire sulle prestazioni del sito. Sospensioni del

sito o errori 500 si verificano di frequente in quanto gli host devono porre dei limiti un po' su tutto, e devono consolidare le risorse per sopravvivere. Nel caso peggiore, vi troverete di fronte a tempi di inattività. Anche se non lo sapete, il vostro sito WordPress è probabilmente seduto sullo stesso server di altre 200 e più persone. Eventuali problemi di altri siti possono ricadere sul vostro.



Hosting WordPress condiviso

Non importa come fate i conti, tolte le spese, \$3 al mese non generano entrate per la società di hosting. Soprattutto quando ci aggiungete il servizio di supporto. Basta che aprite un ticket di supporto e sono già andati in rosso. Il modo in cui guadagnano è dato da tariffe aggiuntive e commissioni nascoste. Queste tariffe aggiuntive includono migrazioni, registrazioni di domini, certificati SSL, ecc. Un'altra tattica comune è quella di fornire enormi sconti di iscrizione. Ma una volta che arriva il momento di rinnovare, ecco che avete il conto effettivo.

La maggior parte di questi host offre quello che chiamano un piano a "risorse illimitate". Probabilmente l'avete visto tutti. Bene, nel mondo reale non esiste una cosa come le risorse illimitate. Dietro le quinte, gli host strozzano i clienti che consumano molte risorse. Questo, a sua volta, si conclude con clienti arrabbiati che vanno via, lasciando spazio ad altri clienti che non usano molte risorse. Alla fine, si avrà un circolo vizioso in cui la società di hosting spinge piani economici e iscrizioni di clienti che sperano non utilizzino molte risorse e invece acquistino servizi aggiuntivi.

Con l'hosting condiviso, il servizio clienti e l'assistenza sono quasi sempre scadenti, per l'enorme quantità di siti rispetto agli addetti al servizio di supporto. Gli host condivisi devono tenersi molto leggeri per ottenere un profitto anche minimo, e questo di solito porta a un'esperienza spiacevole per il cliente.

*Quando si tratta di hosting  
condiviso, di solito si ottiene quello  
per cui si paga.*

## 2. Hosting VPS WordPress Fai Da Te

Il secondo tipo di hosting di WordPress è il VPS fai-da-te, ossia il "Fai da te su un server virtuale privato". Questa tipologia è in genere adottata da startup bootstrap e utenti con maggiore esperienza di sviluppo, di gestione server e di WordPress. Sono il gruppo dei fai-da-te. Queste persone in genere cercano di risparmiare denaro, ma di solito si preoccupano anche delle prestazioni e si rendono conto della loro importanza per il successo della loro attività. Le impostazioni più comuni potrebbero includere l'utilizzo di un provider VPS di terze parti come Digital Ocean, Linode o Vultr; insieme a strumenti come ServerPilot per una gestione più semplice.

Un piccolo VPS di DigitalOcean parte da \$5 al mese, e il piano più economico di ServerPilot parte da \$10 al mese. Quindi, a seconda della configurazione, potreste trovarvi di fronte ad un costo compreso tra i 5 e i 15 dollari o più al mese. L'approccio fai-da-te può ridurre i costi, ma significa anche che voi siete responsabili se si rompe qualcosa e per l'ottimizzazione delle prestazioni del server.

L'approccio fai-da-te può essere ottimo, ma può anche ritorcersi contro di voi se non state attenti. Non prendete questa strada se non siete esperti di tecnologia o solo perché volete smanettare! Il vostro tempo è denaro e dovrete spenderlo per far crescere la vostra attività.

### 3. Hosting WordPress Gestito

Il terzo tipo di hosting è quello che offriamo da Kinsta e cioè l'hosting WordPress Gestito. Questi tipi di host gestiscono al vostro posto tutte le attività di back-end relative al server, oltre a fornire supporto quando ne avete bisogno. Di solito, sono ottimizzati per funzionare con WordPress e in genere includono funzionalità come ambienti di gestione temporanea ad un clic e backup automatici. I loro team di supporto saranno più preparati quando bisogna conoscere il modo in cui aggirare il CMS, poiché sono impegnati ogni giorno sulla stessa piattaforma.

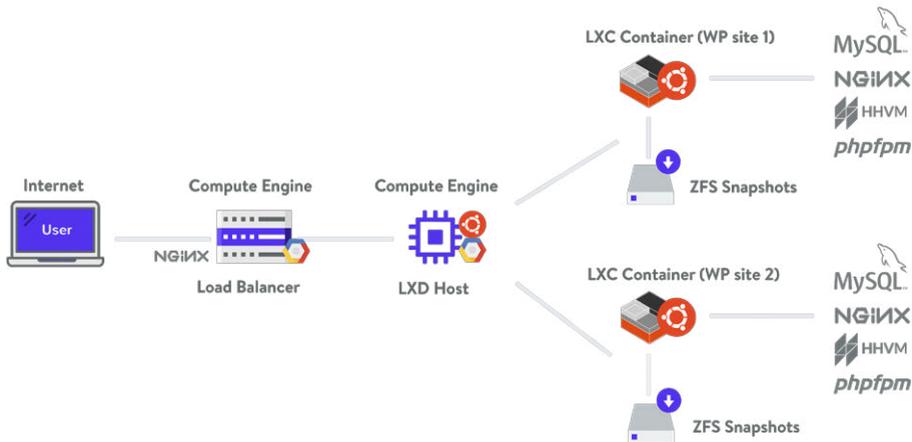
***Se volete risparmiare tempo, l'hosting WordPress gestito è la soluzione giusta!***

I piani di hosting gestito di WordPress vanno in genere da \$25 a \$150 al mese o più, a seconda delle dimensioni del sito e delle esigenze. Grandi aziende come jQuery, Intuit, Plesk, Dyn, NGINX e persino la Casa Bianca utilizzano WordPress per la gestione del proprio sito web. Alcuni popolari host WordPress gestito con cui probabilmente avete familiarità, o che forse utilizzate in questo momento, comprendono WP Engine, Flywheel, Pressable, Media Temple, Pressidium e Pagely.

### Kinsta Ha un Approccio Diverso

Ma Kinsta porta l'hosting WordPress gestito ad un livello superiore. La nostra piattaforma di hosting non rientra nelle consuete categorie di hosting. La nostra intera infrastruttura è costruita su Google Cloud Platform ed è diversa dalla tradizionale infrastruttura condivisa, VPS o dedicata.

Ogni sito WordPress sulla nostra piattaforma viene eseguito in un container software isolato, che contiene tutte le risorse software necessarie per l'esecuzione del sito (Linux, NGINX, PHP, MySQL). Ciò significa che il software che esegue ciascun sito è completamente privato e non è messo in condivisione nemmeno tra i vostri stessi siti.



Architettura di Kinsta

Il container di ogni sito viene eseguito su macchine virtuali in uno dei molteplici data center GC. Ogni macchina ha fino a 96 CPU e centinaia di GB di RAM. Le risorse hardware (RAM/CPU) vengono assegnate automaticamente a ciascun container dalle nostre macchine virtuali in base alle necessità (una funzionalità pulita che definiamo auto-scaling).

Ogni anno Review Signal pubblica i suoi benchmark sulle performance dell'hosting WordPress, e siamo orgogliosi che per cinque anni di fila Kinsta abbia dimostrato di essere la migliore azienda su tutti i livelli! E non solo su uno o due dei nostri piani, ma su ogni piano, da Starter fino a Enterprise.

“Kinsta ha registrato test LoadStorm e Blitz sostanzialmente perfetti. In più, non sono stati riscontrati problemi in nessun altro test. Non ho parole sufficienti per elogiare la loro prestazione.”



Kevin Ohashi  
Fondatore e consulente WP, ReviewSignal

Inoltre, non abbiamo addetti al supporto di livello 1 o livello 2. Il nostro intero team di supporto è composto da sviluppatori WordPress e tecnici specializzati nell'hosting Linux, molti dei quali hanno gestito i propri server, hanno creato

temi e plugin e contribuito al core. Ciò vi assicura di ricevere i consigli esperti di qualcuno che utilizza attivamente e sviluppa con WordPress.

Potete chattare con gli stessi membri del team di supporto che assistono clienti aziendali e Fortune 500. Siamo così esigenti riguardo alla qualità del nostro team di supporto che assumiamo meno dell'1% dei candidati. Non troverete un supporto migliore da nessun'altra parte!

Per saperne di più sul perché dovrete scegliere Kinsta per l'hosting WordPress gestito, leggete la pagina perché noi – in cosa Kinsta è diverso. Ma, a prescindere da chi scegliate come hosting provider, dovrete sempre cercare le funzionalità del server riportate di seguito, per essere certi che il vostro sito web sia eseguito il più velocemente possibile.

“Con WP Engine, i problemi semplici vengono solitamente risolti rapidamente. Tuttavia, per qualsiasi problema complesso, una risoluzione richiederà un po' di tempo e ci saranno un sacco di comunicazioni in entrambe le direzioni. Questo è un problema quando si esegue un sito WordPress di fascia alta ed avete un problema urgente da risolvere in fretta. Se mi chiedete un consiglio tra i due, a mio avviso, Kinsta è migliore. Offrono molto più di quanto promettano. Non dovete mai preoccuparvi della lentezza del sito, dei tempi di inattività, del ricevere assistenza qualificata o di altri problemi relativi all'hosting.”



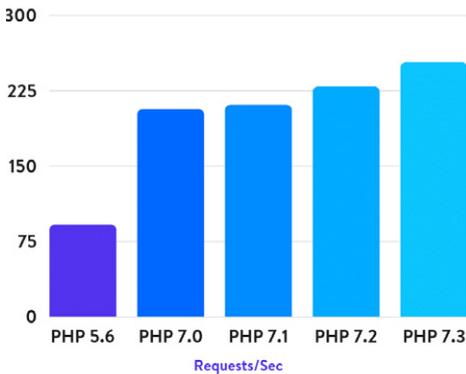
Harsh Agrawal  
Blogger premiato, ShoutMeLoud

## PHP 7 o Versioni Superiori per Prestazioni Ottimali

PHP è un linguaggio di scripting e programmazione open source lato server, utilizzato principalmente per lo sviluppo web. La maggior parte del software core di WordPress è scritto in PHP, così come i temi e i plugin, il che rende il linguaggio PHP molto importante per la comunità di WordPress. Dovreste assicurarvi che il vostro host WordPress disponga almeno di PHP 7 o versioni superiori.

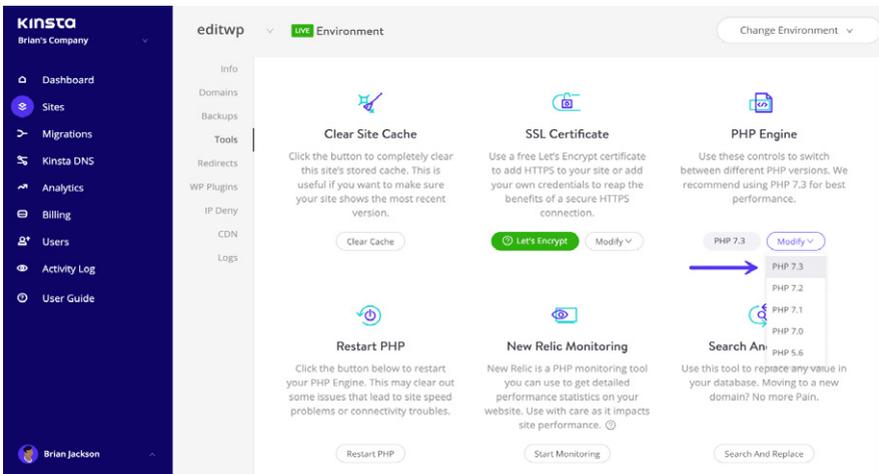
Ci sono diverse versioni di PHP che il vostro host vi metterà a disposizione sul vostro server, e il più recente PHP 7.3 offre enormi miglioramenti delle prestazioni.

### WordPress 5.0



Infatti, come risulta dai nostri recenti benchmark PHP, se si confronta PHP 7.3 con PHP 5.6, emerge che PHP 7.3 può gestire un numero 3 volte superiore di richieste (transazioni) al secondo! PHP 7.3 è anche in media il 9% più veloce di PHP 7.2. Ciò può anche influire sulla reattività del pannello di amministrazione di WordPress.

Velocità più elevate e sicurezza migliorata sono il motivo per cui Kinsta offre sempre le versioni più recenti di PHP. Potete cambiare la versione di PHP con un solo clic.



E fate attenzione agli host di WordPress che offrono HHVM come alternativa a PHP. HHVM non è più una soluzione adatta per l'hosting di WordPress in quanto non supporta più ufficialmente PHP.

# Scegliete un Host che Utilizza **NGINX**

Dietro le quinte, ogni host WordPress utilizza un server web per alimentare i vostri siti WordPress. Le scelte più comuni sono NGINX e Apache.

Raccomandiamo vivamente di scegliere un host che utilizzi NGINX perché è radicato nell'ottimizzazione delle prestazioni su scala. Nei test di benchmark, NGINX supera spesso in prestazioni altri server web molto diffusi, specialmente in situazioni con contenuto statico o numerose richieste simultanee. Questo è il motivo per cui Kinsta usa NGINX.

Alcune aziende di alto profilo che utilizzano NGINX sono Autodesk, Atlassian, Intuit, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, Salesforce, VMWare, Xerox, LinkedIn, Cisco, Facebook, Target, Citrix Systems, Twitter, Apple, Intel e molti altri. (fonte: Sifter.com)

Secondo W3Techs, Apache alimenta il 44,0% di tutti i siti Web, e ciò lo rende la soluzione più utilizzata. Ma se si guarda al server web più popolare tra i siti web ad alto traffico (i primi 10.000), NGINX ne alimenta il 41,9%, mentre Apache ne alimenta solo il 18,1%. È utilizzato da alcuni dei siti più ricchi di risorse esistenti, tra cui Netflix, NASA e persino WordPress.com.

## Il Network dell'Host È Importante

Quando scegliete un host WordPress, potrebbe non venirci in mente di chiedere o cercare di sapere quale network utilizza, ma è una informazione necessaria. Il network può avere un enorme impatto sulle prestazioni del vostro sito e persino sulla reattività della dashboard di WordPress. Molti host lasceranno il network fuori dalle loro scelte di marketing, e opteranno per la rete più economica per tagliare i costi.

Ecco alcune domande che dovrete porre:

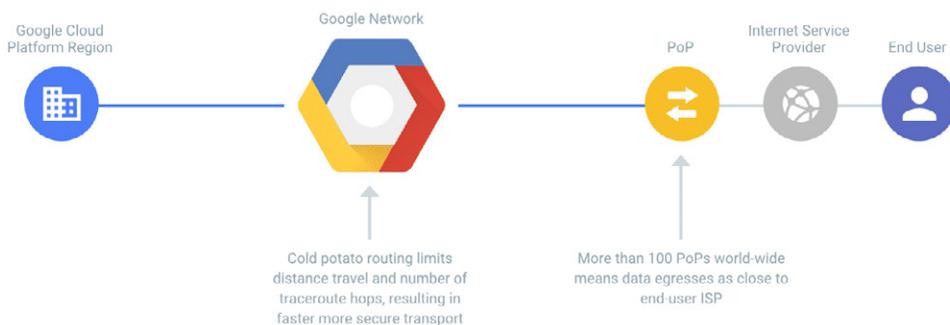
- **Su quali reti trasmettete i dati?** La maggior parte dei dati è su reti ISP pubbliche o su infrastrutture private come Google o Microsoft? Questi grandi provider dispongono di network che sono costruiti e ottimizzati per avere velocità e bassa latenza. Hanno persino i propri cavi internet sotto l'oceano!

- **Le reti che utilizzate sono ridondanti?** Cosa succede se un cavo viene tagliato accidentalmente? Cose del genere succedono più spesso di quanto pensiate.

Già nel 2017 Google aveva presentato il proprio network di livello standard, che è una rete più lenta ma ha un costo inferiore. Da Kinsta utilizziamo il **network di livello premium** per tutti i nostri piani di hosting. Sebbene costituisca per noi un costo extra, vi garantisce velocità eccezionali.

***Kinsta utilizza il network di livello premium di Google Cloud Platform. Perché i vostri siti web meritano il meglio.***

Secondo Google, il network di livello premium raggiunge prestazioni di rete superiori riducendo la durata della trasmissione sulla internet pubblica; i pacchetti entrano (ed escono) dal network di Google il più vicino possibile all'utente e quindi viaggiano lungo la dorsale di Google prima di arrivare alla VM. Il livello standard, invece, consegna il traffico in uscita dalla GCP alla internet tramite reti pubbliche di transito (ISP) invece che tramite il network di Google.



Network Livello Premium di GCP

Per dirla in modo più comprensibile:

- **I pacchetti di livello Premium trascorrono più tempo nella rete di Google,** con meno rimbalzi e quindi con prestazioni superiori (ma costano di più).
- **I pacchetti di livello standard passano meno tempo nella rete di Google e più**

tempo a giocare a ping pong sulle reti pubbliche, e quindi, hanno prestazioni inferiori (ma costano meno).

Che conseguenze ha tutto questo? Bene, per i dati che viaggiano attraverso i continenti, il network di livello premium è circa il **41% più veloce**, in media, rispetto alla rete di livello standard. Per i dati che viaggiano verso una regione vicina (stesso continente), il livello premium è circa dell' **8% più veloce**. Mentre il networking rappresenta solo una frazione dei tempi di caricamento totali della pagina, ogni millisecondo si somma agli altri!

*Per i dati che viaggiano attraverso i continenti, la rete premium di Google Cloud è in media il 41% più veloce!*

Anche la ridondanza è determinante, ed è per questo che Google utilizza almeno tre percorsi indipendenti (ridondanza N+2) tra due posizioni sulla rete Google, contribuendo alla garanzia che il traffico continui a circolare tra le località anche in caso di interruzione.

Come potreste pensare a questo punto, molto succede dietro le quinte per quel che riguarda il networking. Assicuratevi che il vostro host WordPress utilizzi una rete affidabile e che non opti per i livelli inferiori per ridurre i costi.

## HTTP/2 È un Must

HTTP/2 è un protocollo Web rilasciato nel 2015 che è stato progettato per accelerare la modalità di consegna dei siti web. Per essere supportato dai browser, richiede HTTPS (SSL). Se il vostro host WordPress non supporta HTTP/2, dovreste iniziare a cercare un nuovo provider. Con lo spostamento dell'intero Web su HTTPS, questa non è più solo una bella funzionalità; è una necessità.

Il miglioramento delle prestazioni apportato da HTTP/2 è dovuto a diversi fattori, come il supporto del multiplexing, il parallelismo, la compressione HPACK con codifica

Huffman, l'estensione ALPN e server push. In passato c'era un sovraccarico di TLS quando bisognava eseguire HTTPS, ma ora questo si verifica molto meno grazie a HTTP/2 e TLS 1.3. Kinsta supporta HTTP/2 e TLS 1.3 su tutti i server e CDN.

Un altro grande vantaggio di HTTP/2 è che con la maggior parte dei siti WordPress non è più necessario preoccuparsi della concatenazione (combinazione di file) o dello sharding del dominio. Sono ora ottimizzazioni obsolete. E HTTP/3 è prossimo!

CAPITOLO 03:

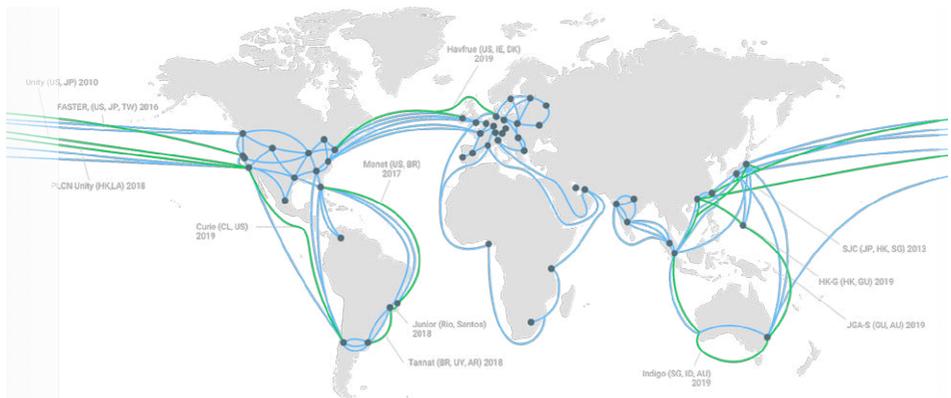
# Scegliere il Server Più Vicino ai Propri Visitatori



**Una delle prime cose che dovrete fare quando scegliete l'hosting del vostro sito WordPress è determinare da dove arriva la maggior parte dei vostri visitatori o clienti. Perché è importante? Perché la posizione in cui si colloca il sito web svolge un ruolo significativo nel determinare la latenza complessiva della rete e il TTFB. Incide anche sulla velocità SFTP e sulla reattività del pannello di amministrazione di WordPress.**

Latenza di rete: si riferisce al tempo e/o al ritardo della trasmissione di dati su una rete. In altre parole, quanto tempo impiega un pacchetto di dati per passare da un punto all'altro. Al giorno d'oggi questo valore viene in genere misurato in millisecondi; tuttavia, potrebbero essere secondi, a seconda della rete. Più si avvicina a zero, meglio è.

In altre parole, maggiore è la distanza, maggiore è la latenza, che equivale a un ritardo maggiore. Non potete battere la velocità della luce quando si tratta di dati che viaggiano su cavi in fibra ottica. Ad esempio, di seguito è riportata una mappa della rete di Google Cloud Platform. Hanno migliaia di chilometri di cavi sotto l'oceano che attraversano l'intero globo. Questa è una delle ragioni per cui è una delle reti più veloci al mondo. E anche con questa estesa rete all'avanguardia, ci saranno sempre ritardi dovuti alla latenza.



Ci sono alcuni fattori che contribuiscono ad aumentare i ritardi dovuti alla latenza della rete. Tra questi:

- **Ritardo di trasmissione:** diversi tipi di supporti, come le connessioni wireless o in fibra ottica, producono un certo ritardo in quanto possono spingere fuori moltissimi bit. Ad esempio, una connessione in fibra ottica avrà un ritardo di trasmissione inferiore rispetto a una linea T1. Dovete anche considerare la dimensione del pacchetto.
- **Ritardo di propagazione:** è il tempo necessario alla trasmissione del pacchetto dati. Su questo possono incidere distanza, ritardi di rete, ecc.
- **Ritardo di accodamento:** se viene superata la larghezza di banda, possono verificarsi ritardi di accodamento per cui i dati devono attendere presso l'host o il router. Su questo può incidere la congestione della rete.

Alcuni potrebbero sostenere che la latenza della rete non è molto importante; tuttavia, questo può dipendere da dove si sceglie di ospitare il proprio sito WordPress. Per i media e per le risorse (immagini, JavaScript, CSS, video), un CDN può ridurre di molto la latenza in eccesso, consegnando una copia da un server più vicino. Tuttavia, nella maggior parte delle configurazioni, è sempre necessario interrogare il server di hosting per il caricamento del DOC iniziale. E questo è il motivo per cui la latenza è importante!

La latenza è particolarmente importante quando si tratta di aziende che servono i visitatori in una specifica posizione geografica. Ad esempio, supponiamo che

abbiate un negozio di e-commerce a Sydney e che il 90% dei vostri clienti provenga dall'Australia. La vostra attività trarrebbe vantaggio dalla collocazione del vostro sito su un server in Australia, invece che in Europa o negli Stati Uniti.

**TTFB:** indica il tempo del primo byte. In parole semplici, è una misura di quanto tempo il browser deve attendere prima di ricevere il suo primo byte di dati dal server. Più tempo è necessario per ricevere questi dati, più tempo sarà necessario per visualizzare la pagina. Di nuovo, più siamo vicini a zero, meglio è.

Un malinteso frequente è che il TTFB viene calcolato dopo i tempi di risoluzione del DNS. Tuttavia, il calcolo originale del TTFB nel networking comprende sempre la latenza di rete. Ciò comporta un processo in 3 fasi e ritardi e latenza possono verificarsi ovunque, sommandosi al totale del TTFB.

Connection Setup		TIME
Queueing		4.56 ms
Stalled		1.35 ms
Request/Response		TIME
Request sent		0.11 ms
Waiting (TTFB)		291.39 ms
Content Download		1.47 ms
<b>Explanation</b>		<b>298.87 ms</b>

**1. Richiesta al Server:** Quando qualcuno visita il vostro sito web, per prima cosa viene inviata una richiesta HTTP dal client (browser) al server. In questo passaggio, ci sono svariati fattori che potrebbero generare ritardi. Tempi di risoluzione del DNS prolungati potrebbero contribuire ad accrescere il tempo necessario per la richiesta. Se il server si trova geograficamente lontano, può aumentare la latenza a causa della distanza che i dati devono percorrere. Inoltre, se avete regole firewall complesse, potrebbero aumentare il tempo di instradamento. E non dimenticate la velocità di connessione a internet del client.

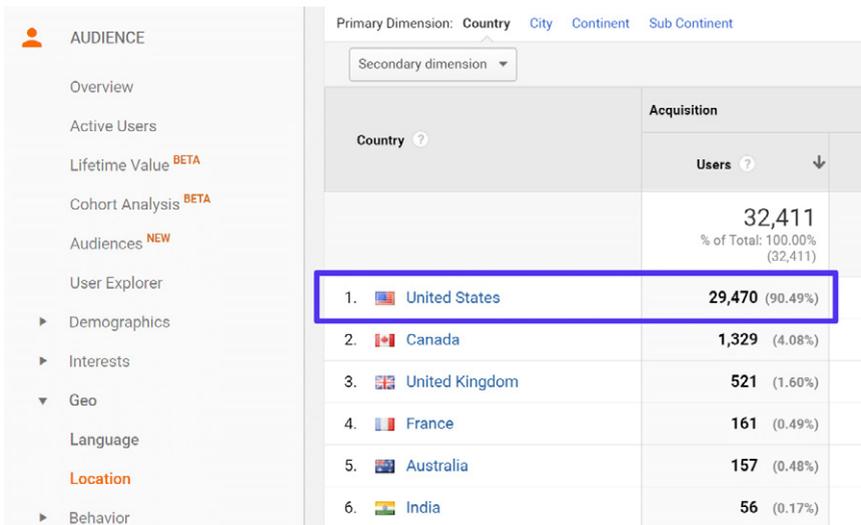
**2. Elaborazione del Server:** Dopo che la richiesta è stata inviata, il server deve elaborarla e generare una risposta. Ciò potrebbe aggiungere diversi ritardi dovuti a chiamate lente al database, eccessivi script di terze parti, la mancata memorizzazione nella cache della prima risposta, codice o tema WordPress non ottimizzati e risorse server, come l'I/O del disco o la memoria, inefficienti.

**3. Risposta al Client:** Quando il server ha elaborato la richiesta, deve rispedirla al client (o piuttosto restituire il primo byte). Ciò è fortemente influenzato dalla velocità della rete del server e del client. Se il client ha una lenta connessione a Internet da un hotspot Wi-Fi, ciò si rifletterà sul TTFB.

Generalmente, **qualunque cosa sotto i 100 ms è ottimo ed è un buon TTFB**. Google PageSpeed Insights raccomanda di mantenere tempi di risposta del server inferiori a 200 ms. Se vi trovate nell'intervallo di 300-500 ms, siete nella norma. E se avete più di 600 ms, potrebbe esserci qualcosa di mal configurato sul vostro server oppure potrebbe essere il momento di passare a uno stack web superiore. Oppure segui i nostri consigli di seguito per ridurre il vostro TTFB. E ricordate che anche la negoziazione SSL/TLS può essere un fattore importante.

Non vi annoieremo con tutti i dettagli tecnici, tutto ciò che **dovete sapere è che per voi è bene che la latenza di rete e il TTFB siano i più bassi possibile**. Uno dei modi più semplici per farlo è scegliere un server più vicino ai vostri visitatori. È possibile determinare la posizione migliore seguendo i prossimi suggerimenti.

## Suggerimento 1 – Verificate la Geolocalizzazione dei Vostri Visitatori in Google Analytics



The screenshot shows the Google Analytics interface for the 'AUDIENCE' section. The primary dimension is set to 'Country'. The acquisition report shows a total of 32,411 users. The United States is the top source, accounting for 29,470 users (90.49%).

Primary Dimension: Country		Acquisition	
Country		Users	
		32,411	% of Total: 100.00% (32,411)
1.	United States	29,470	(90.49%)
2.	Canada	1,329	(4.08%)
3.	United Kingdom	521	(1.60%)
4.	France	161	(0.49%)
5.	Australia	157	(0.48%)
6.	India	56	(0.17%)

Una delle prime cose da fare è guardare la geolocalizzazione dei vostri visitatori in Google Analytics. Potete trovarla in “Pubblico → Dati geografici → Località”.

Nell'esempio che segue, potete vedere che oltre il 90% del traffico proviene dagli Stati Uniti. Quindi, nella maggior parte dei casi, dovrete posizionare il vostro sito WordPress su un server negli Stati Uniti. Potete anche filtrare i dati fino al livello di città. Questo è particolarmente importante se siete un'azienda locale. Ma normalmente consigliamo una posizione centrale come Iowa, USA.

## Suggerimento 2 – Controllate i Dati Ecommerce

Se gestite un negozio di eCommerce, assicuratevi anche di vedere da dove provengono i vostri clienti. Questo è ovviamente il modo in cui generate ricavi, quindi questi sono i vostri visitatori più importanti. Questo dato dovrebbe coincidere con il traffico visto sopra, ma non è sempre così. Se avete impostato dati di eCommerce o obiettivi in Google Analytics, potete facilmente sovrapporre tali informazioni ai dati di geolocalizzazione per prendere una decisione più consapevole. Oppure verificate le informazioni sulla posizione memorizzate nel database della vostra piattaforma di eCommerce.

## Suggerimento 3: Eseguite un Veloce Test di Latenza

Ci sono un sacco di utili strumenti gratuiti in giro per misurare la latenza dalla posizione corrente per diversi fornitori di cloud. Questi strumenti possono aiutarvi a valutare rapidamente quale regione potrebbe essere la migliore per il vostro sito.

- **gcping.com**: (misura la latenza nelle regioni di Google Cloud Platform, inclusi i server di Kinsta)
- **cloudping.info**: (misura la latenza nelle regioni di Amazon Web Services)
- **azurespeed.com**: (misura la latenza nelle regioni di Azure)

Nell'esempio che segue, possiamo vedere che Los Angeles, USA (us-west2) è il più veloce rispetto alla posizione dove ci troviamo. Tuttavia, se servite i clienti in tutti gli

Stati Uniti, potrebbe essere preferibile scegliere Iowa, USA (us-central1) per assicurarvi una bassa latenza per i visitatori sia della costa occidentale che della costa orientale.

## Misurare la latenza verso le regioni di GCP

REGION	MEDIAN LATENCY
Los Angeles, USA us-west2	28ms
Global HTTP Load Balancer global	31ms
Oregon, USA us-west1	53ms
Northern Virginia, USA us-east4	82ms
Iowa, USA us-central1	83ms
South Carolina, USA us-east1	88ms

Qui da Kinsta, offriamo 20 diversi data center in tutto il mondo. Potete facilmente scegliere un sito che abbia sia bassa latenza sia basso TTFB! Questo vi aiuta anche a ridurre i salti di rete.



## Altri Modi per Ridurre Latenza e TTFB

Oltre a scegliere una posizione vicina al server, ecco alcuni altri modi per ridurre la latenza.

- Implementate la cache sul vostro sito WordPress. Nei nostri test il caching ha ridotto il TTFB di un sorprendente 90%!
- Utilizzate un Content Delivery Network (CDN) per offrire risorse memorizzate nella cache dai POP di tutto il mondo. Ciò contribuisce a ridurre la latenza di rete per i visitatori che potrebbero non essere vicini al server host.
- Sfruttate il protocollo HTTP/2 per ridurre al minimo il numero di round trip, grazie alla parallelizzazione. HTTP/2 è abilitato su tutti i server di Kinsta.
- Riducete il numero di richieste HTTP esterne. Ognuna di queste può aggiungere la propria latenza in base alla posizione del proprio server.
- Il DNS contribuisce al TTFB, quindi dovrete utilizzare un provider DNS premium con tempi di lookup rapidi.
- Utilizzate prefetch e pre-render per eseguire attività dietro le quinte mentre la pagina viene caricata.

Non vi preoccupate; analizzeremo dettagliatamente tutte queste raccomandazioni più avanti.

## Velocità SFTP e Pannello di Amministrazione di WordPress

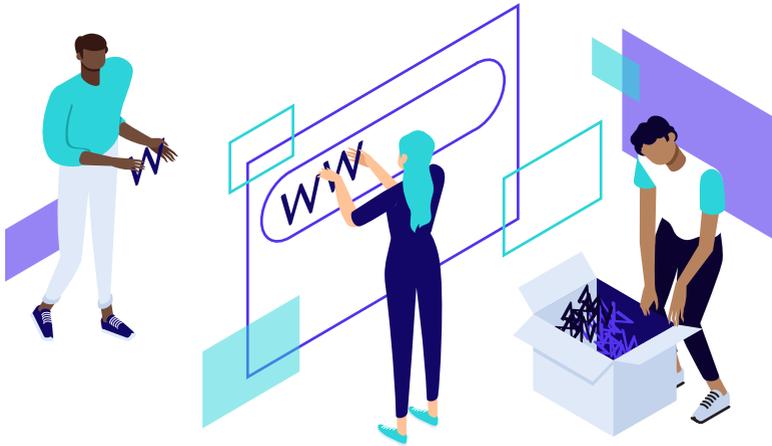
I vostri visitatori e clienti dovrebbero sempre essere la vostra priorità. Ma un altro aspetto della performance di cui molti non parlano riguarda il modo in cui **alcune di queste decisioni influenzano il vostro lavoro quotidiano**. La posizione del data center scelta influisce sulla velocità di download e upload SFTP (trasferimento di file con un client FTP) e sulla reattività del pannello di amministrazione di WordPress.

Quindi, anche se vorrete essere certi di scegliere una posizione migliore per i vostri visitatori, tenete presente che questa può influire anche sulla gestione del sito. Task come il caricamento di file nella libreria multimediale di WordPress saranno più veloci quando il sito è ospitato su un data center più vicino a voi.

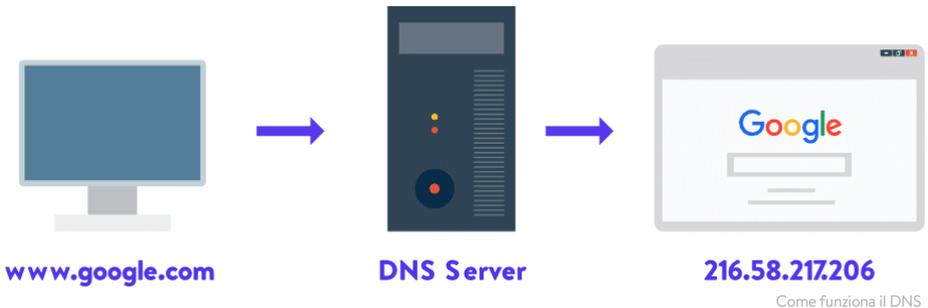
Sentiamo spesso dai parte dei clienti di Kinsta quanto siano sorpresi di quanto sia più veloce il loro pannello di amministrazione con noi. C'è una moltitudine di fattori che incidono, ma avere 20 diversi data center è un ottimo punto di partenza! Scegliete un luogo che funzioni sia per i vostri visitatori che per voi stessi! Dopo tutto, siete voi quelli che probabilmente passeranno migliaia di ore lavorando sul vostro sito web.

CAPITOLO 04:

# Un DNS Premium È Meglio di un DNS Gratuito



**DNS, abbreviazione di Domain Name System, è uno dei componenti più comuni ma meno compresi del panorama web. Per dirla con parole semplici, il DNS aiuta a dirigere il traffico su Internet collegando i nomi di dominio con i server web reali. In sostanza, prende una richiesta human-friendly – un nome di dominio come kinsta.com – e la traduce in un indirizzo IP del server, come 216.58.217.206.**



Potete trovare sia DNS gratuiti che DNS premium. Tutti i clienti di Kinsta hanno accesso al DNS premium tramite Amazon Route 53. E, in generale, crediamo che, oggi come oggi, il DNS premium sia una necessità.

Un ottimo motivo per scegliere un DNS premium è costituito **dalla velocità e dall'affidabilità**. Cercare i record DNS e dirigere il traffico richiede tempo, anche se è solo questione di millisecondi.

In genere, il DNS gratuito messo a disposizione dal registrar del vostro nome di dominio è relativamente lento, mentre i DNS premium offrono spesso prestazioni migliori. Ad esempio, nei nostri test, abbiamo rilevato che **il DNS gratuito di NameCheap era il 33% più lento** del DNS premium di Amazon Route 53. Inoltre, i DNS premium possono offrire maggiore sicurezza e disponibilità, specialmente quando si è sotto attacco DDoS.

Potete utilizzare uno strumento come lo speed test di SolveDNS per misurare i tempi di risoluzione del DNS. Anche DNSPerf fornisce ottimi dati sulle prestazioni su tutti i principali provider DNS.

Per una buona via di mezzo tra il DNS gratuito fornito dal vostro registrar di dominio e il DNS premium, DNSPerf è un servizio gratuito che offre molti dei vantaggi di un DNS premium. Ed è incredibilmente veloce, con tempi di risposta medi inferiori a 20 ms in tutto il mondo (come mostrato qui sotto).

Name Server	Result	Average (milliseconds)	Standard Deviation	95% Confidence Interval
algin.ns.cloudflare.com	✓ Los Angeles: 5.65 ms	5.48 ms	2.69	[3.49, 7.47]
	✓ Dallas: 6.04 ms			
	✓ New York: 3.47 ms			
	✓ Singapore: 3.29 ms			
	✓ London: 3.81 ms			
	✓ Amsterdam: 4.04 ms			
	✓ San Francisco: 12.4 ms			
	✓ Sydney: 6.57 ms			
	✓ Tokyo: 4.02 ms			
heather.ns.cloudflare.com	✓ Los Angeles: 1.55 ms	3.44 ms	3.12	[1.13, 5.75]
	✓ Dallas: 3.98 ms			
	✓ New York: 2.06 ms			
	✓ Singapore: 2.11 ms			
	✓ London: 2.95 ms			
	✓ Amsterdam: 2.17 ms			
	✓ San Francisco: 12.04 ms			
	✓ Sydney: 1.51 ms			
	✓ Tokyo: 2.62 ms			

Tuttavia avvertiamo che Cloudflare registra maggiori tempi di inattività rispetto a molti altri provider. Se servite i visitatori principalmente negli Stati Uniti, DNS Made Easy è un altro ottimo provider DNS premium che potreste tenere in considerazione.

Quest'ultimo ha la fama di fornire alcuni dei migliori tempi di attività DNS visti negli ultimi dieci anni.

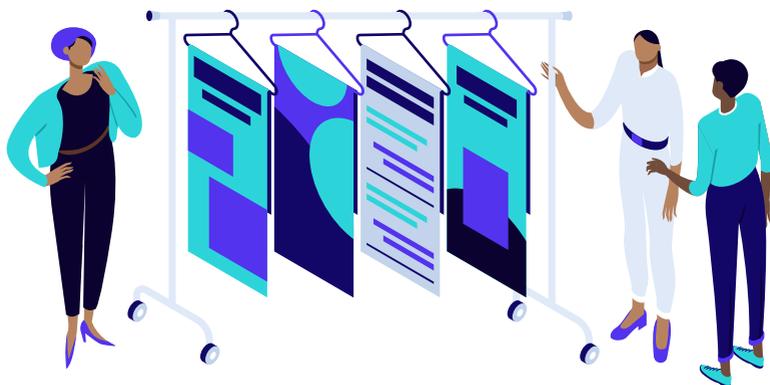
Negli ultimi 30 giorni, DNSPerf mostra questi tempi di uptime riferiti ai seguenti fornitori:

- DNS Made Easy: 99,99% che equivale a 4m 23,0s di inattività mensile.
- Amazon Route 53: 99,88% che equivale a un downtime mensile di 52m 35,7s.
- Cloudflare: 99,85% che equivale a un downtime mensile di 1h 5m 44.6s.

I tempi di inattività dei provider DNS sono così importanti? La risposta, in realtà, è sì e no. Il DNS viene in genere memorizzato nella cache, con gli ISP che utilizzano il valore time to live (TTL) nel record DNS. Pertanto, se un provider DNS si interrompe per 10 minuti, molto probabilmente non noterete nulla. I tempi di inattività sono importanti se il provider incorre costantemente in interruzioni più lunghe e frequenti, o se i record ISP e DNS utilizzano entrambi valori TTL molto bassi.

CAPITOLO 05:

# Il Vostro Tema WordPress È Importante



**A tutti piace un tema WordPress nuovo di zecca, ma fate attenzione prima di installarlo e scegliete quello con tutte le migliori caratteristiche. Innanzitutto, dovrete dare una lettura al nostro articolo sulle differenze tra temi gratuiti e temi a pagamento. Per quanto riguarda le prestazioni, ogni elemento che vedete in un tema ha delle conseguenze sulla velocità complessiva del vostro sito web. E, sfortunatamente, tra le migliaia di temi in circolazione, ce ne sono sia di buoni che di cattivi.**

***Il tema WordPress è importante per le prestazioni.  
Scegliete quello giusto fin dall'inizio.***

Quindi, come fate a sapere quale scegliere? Vi consigliamo di optare per una delle seguenti soluzioni:

- Un tema WordPress veloce e leggero che è sviluppato solo con le funzionalità di cui avete realmente bisogno, e nient'altro.
- Un tema WordPress più ricco di funzionalità, ma in cui sia possibile disabilitare le funzionalità che non vengono utilizzate.

Cose come i Google Fonts, le icone di Font Awesome, slider, gallerie, video e script parallax, ecc. sono solo alcune delle molte cose che dovrete essere in grado di disattivare se non le utilizzate. Non dovrete provare a modificarle manualmente dopo l'installazione, e non vi mostreremo 50 modi diversi per eliminare queste cose. Invece, dovrete partire con, o passare a, un tema WordPress che sia leggero fin dall'inizio o che vi offra questa possibilità.

Di seguito riportiamo alcuni temi WordPress che consigliamo, su cui non potete sbagliare! Fidatevi di noi, ci ringrazierete più tardi.

Ogni tema menzionato di seguito è completamente compatibile con WooCommerce e Easy Digital Downloads, WPML, BuddyPress e bbPress. Abbiamo eseguito alcuni test di velocità con ciascun tema utilizzando la seguente configurazione:

- Ospitato su Kinsta, con WordPress 5.0 in esecuzione
- PHP 7.3 e SSL (HTTPS)
- CDN di Kinsta
- Per comprimere automaticamente le immagini, è stato utilizzato Imagify.

## GeneratePress

GeneratePress è un tema WordPress veloce, leggero (meno di 1 MB zippato), reattivo su mobile, sviluppato con attenzione a velocità, SEO e usabilità. È stato sviluppato da Tom Osborne, uno sviluppatore canadese. È attivamente aggiornato e ben supportato. Anche alcuni membri del team di Kinsta usano GeneratePress per i loro progetti.

Sono disponibili sia una versione gratuita che una versione premium. Se date un'occhiata alla repository di WordPress, vedrete che la versione gratuita vanta al momento più di 200.000 installazioni attive, oltre 2 milioni di download e un'impressionante 5 stelle su 5 (oltre 850 persone hanno assegnato 5 stelle).

Una delle grandi caratteristiche di GeneratePress è che questo tema utilizza per tutte le opzioni il Customizer nativo di WordPress, il che significa che potete visualizzare istantaneamente ogni modifica che apportate prima di premere il pulsante pubblica. Questo significa anche che non dovete imparare ad usare un nuovo pannello di controllo del tema.

The perfect **lightweight**  
theme for your next project.

Install

Premium

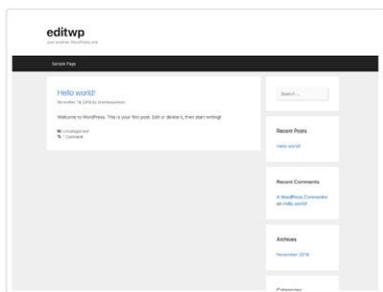
**1,643,956+**  
Downloads

**700+**  
★★★★★

**100,000+**  
Active websites

**40,000+**  
Happy customers

Ma quanto è veloce? Abbiamo effettuato una nuova installazione di GeneratePress, e abbiamo eseguito cinque test di velocità su Pingdom facendone la media. Il tempo di caricamento totale è stato di 305 ms, con una **dimensione totale della pagina di soli 16,8 KB**. È sempre utile avere un test di base per vedere di cosa è capace il tema in termini di prestazioni grezze.



Performance grade

**A 97**

Page size

**16.8 KB**

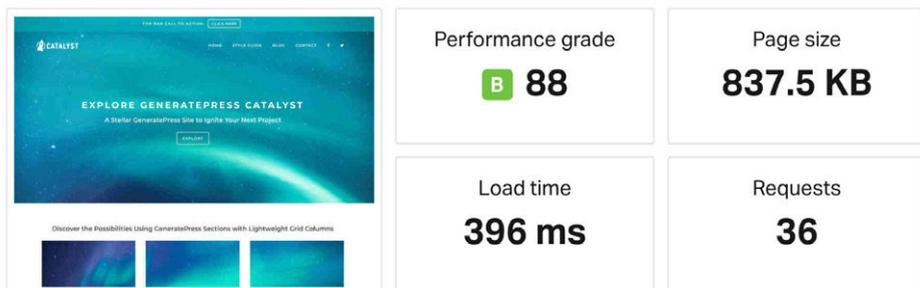
Load time

**305 ms**

Requests

**7**

Abbiamo quindi eseguito un'altra serie di test con uno dei temi precostruiti preso dalla libreria del sito GeneratePress. Questo contiene immagini, sfondi, nuove sezioni, ecc. Un vantaggio di GeneratePress è che offre un sacco di temi precostruiti, i quali non richiedono plugin page builder. Come potete vedere, registra ancora un tempo inferiore ai 400 ms.



Ora, naturalmente, in un ambiente reale potreste avere altre cose, come Google Analytics, pixel di remarketing di Facebook, Hotjar, ecc. Ma dovrete essere in grado di centrare facilmente un obiettivo inferiore a 1 secondo. Per approfondire, leggete la recensione di GeneratePress su woorkup.

Vi mostreremo altri modi per ottimizzare e accelerare WordPress di seguito.

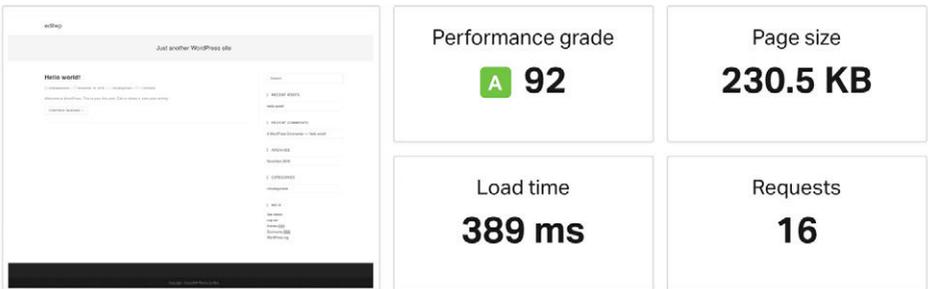
## OceanWP

The screenshot shows the OceanWP website homepage. At the top left is the 'OceanWP.' logo. To its right are navigation links: 'Demos', 'Extensions', 'Blog', 'Support', and 'Download'. Further right are 'Sign In / Sign Up' links. The main heading is 'Create Beautiful Websites', with 'Beautiful' underlined in blue. Below the heading is a paragraph: 'With 899,712 downloads, OceanWP is the fastest growing theme for WordPress. It's the favorite choice of thousands of developers and hobby-users.' A prominent blue button with a white WordPress logo and the text 'FREE DOWNLOAD' is centered below the paragraph. At the bottom, there are two large, stylized buttons: 'DEVELOPER' on the left and 'HOBBYIST' on the right, set against a background of a computer monitor and keyboard.

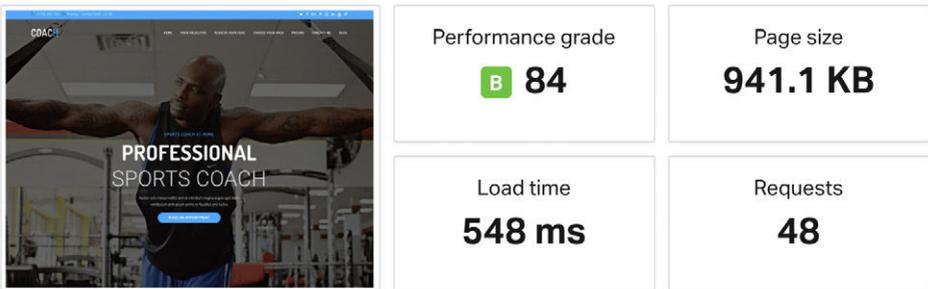
Il tema OceanWP è leggero ed estremamente estensibile. Vi permette di creare praticamente qualsiasi tipo di sito web come blog, portfolio, siti aziendali e negozi online WooCommerce con un design bello e professionale. Sviluppato da Nicolas Lecocq, è aggiornato regolarmente ed è ben supportato.

Proprio come GeneratePress, è disponibile una versione gratuita e una versione premium. Se date un'occhiata alla repository di WordPress, la versione gratuita vanta attualmente oltre 400.000 installazioni attive, e un'altra impressionante valutazione di 5 stelle su 5 (oltre 2.600 persone hanno dato 5 stelle).

E quanto è veloce? Abbiamo eseguito una nuova installazione di OceanWP, abbiamo effettuato cinque test di velocità su Pingdom e ne abbiamo fatto la media. Il tempo di caricamento totale è stato di **389 ms** con una **dimensione totale della pagina di soli 230,8 KB**. Gli script in OceanWP sono leggermente più grandi, ma nulla in particolare da riportare.



Abbiamo quindi eseguito un'altra serie di test con uno dei temi demo preso della libreria del sito OceanWP. Questo contiene immagini, sfondi, nuove sezioni e richiede il plugin page builder Elementor. Con questa configurazione, il tema registra ancora un tempo inferiore a **600 ms**.

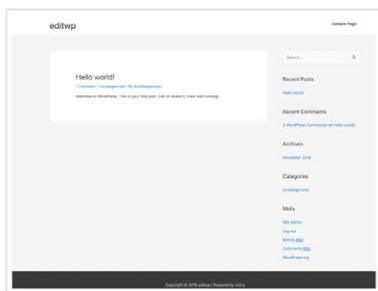


## Astra

Astra è un tema veloce, completamente personalizzabile e graficamente gradevole, adatto per blog, portfolio personali, siti web aziendali e vetrine di negozi WooCommerce. È molto leggero (meno di 50 KB sul front-end) e offre una velocità senza rivali. Sviluppato dal team di Brainstorm Force, è aggiornato attivamente e ben supportato. Potreste ricordarvi di loro come gli sviluppatori del famoso plugin All In One Schema Rich Snippets, che è stato utilizzato per molti anni.

Proprio come GeneratePress e OceanWP, il tema è disponibile in versione gratuita e premium. Se date un'occhiata alla repository di WordPress, la versione gratuita conta attualmente oltre 400.000 installazioni attive, oltre 1,6 milioni di download e un altro impressionante 5 stelle su 5 (oltre 2.500 persone hanno assegnato 5 stelle).

Quanto è veloce? Abbiamo predisposto una nuova installazione di Astra, abbiamo eseguito cinque test di velocità su Pingdom e abbiamo fatto la media. Il tempo di caricamento totale è stato di **243 ms** con una **dimensione totale della pagina di soli 26,6 KB**.



Performance grade

**A 98**

Page size

**26.6 KB**

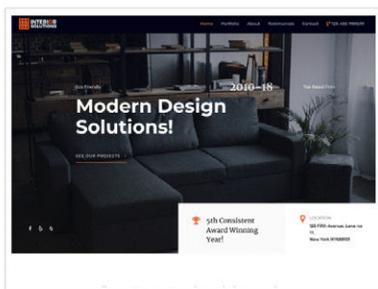
Load time

**243 ms**

Requests

**5**

Abbiamo quindi eseguito un'altra serie di test con uno dei temi demo della libreria del sito Starter kit di Astra. Il tema contiene immagini, sfondi, nuove sezioni e richiede il plugin page builder Elementor. Come potete vedere, siamo ancora sotto i 700 ms. Nota: le immagini di questa demo sono state completamente compresse, ma gli sviluppatori hanno scelto inizialmente delle immagini ad altissima risoluzione.



Performance grade

**B 86**

Page size

**1.6 MB**

Load time

**652 ms**

Requests

**54**

È importante prendere con un pizzico di sale le differenze tra i test di velocità di questi tre temi. Il problema è che è quasi impossibile eseguire un confronto diretto assolutamente preciso. La cosa importante che volevamo dimostrare è che tutti questi temi per WordPress sono incredibilmente veloci, sia se pronti per l'uso che in demo!

## Avvertenza sui Page Builder

Come avrete probabilmente notato, OceanWP e Astra richiedono entrambi un page builder per utilizzare i temi della libreria del sito. Ci sono alcune cose da tenere a mente quando si utilizza un page builder:

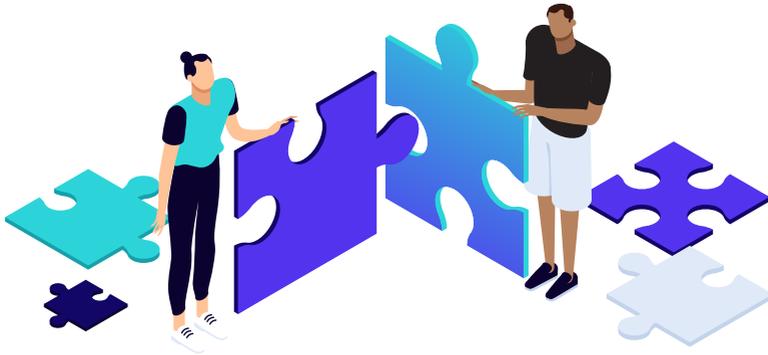
- Alcuni page builder potrebbero aumentare il tempo di caricamento del vostro sito. Questo perché devono caricare CSS e JS aggiuntivi per farvi gestire le cose senza codice. Ecco come avviene la magia! Consigliamo sempre di testare la velocità dei siti WordPress prima e dopo aver installato un page builder.
- Vi troverete a dipendere da quel page builder per il vostro design. Assicuratevi di sceglierne uno che venga aggiornato regolarmente e abbia tutto ciò di cui potreste aver bisogno nel lungo periodo.

Detto questo, siamo sempre stati grandi estimatori di page builder come Elementor and Beaver Builder. Per gran parte, sono sviluppati tenendo in debito conto le prestazioni e aumentano solo un po' i tempi complessivi. Per la maggior parte, la funzionalità e l'usabilità valgono il gioco, in quanto questi plugin vi consentono di creare qualsiasi cosa possiate immaginare! In alcuni casi potrebbero anche farvi migliorare le prestazioni, in quanto potrebbero sostituire altri 5 o più plugin che avreste dovuto utilizzare al loro posto.

Tuttavia, se non avete bisogno di un plugin per la creazione di pagine, non installatene uno solo perché è figo. Sarà anche interessante vedere come il nuovo editor Gutenberg influirà sul design dei siti nei prossimi due anni.

CAPITOLO 06:

# La Verità sui Plugin di WordPress



**Ora lo scoop sui plugin WordPress. Potrebbero avervi detto che non dovrete installare troppi plugin per non rallentare il vostro sito WordPress. Anche se a volte questo è vero, non è il fattore più critico. Il numero di plugin non è importante quanto la qualità dei plugin. Ecco, l'abbiamo detto.**

Proprio come i temi, è importante sapere come viene sviluppato il plugin e se è stato prodotto tenendo conto delle prestazioni. Abbiamo molti clienti da Kinsta che eseguono 30-40 plugin e i loro siti continuano a caricarsi in meno di un secondo.

Sebbene sia divertente aggiungere codice al vostro sito, questo non è sempre pratico per i seguenti motivi:

- 1.** Dovete mantenere il codice da soli e tenerlo aggiornato man mano che cambiano gli standard. Le persone sono impegnate, perché non fare affidamento su ottimi sviluppatori che conoscono gli standard meglio di molti altri?
- 2.** Il più delle volte, un plugin ben codificato non causerà rallentamenti più dello stesso core.
- 3.** Dovete ricordare che la maggior parte della community di WordPress non è così esperta di tecnologia come il gruppo degli sviluppatori. I plugin sono soluzioni che aiutano a risolvere i problemi.

Ovviamente non ci sono plugin ben progettati da cui vorreste stare alla larga. Ma fidatevi di noi: abbiamo visto il peggio del peggio qui da Kinsta. Abbiamo visto di persona che molti, non tutti, i plugin che abbiamo vietato su Kinsta causavano problemi di prestazioni.

Tuttavia, non si può ignorare che una delle cose che la gente ama di WordPress è la sua enorme libreria di plugin di terze parti. Ma con più di 56.000 plugin gratuiti presenti solo su WordPress.org e altri ancora ospitati altrove, può essere difficile trovare quell'unico plugin che vi è necessario. È come cercare un ago in un pagliaio!

Cerchiamo solo di condividere cose che utilizziamo quotidianamente. E, sì, utilizziamo plugin WordPress nel nostro sito come tutti voi. Tra l'altro, molti membri del team di Kinsta sviluppano e vendono plugin.

## Un Grosso Problema dei Plugin di WordPress

Un grosso problema dei plugin di WordPress è la procedura di disinstallazione. Ogni volta che installate un plugin o un tema WordPress, questo memorizza i dati nel database. Il problema è che quando si elimina un plugin utilizzando uno dei metodi standard, in genere questo si lascia dietro le tabelle e le righe del database. Con il passare del tempo questo può far sì che si accumulino molti dati e può anche rallentare il vostro sito. Nel nostro esempio, abbiamo disinstallato il plugin di sicurezza di Wordfence e questo ha lasciato nel nostro database 24 tabelle (come si vede qui sotto). Ed è ancora peggio se rimangono i dati nella tabella `wp_options`.

 <b>wp_wfBadLeechers</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	0	InnoDB	latin1_swedish_ci	16	KiB
 <b>wp_wfBlockedIPLog</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	35	InnoDB	utf8_general_ci	16	KiB
 <b>wp_wfBlocks</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	32	InnoDB	utf8_general_ci	32	KiB
 <b>wp_wfBlocksAdv</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	0	InnoDB	utf8_general_ci	16	KiB
 <b>wp_wfConfig</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	156	InnoDB	utf8_general_ci	16	KiB
 <b>wp_wfCrawlers</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	10	InnoDB	latin1_swedish_ci	16	KiB
 <b>wp_wfFileMods</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	0	InnoDB	utf8_general_ci	16	KiB
 <b>wp_wfHits</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	1,597	InnoDB	latin1_swedish_ci	1.3	MiB
 <b>wp_wfHoover</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	0	InnoDB	utf8_general_ci	32	KiB
 <b>wp_wfIssues</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	0	InnoDB	utf8_general_ci	16	KiB
 <b>wp_wfKnownFileList</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	0	InnoDB	utf8_general_ci	16	KiB
 <b>wp_wfLeechers</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	156,922	InnoDB	latin1_swedish_ci	13.5	MiB
 <b>wp_wfLockedOut</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	0	InnoDB	utf8_general_ci	16	KiB
 <b>wp_wfLocs</b>		 Browse	 Structure	 Search	 Insert	 Empty	 Drop	4	InnoDB	utf8_general_ci	16	KiB

<input type="checkbox"/>	wp_wfLogins	★	Browse	Structure	Search	Insert	Empty	Drop	5	InnoDB	utf8_general_ci	48	KiB
<input type="checkbox"/>	wp_wfNet404s	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8_general_ci	32	KiB
<input type="checkbox"/>	wp_wfNotifications	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	wp_wfPendingIssues	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	wp_wfReverseCache	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	latin1_swedish_ci	16	KiB
<input type="checkbox"/>	wp_wfScanners	★	Browse	Structure	Search	Insert	Empty	Drop	618	InnoDB	latin1_swedish_ci	64	KiB
<input type="checkbox"/>	wp_wfSNIPCache	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	64	KiB
<input type="checkbox"/>	wp_wfStatus	★	Browse	Structure	Search	Insert	Empty	Drop	40	InnoDB	utf8_general_ci	48	KiB
<input type="checkbox"/>	wp_wfThrottleLog	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	32	KiB
<input type="checkbox"/>	wp_wfVulnScanners	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	latin1_swedish_ci	16	KiB

Tabelle di WordFence

E, a parte il database, molti plugin lasciano anche cartelle e file. Nella nostra esperienza, questo succede normalmente con i plugin di sicurezza e di caching, che creano directory aggiuntive per la memorizzazione. Ad esempio, dopo che il plugin Wordfence è stato eliminato, ci è rimasta una cartella “wflog” nella nostra directory wp-content. Ma non ce l’abbiamo con Wordfence, la maggior parte dei plugin e dei temi sul mercato funzionano in questo modo.

Remote site: /www/.../public/wp-content/wflogs

- upgrade
- uploads
- wflogs

Filename	File...	Filetype	Last modified
..			
.htaccess	133	HTACCESS File	7/24/2017 10:54:49 AM
attack-data.php	40,0...	PHP File	7/24/2017 10:54:49 AM
config.php	594,...	PHP File	7/24/2017 10:57:30 AM
ips.php	51	PHP File	7/24/2017 10:54:50 AM
rules.php	101,...	PHP File	7/24/2017 10:54:59 AM
wafRules.rules	44,2...	RULES File	7/24/2017 10:54:59 AM

Log di WordFence

## Perché Gli Sviluppatori Lo Fanno?

Ora probabilmente vi starete chiedendo perché gli sviluppatori non aggiungo opzioni che permettano di fare automaticamente pulizia quando si disinstalla e si elimina un plugin? Bene, lo fanno. Ma ci sono alcuni di motivi per cui probabilmente non è così ovvio sin da subito.

- 1. Vogliono mantenere le impostazioni per l'utente.** Se eliminate un plugin di WordPress e decidete di riprovarlo più tardi, tutte le impostazioni e i dati saranno ancora lì. Sebbene questo sia super conveniente, non è il metodo più efficiente.
- 2. Non si preoccupano delle prestazioni.** Alcuni sviluppatori potrebbero obiettare che lasciare le tabelle non influisce sulle prestazioni. Ma immaginate un sito nel corso di dieci anni, dopo aver usato centinaia di plugin che hanno generato probabilmente migliaia di righe o tabelle. Le query sul database hanno un impatto significativo sulle prestazioni del vostro sito WordPress e i plugin possono fare molte di queste richieste se lo sviluppatore non è attento. Generalmente, un plugin ben scritto dovrebbe solo interrogare le tabelle o le righe a cui è legato, tuttavia non è sempre così. Lo abbiamo visto in prima persona da Kinsta, lunghe interrogazioni al database che hanno portato alla scansione di un sito a causa di dati auto-caricati non necessari lasciati nella tabella `wp_options`.
- 3. Hanno fatto un errore.** Il manuale dei plugin di WordPress dice anche che “gli sviluppatori meno esperti a volte commettono l'errore di utilizzare l'hook di disattivazione a questo scopo”.

La buona notizia? Ci sono diversi metodi per far pulizia e rimuovere correttamente un plugin. I link ai seguenti tutorial approfonditi sono disponibili nel nostro sito web all'indirizzo <https://kinsta.com/speed-up-wordpress>.

- Come Disinstallare un Plugin WordPress (nel Modo Giusto)
- Come Pulire Manualmente le Tabelle Rimaste Indietro

CAPITOLO 07:

# Impostazioni Ottimali di WordPress



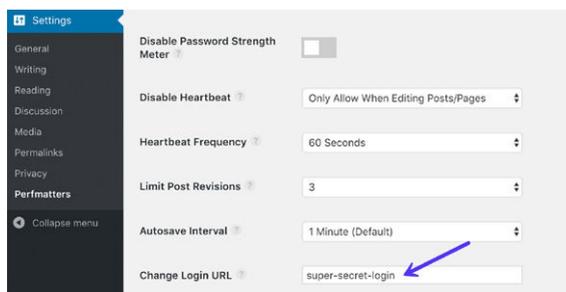
**Ora passiamo alle impostazioni ottimali di WordPress. Si tratta di alcune modifiche che potete apportare per velocizzare il vostro sito WordPress. In molti casi sono modifiche molto piccole, ma tutto ci può essere di aiuto!**

## **Cambiare l'URL di Accesso a WordPress**

Di default, l'URL di accesso al vostro sito WordPress è `domain.com/wp-admin/`. Uno dei problemi che ne conseguono è che tutti i bot, gli hacker e gli script che sono in circolazione lo sanno. Modificando l'URL, potete rendervi meno raggiungibili, proteggervi meglio dagli attacchi brute force e diminuire la larghezza di banda utilizzata dai robot che raggiungono ripetutamente questo URL.

Cambiare l'URL di accesso a WordPress può anche aiutare a prevenire errori comuni come il "429 Too Many Requests". Questa non è una soluzione per tutto, è solo un piccolo trucco che può contribuire a proteggervi e a ridurre il carico su quella pagina.

Per cambiare l'URL di accesso a WordPress, vi consigliamo di utilizzare uno dei seguenti plugin:



Cambiare l'URL di accesso a WordPress in Perfmatters

- **WPS Hide Login** (gratuito, disponibile su [WordPress.org](https://WordPress.org))
- **Perfmatters** (premium, ma include altre impostazioni per l'ottimizzazione delle prestazioni. Sviluppato da un membro del team di Kinsta e disponibile all'indirizzo [perfmatters.io](https://perfmatters.io))

## Disabilitare o Specificare gli Aggiornamenti di Temi e Plugin

La lentezza della dashboard di WordPress può essere causata dal network, dall'ubicazione del data center e persino dalle versioni di PHP. Ma un altro fattore di cui molte persone non parlano è il controllo degli aggiornamenti di WordPress eseguito in background. Questo è un caso in cui avere un sacco di plugin e temi WordPress potrebbe avere conseguenze negative. WeFoster ha pubblicato un ottimo post in cui hanno coniato la frase "Third Party Plugin Update Check Syndrome" o TPPUCS.

Essenzialmente il problema è che il checker degli aggiornamenti di WordPress invia in background una richiesta GET esterna (<https://third-party-plugin/update-check.php>). A volte questa operazione può essere regolare o molto frequente. Se avviene di continuo, potrebbe portare a un intasamento del pannello di amministrazione.

Questo è un problema che riguarda più che altro il modo in cui è stato sviluppato il checker degli aggiornamenti di WordPress. Se registrate lunghi tempi di caricamento della dashboard di WordPress, potreste provare a disabilitare gli aggiornamenti automatici. Avviso: eseguite questa operazione solo se intendete verificare manualmente gli aggiornamenti. Molti aggiornamenti riguardano sicurezza e correzioni di bug.

Per disabilitare gli aggiornamenti, vi consigliamo di utilizzare uno dei seguenti plugin (entrambi disponibili su [WordPress.org](https://WordPress.org)):

- **Disable All WordPress Updates:** completamente gratuito e senza impostazioni. Fa bene quello che promette.
- **Easy Updates Manager:** fornisce un controllo selettivo sugli aggiornamenti. La versione core è gratuita.

Potreste facilmente impostare un promemoria nel calendario, disattivare il plugin una volta alla settimana, controllare gli aggiornamenti e quindi riattivarlo.

## Disattivare i Pingback

Un pingback è un commento automatico che viene creato quando un altro blog si collega al vostro. Ci possono anche essere auto-pingback che vengono generati quando ci si collega ad un articolo all'interno del proprio blog.

Consigliamo semplicemente di disabilitarli perché generano query inutili e spam aggiuntivo sul vostro sito. Ricordate, meno chiamate il vostro sito WordPress deve fare, meglio è, specialmente in siti ad alto traffico. Per non parlare del fatto che un pingback sul proprio sito web è semplicemente fastidioso. Seguite questi passaggi per disabilitare i pingback.

## Passo 1 – Disattivare i Pingback Provenienti da Altri Blog

Nella dashboard di WordPress, fate clic su “Impostazioni → Discussione”. Nella sezione Impostazioni Discussione deselezionate l’opzione “Permetti le notifiche dei link da altri blog (pingback e trackback) sui nuovi articoli”.

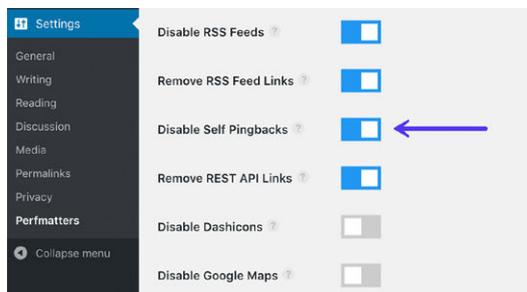
### Discussion Settings

#### Default article settings

- Attempt to notify any blogs linked to from the article
- Allow link notifications from other blogs (pingbacks and trackbacks) on new articles
- Allow people to post comments on new articles

*(These settings may be overridden for individual articles.)*

## Passo 2: Disabilitare gli Auto-Pingback



Disabilitare gli auto-pingback con Perfmatters

Per disabilitare gli auto-pingback avete un paio di opzioni. Potete utilizzare il plugin gratuito No Self Pings (disponibile su WordPress.org), oppure potete utilizzare un plugin premium come Perfmatters (disponibile su perfmatters.io).

In alternativa, potete anche disabilitare gli auto-pingback aggiungendo il seguente codice al file `functions.php` del tema WordPress. Attenzione, la modifica del codice di un tema WordPress potrebbe interrompere il vostro sito se non eseguite l'operazione correttamente. Suggerimento: potete facilmente aggiungere snippet PHP come questo con il plugin gratuito Code Snippets (disponibile su WordPress.org). Ciò significa che non dovrete mai toccare il vostro tema.

```
function wpsites_disable_self_pingbacks( &$links ) {
    foreach ( $links as $l => $link )
        if ( 0 === strpos( $link, get_option( 'home' ) ) )
            unset($links[$l]);
}

add_action( 'pre_ping', 'wpsites_disable_self_pingbacks' );
```

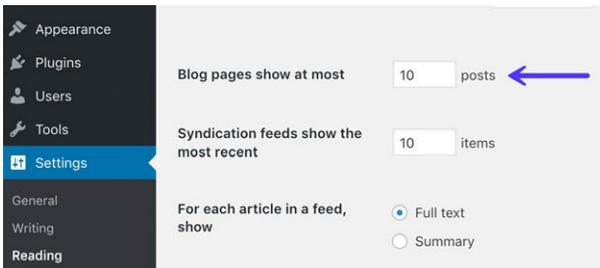
## Limitate i Post sul Feed del Vostro Blog

Se il feed del vostro blog è impostato come pagina iniziale o se è un'altra pagina del vostro sito, non è necessario che vengano caricate 50 miniature contemporaneamente. Per chi ha un blog ad alto traffico, la home page è la pagina più importante del sito e si vorrà certamente caricarla rapidamente. Meno richieste e meno media ci sono, meglio è in termini di prestazioni.

Inoltre, questo è esattamente il motivo per cui è stata inventata la paginazione (si veda sotto). La paginazione è in fondo ai feed del blog e vi permette di navigare alla pagina successiva. In genere sono numeri, ma potrebbero anche essere le etichette “precedente/successivo”. Molto probabilmente il vostro tema WordPress avrà già integrato l’impaginazione personalizzata.



WordPress imposta di default il limite di post per pagina sulle nuove installazioni a 10, ma abbiamo visto che questo valore è cambiato tante volte che abbiamo perso il conto. Assicuratevi quindi di ricontrollare quale valore è stato impostato. Consigliamo di fissare un valore compreso tra 8 e 12. Se siete curiosi, abbiamo impostato 12 sulla homepage del blog di Kinsta.



Potete trovare questa opzione nel pannello di amministrazione di WordPress, alla voce “Impostazioni → Lettura”. Potete quindi modificare il valore del campo “Le pagine del blog visualizzano al massimo”.

CAPITOLO 08:

# Perché la Cache È Così Importante



**Il caching è di gran lunga uno degli strumenti più importanti e più semplici per velocizzare WordPress! Ma prima di mostrarvi come utilizzare la cache, è essenziale innanzitutto capire come funziona e quali sono i diversi tipi di cache disponibili.**

## **Che Cosa è il Caching?**

In breve, ogni pagina web visitata del vostro sito WordPress implica una richiesta al server, l'elaborazione da parte di quel server (comprese le query del database), e quindi un risultato finale inviato dal server al browser dell'utente. Il risultato è il vostro sito web, completo di tutti i file e di tutti gli elementi che lo fanno apparire così com'è.

Ad esempio, potreste avere un'intestazione, delle immagini, un menu e un blog. Dal momento che il server deve elaborare tutte queste richieste, ci vuole del tempo prima che la pagina web completa venga consegnata all'utente, specialmente con siti web ingombranti o più grandi.

È qui che entra in gioco un plugin per la gestione della cache in WordPress! Il caching indica al server di memorizzare alcuni file su disco o RAM, a seconda della

configurazione. Pertanto, il server può ricordare e duplicare lo stesso contenuto che è stato utilizzato in precedenza. Fondamentalmente, riduce la quantità di lavoro necessaria per generare una visualizzazione di pagina. Di conseguenza, le vostre pagine web vengono caricate molto più velocemente, direttamente dalla cache.

Alcuni altri vantaggi del caching:

- **Il vostro server utilizza meno risorse** – Questo si lega alla velocità, dal momento che meno risorse rendono un sito più veloce. Inoltre, il caching permette anche meno lavoro a carico del vostro server. Questo è molto importante quando si tratta di siti altamente dinamici, come i siti ad iscrizione, ed è determinante ciò che potete o non potete servire dalla cache.
- **Vedrete un TTFB più basso** – Il caching è uno degli strumenti più semplici che avete a disposizione per ridurre il vostro TTFB. In effetti, nei nostri test il caching riduce in genere il TTFB fino al 90%!

## Tipi di Caching

Per quanto riguarda i tipi di caching, esistono due diverse tipologie generali utilizzate comunemente:

1. Caching a livello di Server
2. Caching con un Plugin

### 1. Caching a Livello di Server

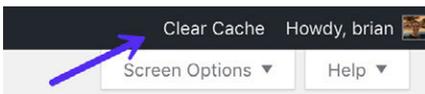
Il caching a livello di server è di gran lunga uno degli approcci più semplici per l'utente finale. Il provider di hosting WordPress lo gestisce al posto vostro. Da Kinsta utilizziamo i seguenti **quattro tipi di cache**, che vengono tutti automaticamente eseguiti a livello di software o di server:

- Bytecode cache
- Object cache
- Page cache
- CDN cache

Questo significa che non dovete preoccuparvi di fare confusione con complicati plugin di caching. Potete smettere di cercare su Google i “migliori plugin di caching” e concentrarvi su attività più produttive.

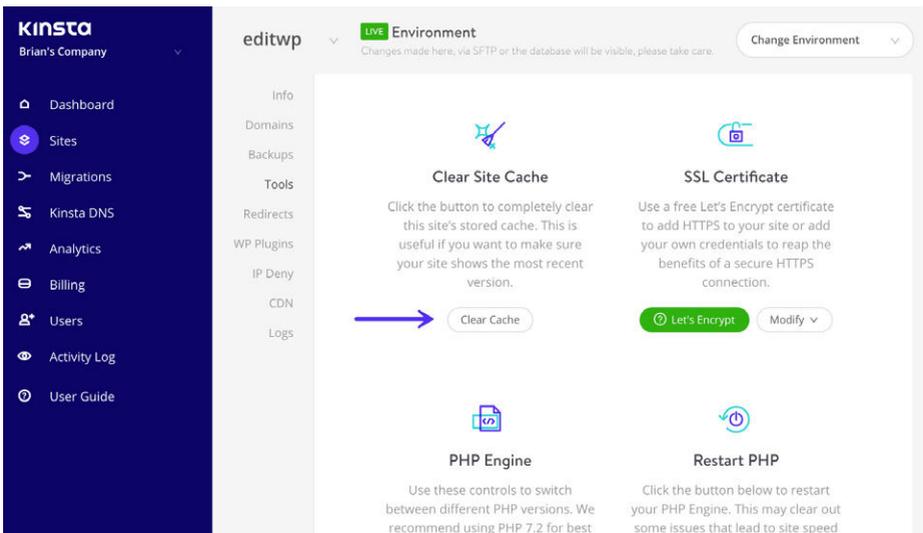
La cache di pagina è configurata per funzionare subito con un’installazione standard di WordPress. Non dovete fare niente! Avviate semplicemente il vostro sito WordPress e il caching delle pagine inizierà subito.

Abbiamo inoltre regole di caching specifiche per i siti di ecommerce come WooCommerce e Easy Digital Downloads. Di default, alcune pagine che non dovrebbero mai essere memorizzate nella cache, come cart, my-account e checkout, sono escluse dalla cache. Gli utenti bypassano automaticamente la cache quando vengono rilevati il cookie `woocommerce_items_in_cart` o il cookie `edd_items_in_cart`, per garantire una procedura di checkout regolare e sincronizzata.



Potete facilmente cancellare la cache del vostro sito WordPress in qualsiasi momento dalla toolbar di amministrazione.

È, inoltre, una funzionalità integrata nel nostro cruscotto MyKinsta. Basta fare clic su Strumenti e poi su “Svuota Cache”.



## 2. Caching con un Plugin

Se il vostro provider di hosting non fornisce supporto per la cache, è possibile utilizzare un plugin di caching di WordPress di terze parti. La nostra esperienza ci fa consigliare uno dei seguenti plugin:

- **WP Rocket** (premium, disponibile su [wp-rocket.me](http://wp-rocket.me))
- **Cache Enabler** (gratuito, disponibile su [WordPress.org](http://WordPress.org))
- **W3 Total Cache** (gratuito, disponibile su [WordPress.org](http://WordPress.org))

Da Kinsta supportiamo anche WP Rocket! Di solito non consentiamo di installare plugin di caching nel nostro ambiente perché questi vanno in conflitto con la nostra soluzione integrata di caching. Tuttavia, a partire da WP Rocket 3.0, la funzionalità di caching di pagina sarà automaticamente disabilitata durante l'esecuzione sui server di Kinsta.

Ciò consente ai clienti di Kinsta di utilizzare la nostra veloce cache a livello di server, ma di sfruttare anche le fantastiche funzioni di ottimizzazione offerte da WP Rocket.

## Caching o Non Caching

Quanto aiuta il caching? La prova è nei fatti.

Abbiamo eseguito alcuni speed test con il caching a livello di server di Kinsta, per farvi vedere la differenza che genera, sia in termini di velocità generale, che di TTFB.

## Nessun Caching

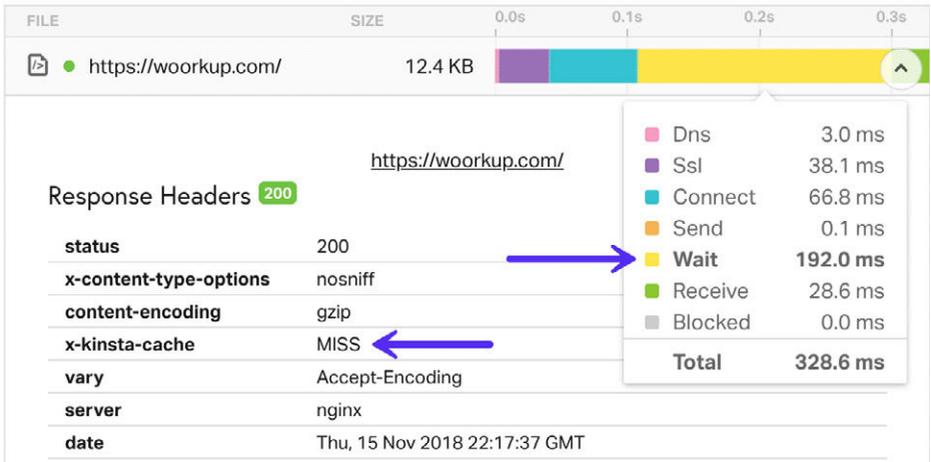


Performance grade <b>B 89</b>	Page size <b>230.6 KB</b>
Load time <b>507 ms</b>	Requests <b>40</b>

Abbiamo eseguito per la prima volta cinque test su Pingdom senza abilitare il caching e, quindi, abbiamo fatto la media.

## TTFB Senza Caching

È importante notare la differenza nel TTFB con e senza cache. Il TTFB in Pingdom è rappresentato dalla barra gialla “wait”. Come potete vedere, il TTFB senza cache è di 192 ms. Si vede subito che non viene servito dalla cache perché l’intestazione x-kinsta-cache mostra un MISS.



## Con Caching Attivo



Performance grade  
**B 89**

Load time  
**351 ms**

Page size  
**230.6 KB**

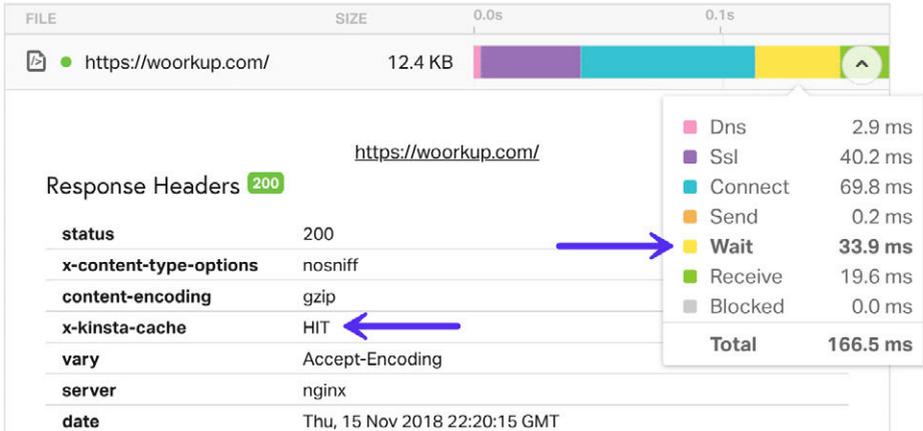
Requests  
**40**

Abbiamo quindi abilitato il caching a livello di server e abbiamo eseguito cinque test su Pingdom, facendone la media.

Come potete vedere, **il caching a livello di server riduce il tempo di caricamento delle pagine del 33,77%**! E questo senza alcun lavoro extra. Inoltre, il sito che abbiamo testato è abbastanza ottimizzato, quindi i siti più grandi e non ottimizzati sono destinati a registrare differenze ancora maggiori.

# TTFB con Caching Attivo

Ora, se diamo un'occhiata al TTFB con la cache attiva, notiamo subito che questo è inferiore a 35 ms. Si può capire che viene servito dalla cache in quanto l'intestazione `x-kinsta-cache` mostra un `HIT`.



La cache del CDN è altrettanto importante della cache del vostro host WordPress. Approfondiremo i CDN più avanti.

*Il caching di WordPress può facilmente ridurre i tempi di caricamento della pagina di oltre il 33%!*

## Anomalie con Caching e Siti ad Iscrizione

I siti ad iscrizione contengono molti **contenuti non memorizzabili nella cache** e pagine che cambiano continuamente. Cose come la pagina di accesso per i membri della comunità (che potrebbero essere raggiunte regolarmente a seconda delle dimensioni del sito), le pagine di checkout per prodotti o corsi digitali e i forum di discussione sono comuni colpevoli e punti dolenti, in quanto queste pagine non possono in genere essere memorizzate nella cache.

Ma non finisce qui. Su siti WordPress standard, anche la dashboard di WordPress non viene memorizzata nella cache per gli **utenti “connessi”**. Questo va bene quando avete solo pochi autori e amministratori, ma quando i membri che usano la dashboard sono migliaia, nascono immediatamente problemi di prestazioni poiché nessuna di queste pagine può essere servita dalla cache sul server. Ciò significa che avete bisogno di potenza e di essere sostenuti in background da una solida architettura. I provider di hosting condiviso normalmente si bloccano in queste circostanze.

## La Cache degli Oggetti per Siti Altamente Dinamici

Per quel che riguarda i siti ad iscrizione basati su WordPress, le normali impostazioni di caching di solito non sono sufficienti in quanto non sempre si riesce a trarne il dovuto vantaggio. È qui che **entra in gioco il caching degli oggetti**.

La cache degli oggetti memorizza i risultati delle query del database in modo che la prossima volta che quel particolare bit di dati sia necessario, possa essere servito dalla cache senza interrogare nuovamente il database. Ciò accelera i tempi di esecuzione di PHP e riduce il carico sul database. Questo diventa estremamente importante con i siti di membership! Con WordPress, potete implementare il caching degli oggetti in diversi modi:

1. Una soluzione di caching di terze parti come W3 Total Cache
2. **Redis (consigliato)**
3. Memcached

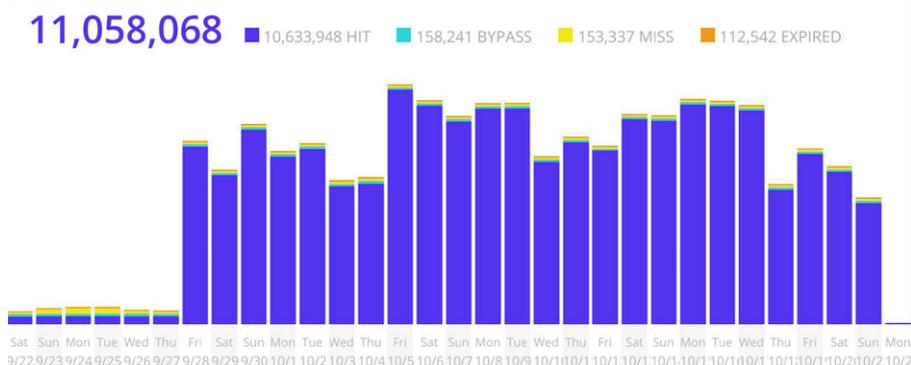
We offer Redis as an add-on at Kinsta so you can take full advantage of persistent object caching for your membership sites.

## Analizzare la Cache

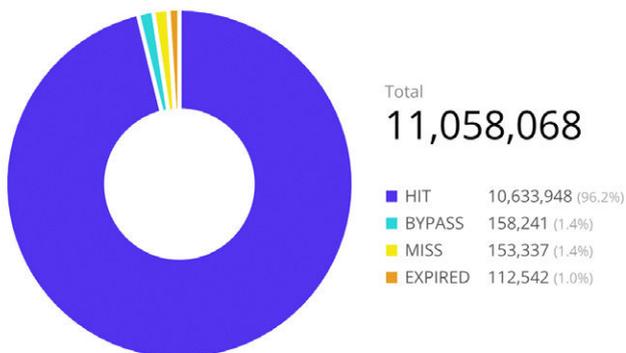
Ricordate l'intestazione `x-kinsta-cache` che abbiamo menzionato sopra? A seconda del provider di hosting o della soluzione di caching, l'intestazione potrebbe avere un nome leggermente diverso. Ogni volta che viene effettuata una richiesta dal vostro sito WordPress, tale intestazione assume un valore, come HIT, BYPASS, MISS e EXPIRED. Questo vi permette di vedere come si comporta la vostra cache.

È importante aumentare l'indice di utilizzo della cache del vostro sito WordPress perché l'ideale è che il vostro sito venga servito dalla cache il più possibile. In Kinsta potete analizzare i dati nel nostro tool di analisi di MyKinsta e nei log della cache di Kinsta per determinare se ci sono richieste GET che BYPASSano la cache, che invece potrebbero essere memorizzate nella cache, o richieste POST che potrebbero essere eliminate.

Lo stack del componente cache (come mostrato di seguito) consente di visualizzare lo stato di ciascuna richiesta, sia questa HIT, BYPASS, MISS o EXPIRED. Potete filtrare i dati delle ultime 24 ore, degli ultimi 7 o 30 giorni.



Il grafico dei componenti della cache vi offre una visione del vostro indicatore di caching. Più richieste vengono servite dalla cache, meglio è. Come potete vedere nell'esempio qui sotto, questo sito WordPress ha un indice cache HIT del 96,2%. È un buon valore!

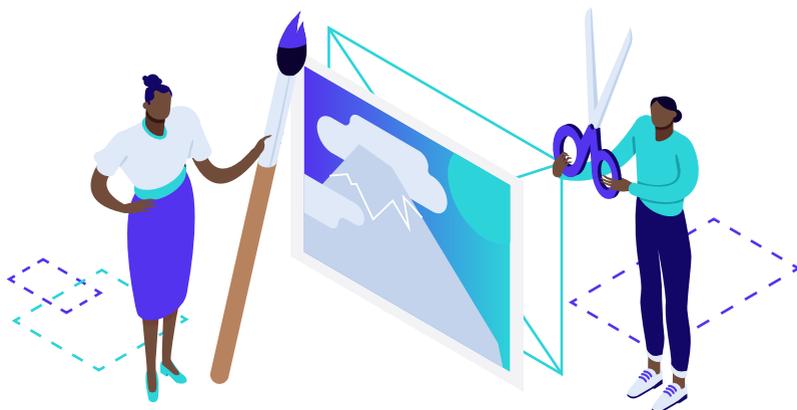


La sezione superiore, dedicata ai principali bypass della cache, consente di vedere quali richieste non vengono servite dalla cache. In genere questi possono includere CRON job, richieste admin-ajax, pagine di checkout ecommerce, query string e parametri UTM, ecc.

PATH	REQUESTS
/wp-cron.php?server_triggered_cronjob	28,977
/wp-admin/admin-ajax.php?action=edd_jilt_get_cart_data	22,279
/?ref=1	796
/	667
/sitemap_index.xml	490
/?ajax=1&ht-kb-search=1&s=	435
/account/	372
/checkout/	329
/?ref=19&campaign=bw	292

CAPITOLO 09:

# L'Ottimizzazione delle Immagini È un Must

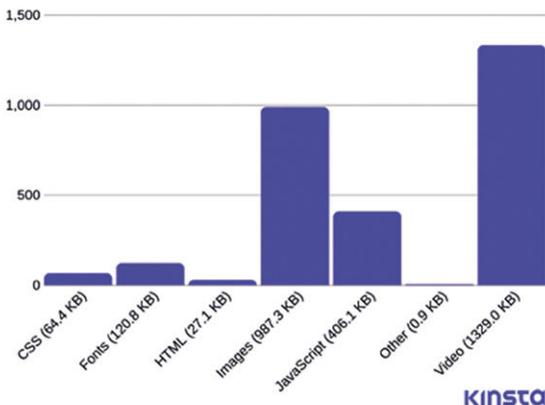


**L'ottimizzazione delle immagini è un'altra operazione semplice da attuare, ma con un impatto significativo sui tempi di caricamento complessivi delle pagine. E non è un optional; bisognerebbe farlo su ogni sito!**

Le immagini di grandi dimensioni rallentano le pagine Web e creano un'esperienza utente non ottimale. L'ottimizzazione delle immagini è il processo di riduzione delle dimensioni dei file, con il ricorso ad un plugin o ad uno script, con la conseguente riduzione del tempo di caricamento della pagina. I due metodi comunemente utilizzati sono la compressione con perdita di dati e la compressione senza perdita di dati.

Secondo HTTP Archive, a partire da agosto 2019, le immagini rappresentano in media **il 34% del peso totale di una pagina web**. Quindi, dopo i video, che sono molto più difficili da ottimizzare, le immagini sono di gran lunga la prima risorsa cui prestare attenzione! E sono più importanti di JavaScript, CSS e Font. E, ironia della sorte, un buon workflow di ottimizzazione delle immagini è semplicissimo da implementare, eppure molti proprietari di siti web ignorano questo aspetto.

## Byte Medi Per Pagina (KB)



Nel dicembre 2017, le immagini costituivano in media il 54% del peso complessivo di una pagina. Quindi sembra che il web nel suo complesso stia progredendo nell'ottimizzazione delle immagini! Ma il 34% è comunque un valore che non può essere ignorato. Se non avete contenuti video nel vostro sito web, le immagini sono probabilmente il vostro elemento critico n.1 per il peso della pagina.

*Le immagini rappresentano in media  
il 34% del peso complessivo  
di una pagina web. Ottimizzate!*

## Alla Ricerca dell'Equilibrio (tra Dimensioni dei File e Qualità)

L'obiettivo principale della formattazione delle immagini è quello trovare **equilibrio tra la minore dimensione dei file e una qualità accettabile**. C'è più di un modo per eseguire quasi tutte queste ottimizzazioni. Uno dei modi più semplici è quello di comprimere le immagini prima di caricarle su WordPress. Normalmente si può fare con uno strumento come Adobe Photoshop o Affinity Photo. O utilizzando la nuova app online Squoosh di Google. Tuttavia, queste operazioni possono anche essere eseguite automaticamente utilizzando i plugin, come vedremo più avanti.

Le due cose principali da considerare sono il **formato del file e il tipo di compressione** che si utilizza. Scegliendo la giusta combinazione di formato di file e tipo di compressione, potete ridurre le dimensioni dell'immagine di ben 5 volte. Dovrete sperimentare su ogni immagine o formato di file per vedere quale combinazione funziona meglio.

Prima di iniziare a modificare le immagini, assicuratevi di aver scelto il tipo di file migliore. Esistono diversi tipi di file che potete utilizzare:

- **PNG** – produce immagini di qualità superiore, ma ha anche una dimensione di file più grande. È stato creato come un formato immagine senza perdita di dati, anche se può anche essere con perdita di dati.
- **JPEG** – utilizza l'ottimizzazione lossy e lossless (con e senza perdita di dati). È possibile regolare il livello di qualità per avere un buon bilanciamento tra qualità e dimensioni del file.

In teoria, dovrete utilizzare JPEG (o JPG) per immagini con molti colori e PNG per immagini semplici.

E le GIF? Le GIF animate sono sempre divertenti, ma stroncano le prestazioni web. Molte GIF hanno dimensioni superiori a 1 MB. Consigliamo di destinarle ai social media e Slack. Se c'è una GIF di cui non potete fare a meno nel vostro post, dovrete sempre comprimerle in modo significativo.

## Compressione: Qualità contro Dimensione

Ecco un esempio di cosa può accadere se si comprime troppo un'immagine. Nel primo caso scegliamo un tasso di compressione molto basso, che si traduce nella massima qualità (ma con dimensioni maggiori del file). Nel secondo impostiamo un tasso di compressione molto alto, che si traduce in un'immagine di bassa qualità (ma con dimensioni del file minori). Nota: l'immagine originale non modificata è di 2,06 MB.

Come potete vedere, la prima immagine qui sopra è 590 KB. È abbastanza grande per una foto! Di solito è meglio cercare di mantenere il peso totale di una pagina web

al di sotto di 1 o 2 MB. 590 KB sarebbero già un quarto di questo valore. La seconda immagine è orribile, ma pesa solo 68 KB. Quello che dovete fare è trovare un giusto mezzo tra il tasso di compressione (qualità) e la dimensione del file.



Bassa compressione (alta qualità) JPG – 590 KB



Alta compressione (bassa qualità) JPG – 68 KB



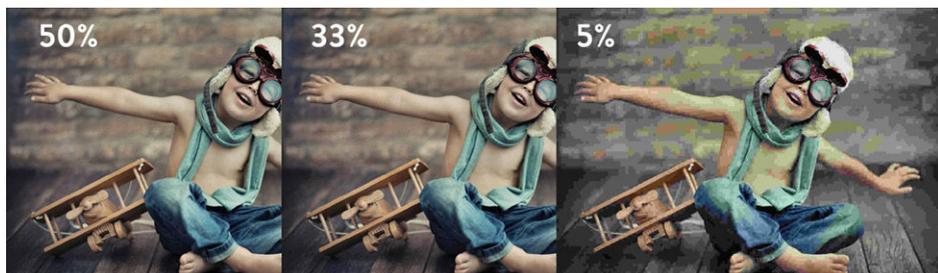
Compressione media (ottima qualità) JPG – 151 KB

Quindi abbiamo ripreso l'immagine a un tasso di compressione medio e, come potete vedere qui sotto, la qualità adesso è buona e la dimensione del file è di 151 KB, che è un valore accettabile per una foto ad alta risoluzione. L'immagine finale è quasi 4 volte più piccola della foto originale con una compressione bassa. Cerchiamo di mantenere la maggior parte delle nostre immagini sotto il livello dei 100 KB per avere le migliori prestazioni.

## Ottimizzazione Lossy vs. Lossless

Ci sono due tipi di compressione, uno con perdita di dati e l'altro senza perdita di dati.

La compressione Lossy comporta l'**eliminazione di alcuni dati dell'immagine**. Per questo motivo, potreste vedere un degrado (riduzione della qualità o quella cui alcuni si riferiscono come immagine pixelata). Quindi dovete stare attenti alla misura in cui state riducendo la vostra immagine. Non solo per la qualità, ma anche perché non potrete invertire il processo. Naturalmente, uno dei grandi vantaggi della compressione con perdita di dati, e il motivo per cui è uno dei metodi di compressione più utilizzati, è che è possibile **ridurre le dimensioni dei file in modo considerevole**.



La compressione senza perdita di dati, a differenza di quella con perdita di dati, **non riduce la qualità dell'immagine**. Com'è possibile? Di solito l'immagine viene compressa rimuovendo i metadati non necessari (dati generati automaticamente dal dispositivo che cattura l'immagine). Tuttavia, il maggiore inconveniente di questo metodo è che **non vedrete una riduzione significativa delle dimensioni dei file**. In altre parole, nel tempo si finirà con l'occupare molto spazio sul disco.

Dovrete sperimentare il metodo che va meglio per voi. Tuttavia, per la maggior parte degli utenti, consigliamo di utilizzare la compressione con perdita di dati, per il fatto che è possibile comprimere facilmente un'immagine di oltre il 70% (a volte anche di oltre il 90%!) senza grandi perdite di qualità. Moltiplicatelo per 15 immagini su una pagina e capirete quanto sia significativo per ridurre il tempo di caricamento del vostro sito.

## Plugin di Compressione delle Immagini

La grande notizia è che ci sono alcuni incredibili plugin per la compressione delle immagini di WordPress che potete utilizzare per automatizzare l'intero processo. Ecco alcuni plugin che consigliamo (sono tutti disponibili su WordPress.org):

- **Imagify** (lossy e lossless – ottimizza le immagini esternamente)
- **WP Smush** (lossy e lossless – ottimizza le immagini esternamente)
- **Optimole** (lossy e lossless – ottimizza le immagini esternamente)
- **EWWW Cloud** (lossy e lossless – ottimizza le immagini esternamente)
- **ShortPixel** (lossy e lossless – ottimizza le immagini esternamente)

La cosa più importante da considerare quando si sceglie un plugin per l'ottimizzazione delle immagini è **sceglierne uno che comprima e ottimizzi le immagini esternamente** sui server dello sviluppatore. Questo riduce il carico sul vostro sito, e lo fanno tutti i plugin elencati sopra.

Se siete curiosi, nel sito web di Kinsta utilizziamo il plugin Imagify. Questo comprime automaticamente le immagini quando le carichiamo nella libreria multimediale di WordPress. Quindi non dobbiamo mai preoccuparci di nulla. Con il passare del tempo potrete farvi un'idea del livello preciso di compressione che desiderate. Il plugin offre un livello di compressione Normale, Aggressivo e Ultra.

Su Kinsta utilizziamo la modalità Aggressiva e in genere osserviamo un risparmio del 60-70% a seconda dell'immagine. Nota: utilizziamo molti più file PNG che JPEG perché la maggior parte delle nostre immagini sono icone e illustrazioni, non foto.



Di quanto sarà più veloce il vostro sito WordPress se comprimate le immagini? Tutto dipende dalle dimensioni delle immagini originali e da come risultano dopo la compressione. Tuttavia, abbiamo eseguito alcuni test di velocità e abbiamo scoperto che una buona compressione delle immagini può **ridurre i tempi di caricamento della pagina di oltre l'80%**!

## Lazy Loading

Se avete molte immagini, potreste prendere in considerazione la possibilità del caricamento differito. Questa è una tecnica di ottimizzazione che consente di caricare

i contenuti visibili ritardando il download e il rendering dei contenuti che appaiono “below the fold” (la parte non visibile della pagina).

Il lazy loading funziona così:

- Il browser crea il DOM della pagina Web senza scaricare le immagini e precaricare i video.
- Viene utilizzato JavaScript per determinare quali immagini scaricare e quali video precaricare in base al contenuto inizialmente visibile quando carica la pagina. Quelle immagini e quei video sono scaricati e precaricati nel modo più appropriato.
- Il download e il rendering dei video aggiuntivi sono ritardati fino a quando un visitatore del sito scorre la pagina verso il basso e viene visualizzato il contenuto aggiuntivo.

Il risultato finale è che le immagini non vengono scaricate e i video caricati finché non sono effettivamente necessari. Ciò può fornire un significativo miglioramento delle prestazioni per i siti che includono molte immagini ad alta risoluzione e incorporano dei video.

Per fortuna ci sono molti ottimi plugin gratuiti per WordPress che rendono abbastanza semplice l’abilitazione del lazy loading (sono tutti disponibili su WordPress.org):

- **a3 Lazy Load**
- **Lazy Load by WP Rocket**
- **Lazy Load for Videos**

## Altri Suggerimenti per l’Ottimizzazione delle Immagini

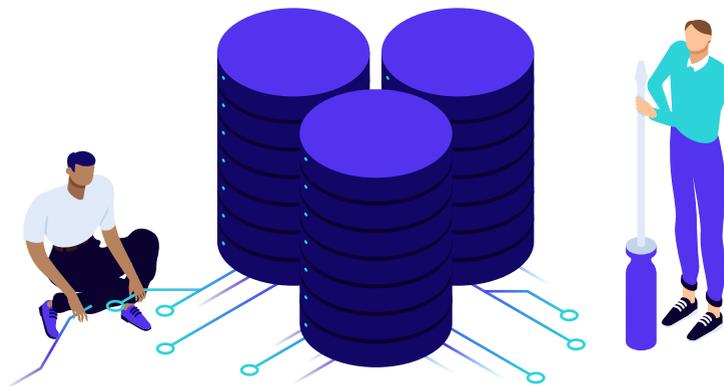
Ecco alcuni consigli conclusivi per l’ottimizzazione delle immagini.

- Sono finiti i giorni in cui si caricavano immagini esclusivamente delle dimensioni della larghezza della colonna o della DIV. Le immagini reattive sono predefinite in WordPress (dalla versione 4.4) e saranno automaticamente visualizzate nelle dimensioni più piccole per gli utenti mobili.

- Gli SVG possono costituire un'altra ottima alternativa all'utilizzo delle immagini. Tutte le illustrazioni disegnate a mano che vedete sul sito web di Kinsta sono SVG (vettoriali). Gli SVG sono in genere molto più piccoli nella dimensione del file, sebbene non sia sempre così.
- Usate gli icon font invece di posizionare il testo all'interno delle immagini – hanno un aspetto migliore quando vengono ridimensionati e occupano meno spazio. E se utilizzate un generatore di font, potete ottimizzarli ancora di più. Date un'occhiata a come abbiamo diminuito le dimensioni dei nostri icon font di **un sorprendente 97,59%** utilizzando un generatore di font.
- Potreste anche dare un'occhiata a WebP, un nuovo formato immagine di Google che sta rapidamente guadagnando consensi.

CAPITOLO 10:

# Ottimizzare il Database



**Di seguito sono riportati alcuni suggerimenti per ottimizzare il Database di WordPress. Proprio come un'auto, il vostro database ha bisogno di manutenzione, perché nel tempo può gonfiarsi.**

I siti ad iscrizione rendono la manutenzione particolarmente complicata, poiché di solito **generano query più complesse**, che a loro volta aumentano la latenza nel recupero delle informazioni dal database MySQL. Molto di questo lavoro è dovuto a tutte le parti dinamiche e alle grandi quantità di dati che caratterizzano questi siti. Ciò potrebbe verificarsi anche in siti che fanno molto affidamento sulle query di ricerca per la navigazione, o che utilizzano `WP_Query`.

Per non dire, poi, che vi trovate con grandi quantità di utenti che eseguono di continuo e in contemporanea query sul database.

## **Utilizzare InnoDB Come Motore di Archiviazione MySQL**

Molti siti meno recenti utilizzano ancora il motore di archiviazione MyISAM nel loro database. Negli ultimi anni, InnoDB ha dimostrato di avere migliori prestazioni e di essere più affidabile.

*InnoDB è come l'olio sintetico, mentre  
MyISAM si è stabilizzato.*

Ecco alcuni vantaggi di InnoDB rispetto a MyISAM:

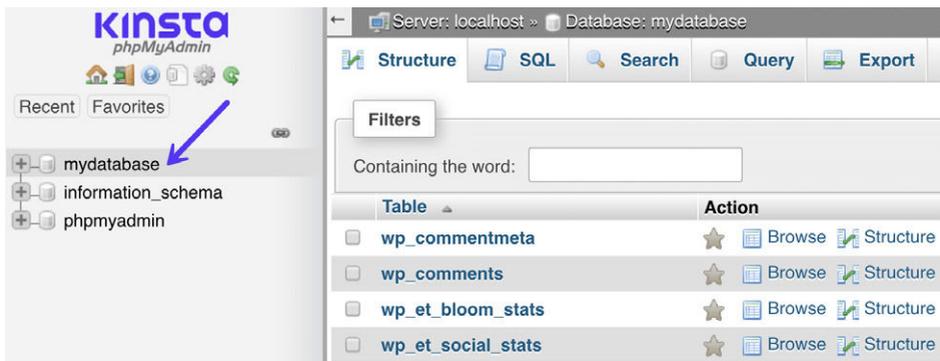
- InnoDB dispone del **blocco a livello di riga**. MyISAM ha solo un blocco completo a livello di tabella. Ciò consente una più rapida esecuzione delle query.
- InnoDB dispone di quella che viene definita integrità referenziale, che implica il supporto di **chiavi esterne** (RDBMS) e vincoli di relazione, mentre MyISAM non ne dispone (DMBS).
- InnoDB supporta le **transazioni**, il che significa che potete eseguire il commit e il rollback. MyISAM no.
- InnoDB è più affidabile in quanto utilizza i registri transazionali per il ripristino automatico. MyISAM no.

Quindi ora vi starete chiedendo se avete in esecuzione InnoDB o MyISAM. Se avete un sito WordPress abbastanza recente, è probabile che stiate già utilizzando InnoDB come motore di archiviazione MySQL. Ma con i vecchi siti WordPress dovrete fare un rapido controllo. Alcuni siti potrebbero anche avere tabelle MyISAM e InnoDB miste e abbinare, caso in cui potrete vedere dei miglioramenti convertendo tutte le tabelle.

Seguite questi semplici passaggi per fare una verifica.

## Passo 1

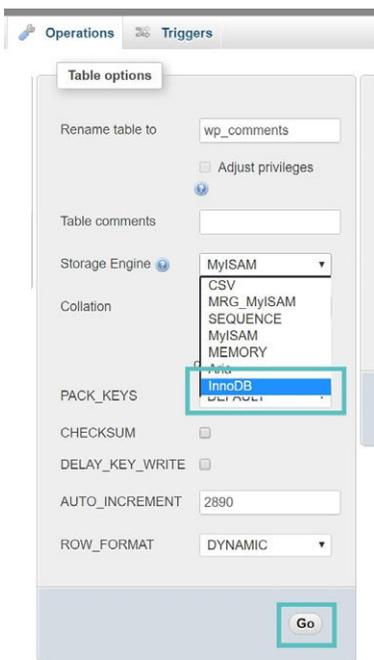
Accedete a phpMyAdmin e fate clic sul vostro database MySQL.



## Passo 2

Date una rapida occhiata o ordinate la colonna “Tipo” e potrete vedere quali tipi di motore di archiviazione utilizzano le vostre tabelle. Nell’esempio qui sotto, potete vedere che due tabelle stanno ancora utilizzando MyISAM.

Icona	Struttura	Search	Insert	Empty	Drop	Count	Engine	Collation	Size
📄	🔍	🔍	➕	🗑️	🚫	0	InnoDB	utf8_unicode_ci	64 KiB
📄	🔍	🔍	➕	🗑️	🚫	0	InnoDB	utf8_unicode_ci	48 KiB
📄	🔍	🔍	➕	🗑️	🚫	0	InnoDB	utf8_unicode_ci	64 KiB
📄	🔍	🔍	➕	🗑️	🚫	0	InnoDB	utf8_unicode_ci	80 KiB
📄	🔍	🔍	➕	🗑️	🚫	0	InnoDB	utf8_unicode_ci	48 KiB
📄	🔍	🔍	➕	🗑️	🚫	0	InnoDB	utf8mb4_unicode_ci	32 KiB
📄	🔍	🔍	➕	🗑️	🚫	113	MyISAM	utf8mb4_unicode_ci	80 KiB
📄	🔍	🔍	➕	🗑️	🚫	471	MyISAM	utf8mb4_unicode_ci	208 KiB
📄	🔍	🔍	➕	🗑️	🚫	0	InnoDB	utf8mb4_unicode_ci	48 KiB
📄	🔍	🔍	➕	🗑️	🚫	1	InnoDB	utf8mb4_unicode_ci	16 KiB



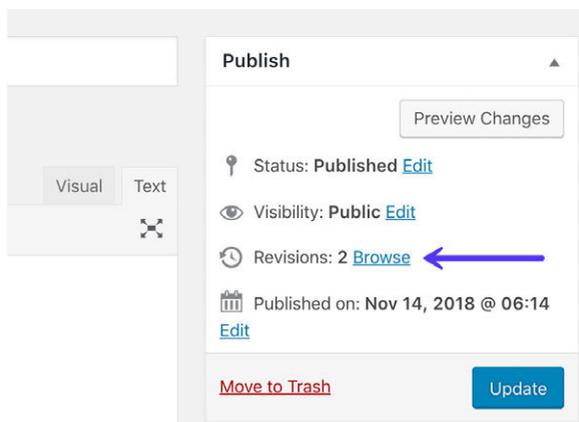
Se ne avete trovata qualcuna, allora forse è il momento di passarle a InnoDB. Raccomandiamo sempre di contattare l’host e chiedere se possono farlo loro per voi. Da Kinsta, le tabelle del database di ogni cliente vengono automaticamente convertite in InnoDB dal nostro team delle migrazioni.

Ma potete anche convertire manualmente le tabelle da phpMyAdmin. Basta fare clic sulla tabella myISAM, poi sulla scheda “Operazioni”, e cambiare il motore di archiviazione.

Se siete sviluppatori, una alternativa è convertire le tabelle con WP-CLI.

## Eliminare e Limitare le Revisioni di Post e Pagine

Ogni volta che salvate una pagina o un post in WordPress, viene creata una cosiddetta revisione. Questo succede sia per le bozze che per i post già pubblicati che vengono aggiornati. Le revisioni possono essere utili nel caso in cui sia necessario ripristinare una versione precedente dei vostri contenuti.

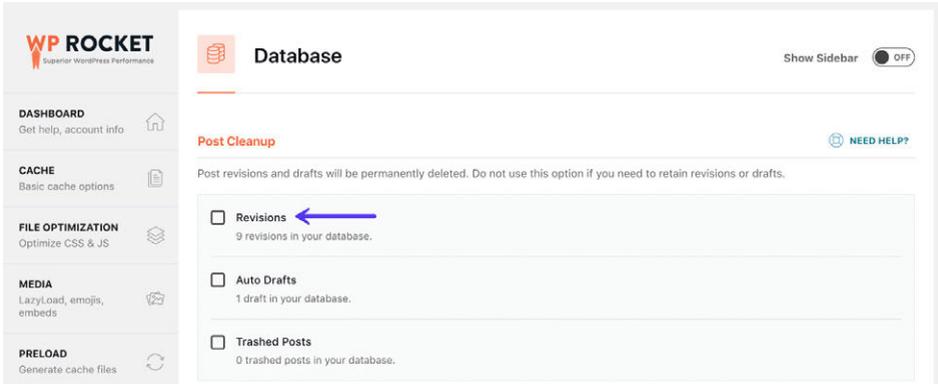


Tuttavia, le revisioni possono anche danneggiare le prestazioni del vostro sito WordPress. Su siti di grandi dimensioni, questo può comportare la rapida aggiunta nel database di migliaia di righe che potrebbero non essere assolutamente necessarie. E più righe avete, maggiore è la dimensione del vostro database, il quale occupa spazio di archiviazione. Sebbene gli indici siano stati creati proprio a questo scopo, abbiamo comunque visto questo problema paralizzare i siti WordPress. Ci sono un paio di cose che potreste fare.

### 1. Eliminate le Vecchie Revisioni

Se avete un sito WordPress più vecchio con un sacco di pagine e post, potrebbe essere il momento di fare una rapida pulizia ed eliminare le vecchie revisioni. Potete farlo con MySQL, ma con tutti i cattivi frammenti di codice che circolano sul web, vi consigliamo di fare prima un backup del sito e di utilizzare un plugin gratuito come WP-Sweep.

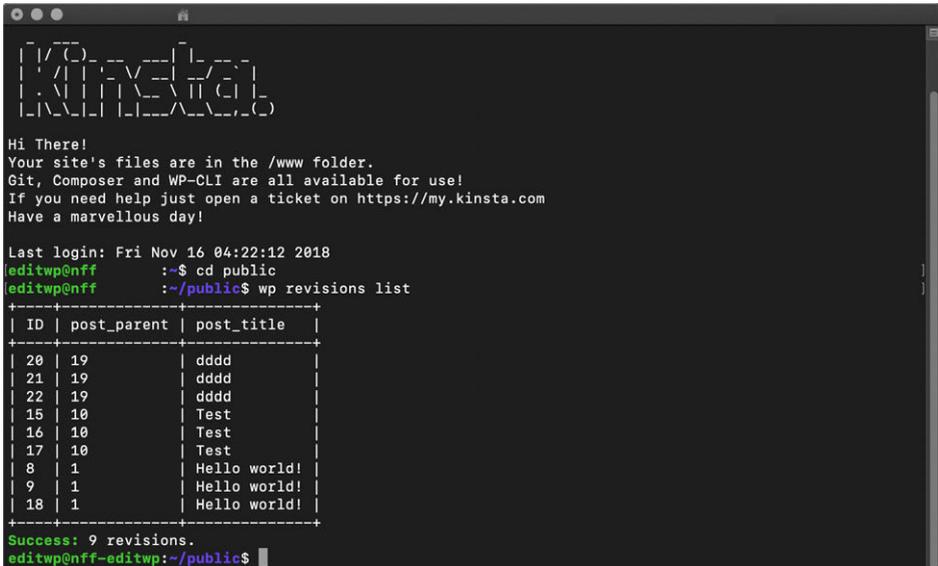
Anche un altro dei nostri plugin preferiti, WP Rocket, dispone di una funzione di ottimizzazione del database per cancellare le revisioni.



Se avete confidenza con WP-CLI, avete a disposizione un paio di comandi.

Accedete al vostro server tramite SSH ed eseguite il seguente comando per vedere il numero di revisioni presenti nel database.

```
wp revisions list
```



Se si verifica un errore, potrebbe essere necessario installare prima il pacchetto `wp-revisions-cli` con il seguente comando:

```
wp package install trepmal/wp-revisions-cli
```

Potrete quindi, eseguire questo comando per ripulire le revisioni:

```
wp revisions clean
```

## 2. Limitare le Revisioni

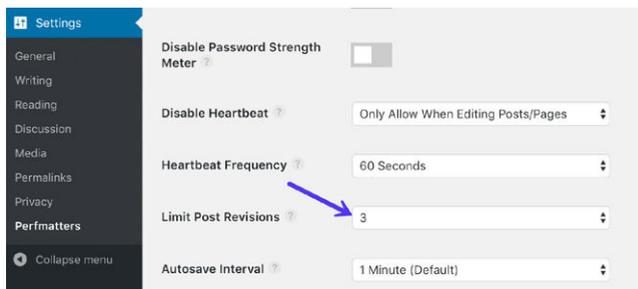
Un'altra buona strategia, che tra l'altro adottiamo da Kinsta, è quella di limitare il numero di revisioni che possono essere memorizzate per ogni post o pagina. Anche impostare questo valore a dieci eviterà che le revisioni vi sfuggano di mano, soprattutto se fate un sacco di aggiornamenti.

Per limitare le revisioni, potete aggiungere il seguente codice al vostro file `wp-config.php`. Il codice qui sotto deve essere inserito sopra `'ABSPATH'`, altrimenti non funzionerà. Potete cambiare il valore in base al numero di revisioni che desiderate conservare nel database.

```
define( 'WP_POST_REVISIONS', 10 );
```

```
59
60 /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66 $table_prefix = 'wp_';
67
68 define( 'WP_POST_REVISIONS', 10 ); ←
69
70 /* That's all, stop editing! Happy blogging. */
71
72 /** Absolute path to the WordPress directory. */
73 if ( ! defined( 'ABSPATH' ) )
74     define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76 /** Sets up WordPress vars and included files. */
77 require_once ABSPATH . 'wp-settings.php';
78
```

Line 28, Column 1



Limitare le revisioni con Perfmatters

In alternativa, per limitare le revisioni potete utilizzare un plugin come Perfmatters (disponibile all'indirizzo [perfmatters.io](https://perfmatters.io)).

### 3. Disabilitare le Revisioni

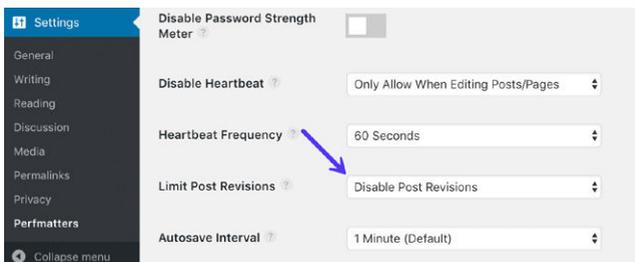
E, ultimo ma non meno importante, potete anche disabilitare completamente le revisioni sul vostro sito. Se scegliete di seguire questa strada, vi consigliamo vivamente di scegliere la prima soluzione descritta sopra per cancellare le revisioni e poi disabilitarle in seguito. In questo modo il vostro database sarà completamente libero da tutte le vecchie revisioni e non ne verranno aggiunte di nuove successivamente.

Per disabilitare le revisioni, potete aggiungere il seguente codice al vostro file `wp-config.php`. Il codice qui sotto deve essere inserito sopra `'ABSPATH'`, altrimenti non funzionerà.

```
define('WP_POST_REVISIONS', false);
```

```
60 /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66 $table_prefix = 'wp_';
67
68 define('WP_POST_REVISIONS', false); ←
69
70 /* That's all, stop editing! Happy blogging. */
71
72 /** Absolute path to the WordPress directory. */
73 if ( ! defined( 'ABSPATH' ) )
74     define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76 /** Sets up WordPress vars and included files. */
77 require_once ABSPATH . 'wp-settings.php';
78
```

Line 33, Column 34



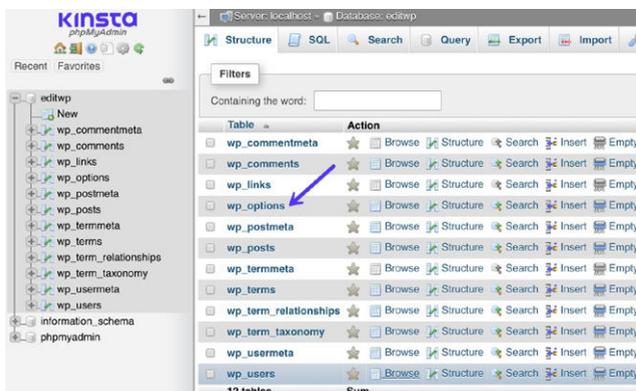
Disabilitare le revisioni con Perfmatters

In alternativa, per disabilitare le revisioni potete utilizzare sempre un plugin come Perfmatters

## Ripulire la Tabella wp\_options e i Dati a Caricamento Automatico

La tabella `wp_options` viene spesso trascurata per quanto riguarda le prestazioni complessive di WordPress e del database. Soprattutto su siti più vecchi e di maggiori dimensioni, ciò può facilmente allungare i tempi delle query sul vostro sito a causa di dati a caricamento automatico, lasciati indietro da plugin e temi di terze parti. Fidatevi, è una cosa che vediamo tutti i giorni!

La tabella `wp_options` contiene ogni sorta di dati del vostro sito WordPress, come:



- URL del sito, URL della home page, email amministratore, categoria predefinita, post per pagina, formato dell'ora, ecc.
- Impostazioni dei plugin, dei temi, dei widget
- Dati memorizzati temporaneamente nella cache

Questa tabella contiene i seguenti campi (colonne):

- option\_id
- option\_name
- option\_value
- **autoload** (questo è il campo di cui ci dobbiamo preoccupare per le prestazioni)

+ Options

	option_id	option_name	option_value	autoload
<input type="checkbox"/> Edit Copy Delete	1	siteurl	https://editwp.com	yes
<input type="checkbox"/> Edit Copy Delete	2	home	https://editwp.com	yes
<input type="checkbox"/> Edit Copy Delete	3	blogname	editwp	yes
<input type="checkbox"/> Edit Copy Delete	4	blogdescription	Just another WordPress site	yes

Una cosa importante da dire a proposito della tabella `wp_options` riguarda il campo **autoload**. Questo campo contiene un valore `yes` o un valore `no` (flag) e essenzialmente controlla se è caricato o meno dalla funzione `wp_load_alloptions()`. I dati a caricamento automatico sono **dati che vengono caricati su ogni pagina** del vostro sito WordPress. Proprio come abbiamo visto per disabilitare certi script dal caricamento in tutto il sito, qui possiamo applicare la stessa idea. L'attributo `autoload` è impostato su `“yes”` di default per gli sviluppatori, ma non tutti i plugin dovrebbero teoricamente caricare i loro dati su ogni pagina.

Il problema in cui potrebbero imbattersi i siti WordPress nasce quando c'è una grande quantità di dati a caricamento automatico nella tabella `wp_options`. Questo è in genere il risultato di quanto segue:

- I dati vengono caricati automaticamente da un plugin quando il valore dovrebbe essere impostato su `“no”`. Un buon esempio potrebbe essere quello di un plugin per i moduli di contatto. Ha davvero bisogno di caricare i dati su ogni pagina del sito o solo sulla pagina dei contatti?
- Plugin o temi che sono stati rimossi dal sito WordPress, ma le cui opzioni sono rimaste nella tabella `wp_options`. Ciò potrebbe significare che i dati a caricamento automatico non necessari vengono prelevati ad ogni richiesta.
- Sviluppatori di temi e plugin caricano i dati nella tabella `wp_options` invece di utilizzare le proprie tabelle. Ci sono opinioni discordanti su questo punto, dato che alcuni sviluppatori preferiscono plugin che non creano tabelle aggiuntive.

Tuttavia, la tabella `wp_options` non è stata progettata per contenere migliaia di righe.

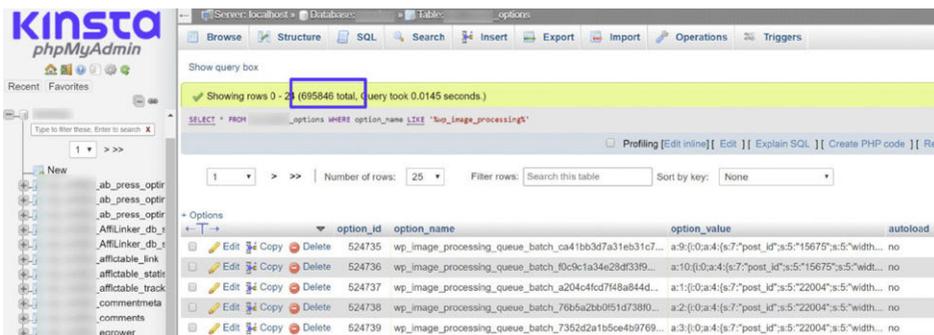
Quando i dati a caricamento automatico possono considerarsi troppi? Naturalmente non c'è una regola fissa, ma, in teoria, dovrebbe essere un valore compreso tra 300 KB e 1 MB. Una volta che cominciate ad avvicinarvi ai 3-5 MB o più, probabilmente ci saranno cose che dovrebbero essere ottimizzate o rimosse dal caricamento automatico. E qualsiasi valore al di sopra di 10 MB dovrebbe essere corretto immediatamente. Questo non deve necessariamente causare delle anomalie, ma è qualcosa da tenere in considerazione.

La pulizia dei dati a caricamento automatico è un'attività in cui aiutiamo sempre i nostri clienti da Kinsta.

*Quando è stata l'ultima volta che avete ripulito la vostra tabella `wp_options`? Già ... lo pensavamo. Fatelo subito!*

## Ripulire i Dati Transient

A meno che non stiate utilizzando una cache di oggetti, WordPress memorizza i record temporanei nella tabella `wp_options`. In genere a questi viene assegnato un tempo di scadenza e dovrebbero scomparire nel tempo. Tuttavia, questo non succede sempre. Abbiamo visto alcuni database in cui ci sono migliaia di vecchi record transient. Infatti, su un sito abbiamo dovuto occuparci di alcuni record transient corrotti che avevano generato **oltre 695.000 righe** nella tabella `wp_options`. Wow!



The screenshot shows the phpMyAdmin interface with the following details:

- Server: localhost - Database: wp - Table: wp\_options
- Query: `SELECT * FROM wp_options WHERE option_name LIKE 'wp_image_processing%'`
- Result: Showing rows 0 - 25 (695046 total, query took 0.0145 seconds.)
- Table structure:

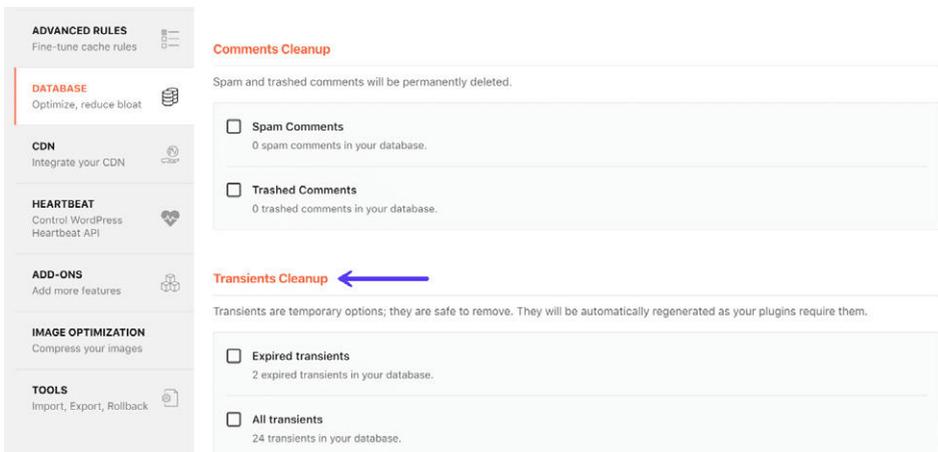
	option_id	option_name	option_value	autoload
[Edit] [Copy] [Delete]	524735	wp_image_processing_queue_batch_ca41bb3d7a31eb31c7...	a:9:{0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width..."	no
[Edit] [Copy] [Delete]	524736	wp_image_processing_queue_batch_f0c9c1a34e28df339...	a:10:{0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width..."	no
[Edit] [Copy] [Delete]	524737	wp_image_processing_queue_batch_a204c4fd7f48a844d...	a:1:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width..."	no
[Edit] [Copy] [Delete]	524738	wp_image_processing_queue_batch_76b5a2bb0f51d7380...	a:2:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width..."	no
[Edit] [Copy] [Delete]	524739	wp_image_processing_queue_batch_7352d2a1b5ce4b9769...	a:3:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width..."	no

I dati transient non devono essere caricati automaticamente di default. È possibile utilizzare una query come la seguente per vedere se ci sono dati transitori caricati automaticamente.

```
SELECT *
FROM `wp_options`
WHERE `autoload` = 'yes'
AND `option_name` LIKE '%transient%'
```

Una soluzione migliore e più sicura potrebbe essere quella di utilizzare un plugin gratuito come Transient Cleaner o Delete Expired Transients (entrambi disponibili su WordPress.org) che possono eliminare esclusivamente i dati transient scaduti dalla vostra tabella `wp_options`. Tuttavia, sembra che ora ci sia una funzione in WordPress, aggiunta nella versione 4.9, che fa pulizia dei dati transient scaduti. Quindi è auspicabile che ora la pulizia avvenga automaticamente sul vostro sito.

Anche WP Rocket permette di ripulire i dati transient nelle opzioni di ottimizzazione del database.



## Ripulire le Sessioni di WordPress

Un altro problema che abbiamo riscontrato di frequente è che a volte i cron job non vengono sincronizzati o non vengono attivati correttamente, pertanto le sessioni non

vengono ripulite. Si può finire per ottenere una quantità enorme di righe `_wp_session_` nel database. Nell'esempio riportato di seguito, il sito in questione ha accumulato oltre 3 milioni di righe nella tabella `wp_options`. E la tabella aveva superato i 600 MB.

Table Name	Engine	Collation	Charset	Rows	Size
wp_links	InnoDB	utf8mb4_unicode_ci	32 K	0	32 K
wp_options	InnoDB	utf8mb4_unicode_ci	620.9 M	~3,419,339	620.9 M
wp_postmeta	InnoDB	utf8mb4_unicode_ci	9.1 M	12,986	9.1 M
wp_posts	InnoDB	utf8mb4_unicode_ci	4 M	2,367	4 M

Potreste utilizzare una query come quella qui sotto per vedere se avete questo problema:

```
SELECT *
FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'
```

<< < 6491 > >> | Number of rows: 500 | Filter rows: Search this table | Sort by key: None

option_id	option_name	option_value	autoload
3301835	_wp_session_254a6d16b4400d5aabf3c48b41b72d2b	a:1:{s:10:"edd_errors";s:0:""} no	no
3301836	_wp_session_expires_254a6d16b4400d5aabf3c48b41b72d...	1520801306	no
3301837	_wp_session_747c86aefae52149d1068a60f240249f	a:1:{s:10:"edd_errors";s:0:""} no	no
3301838	_wp_session_expires_747c86aefae52149d1068a60f24024...	1520801306	no
3301839	_wp_session_17aa53659273873116d83993bae7cfd8	a:1:{s:10:"edd_errors";s:0:""} no	no
3301840	_wp_session_expires_17aa53659273873116d83993bae7cf...	1520801306	no
3301841	_wp_session_f3f29f822eec97951504e4d66afcd081	a:1:{s:10:"edd_errors";s:0:""} no	no
3301842	_wp_session_expires_f3f29f822eec97951504e4d66afcd0...	1520801306	no
3301843	_wp_session_2d4be94abb1e756ee229506e1c26e60e	a:1:{s:10:"edd_errors";s:0:""} no	no

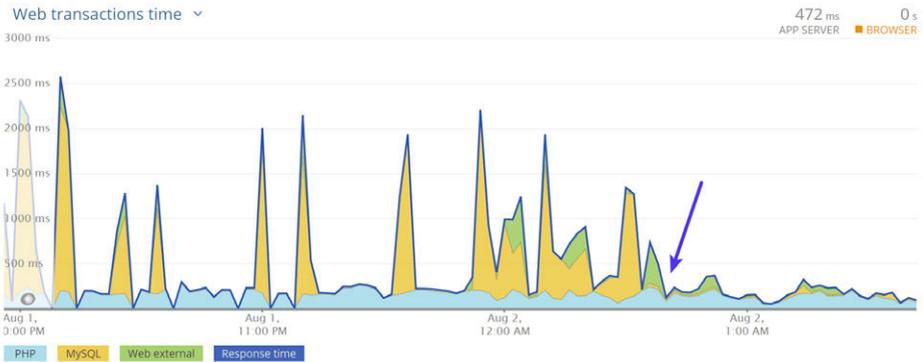
Nella maggior parte dei casi è possibile quindi eliminare queste righe (come dovrebbe aver fatto un cron job) con il seguente comando:

```
DELETE FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'
```

Dopo aver ripulito tutte le righe `_wp_session_` la tabella conteneva meno di 1.000 righe ed era stata ridotta a 11 MB.

Table Name	Engine	Collation	Charset	Rows	Size
wp_links	InnoDB	utf8mb4_unicode_ci	32 K	0	32 K
wp_options	InnoDB	utf8mb4_unicode_ci	11.5 M	672	11.5 M
wp_postmeta	InnoDB	utf8mb4_unicode_ci	9.1 M	12,986	9.1 M
wp_posts	InnoDB	utf8mb4_unicode_ci	4 M	2,367	4 M

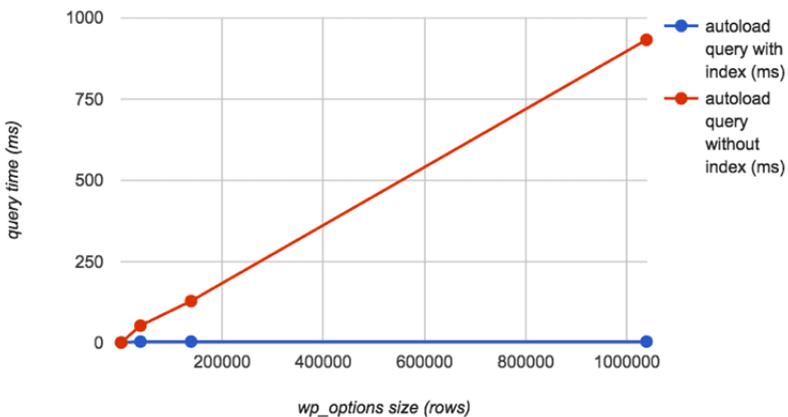
La query ha anche eliminato i picchi che il sito stava registrando in MySQL.



## Aggiungere un Index ad Autoload

Se la pulizia della tabella `wp_options` non fosse sufficiente, potreste provare ad aggiungere un “indice” al campo `autoload`. Questo in sostanza può favorire una ricerca più efficiente. Il fantastico team di 10up ha eseguito alcuni test su una tabella `wp_options` con un numero tipico di record caricati automaticamente, per mostrare come l’aggiunta di un indice di `autoload` alle query su `wp_options` possa migliorare le prestazioni

## Dimensioni di `wp_options` rispetto al tempo di query



Origine immagine: 10up

## Usare Redis come Cache di Oggetti Persistente per WordPress

Redis è un archivio di strutture dati in-memory, open-source. Nel contesto di WordPress, Redis può essere utilizzato per memorizzare i valori generati dalla cache di oggetti nativi di WordPress in modo permanente, così che gli oggetti memorizzati nella cache possano essere riutilizzati tra i caricamenti di pagina.



L'utilizzo di una cache di oggetti persistente come Redis consente il **riutilizzo di oggetti memorizzati nella cache** piuttosto che interrogare il database MySQL un'altra volta per lo stesso oggetto. Il risultato è che Redis può ridurre il carico sul database MySQL di un sito Web, riducendo contemporaneamente i tempi di risposta del sito e aumentando la capacità del sito di scalare e gestire traffico aggiuntivo.

Siti web altamente dinamici (WooCommerce, siti di adesione, forum, forum di discussione, blog con sistemi di commenti estremamente attivi) che non possono fare buon uso del caching delle pagine sono potenziali candidati per una soluzione di caching degli oggetti persistente come Redis.

Se siete clienti di Kinsta, ricordate che offriamo un add-on per Redis.

## Utilizzare Elasticsearch per Velocizzare la Ricerca in WordPress

Elasticsearch è un motore di ricerca full-text open-source. Viene utilizzato per indicizzare i dati e cercarli in modo incredibilmente rapido.

Nel contesto di WordPress, Elasticsearch può essere utilizzato per **velocizzare l'interrogazione del database di WordPress**. Ciò avviene con la creazione di un indice del contenuto del database del sito e quindi utilizzando Elasticsearch per cercare in questo indice in modo molto più rapido di quanto possa fare una query MySQL che esegue la stessa ricerca.



Se avete il tempo e le conoscenze necessarie, Elasticsearch può essere integrato con un sito WordPress da uno sviluppatore molto esperto sia WordPress che di Elasticsearch. Se il vostro sito fa un uso relativamente normale di WP\_Query, Elasticsearch può anche essere integrato installando ElasticPress, un plugin gratuito per WordPress sviluppat da 10up, disponibile su WordPress.org, che si integra automaticamente con l'oggetto WP\_Query in modo da generare i risultati con Elasticsearch invece che con MySQL.

Qualsiasi sito che faccia un uso consistente di WP\_Query può trovare utile Elasticsearch. Ecco alcuni esempi di siti che possono beneficiare di Elasticsearch:

- Siti in cui la ricerca è il mezzo principale di navigazione.
- Siti WooCommerce con un enorme numero di ordini in cui gli amministratori effettuano regolarmente ricerche negli elenchi degli ordini.
- Qualsiasi sito con un gran numero di posts in cui le query MySQL generano risultati inaccettabilmente lenti.

Proprio come con Redis, abbiamo un add-on Elasticsearch. Ecco come aggiungere Elasticsearch al vostro piano.

## **Disabilitare Funzionalità Non Critiche che Fanno Uso Intensivo del Database**

Potrebbe sembrare un po' ovvio, ma si può fare un'enorme differenza se si disabilitano i plugin non critici e le funzionalità del tema che fanno uso intensivo del database.

- I widget e i plugin per post popolari e/o correlati sono terribili. Di solito generano query pesanti su tutto il sito.
- I plugin di ottimizzazione delle immagini che comprimono le immagini utilizzando il vostro server. Dovreste sempre utilizzare un plugin di ottimizzazione delle immagini che esegua i suoi task esternamente.

Se visitate il blog di Kinsta e scorrete verso il basso fino alla fine di un post, noterete la sezione che definiamo articoli correlati “selezionati”. Questi sono selezionati da noi manualmente e assegnati al post. Ciò riduce la query a quasi nulla e non danneggerà le prestazioni dell'intero sito. Richiede maggior lavoro? Sì, ma offre un vantaggio in più, in quanto vi dà la possibilità di scegliere ciò che volete far vedere ai vostri lettori.

### Hand-picked related articles



**Working Remotely – Everything You Need to Know Beforehand**



**The 53 Best Tools for Freelancers to Scale a (Real) Business**



**Where and How to Hire a WordPress Developer**

Quindi, come lo abbiamo realizzato? Abbiamo utilizzato il fantastico plugin Advanced Custom Fields ed abbiamo assegnato questi campi al post type del nostro blog. Questo ci permette di cercare e assegnare qualsiasi contenuto correlato a ciascuno dei nostri post (come si può vedere qui sotto).

**Post Settings**

**Related Posts**  
*Choose hand-picked related posts here. If none are chosen (or fewer than needed for the display) we will fill the rest up with posts from within the same category/tags.*

Search...

10 Best WordPress Directory Plugins to Organize Your Listings (draft)	How to Score 100/100 in Google PageSpeed Insights with WordPress
10 Best WordPress Security Plugins to Lock out the Bad Guys	10 Unique WordPress Ecommerce Strategies to Skyrocket Sales
10 High-converting WordPress Landing Page Plugins With Beauty and Optimization Combined (draft)	Conversion Rate Optimization Tips – 12 Easy Ways to Boost Sales
10 Stunning Places to Find Free Images For WordPress	
10 Things Not To Do In PHP 7	



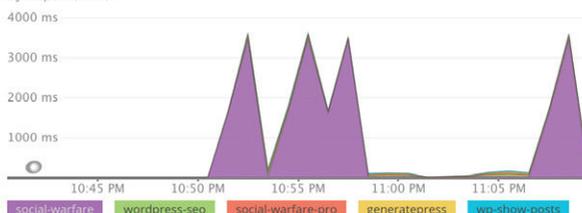
Vi consigliamo anche di evitare plugin che aggiungono un contatore di visite/post al vostro sito, a meno che non ne abbiate assolutamente bisogno. Ad esempio, evitate cose come “792 post” accanto all’avatar di un utente nei post del forum o “5.243 visualizzazioni” negli elenchi dei post del forum. Quando avete una lunga discussione, questi contatori avranno un enorme peso sul vostro database. In generale, è bene ridurre al minimo l’uso di contatori e utilizzarli solo se necessario.

Questo vale anche per molti contatori sociali. Ad esempio, nel sito qui sotto potete vedere che il tempo di risposta del famoso plugin Social Warfare è 30 volte superiore al plugin sottostante. Il caching è abilitato, ma ovviamente questo plugin ha un considerevole peso sulle performance. Dopo aver disabilitato il plugin, i tempi di caricamento sono immediatamente diminuiti e la reattività della dashboard di WordPress è migliorata.

Sort by

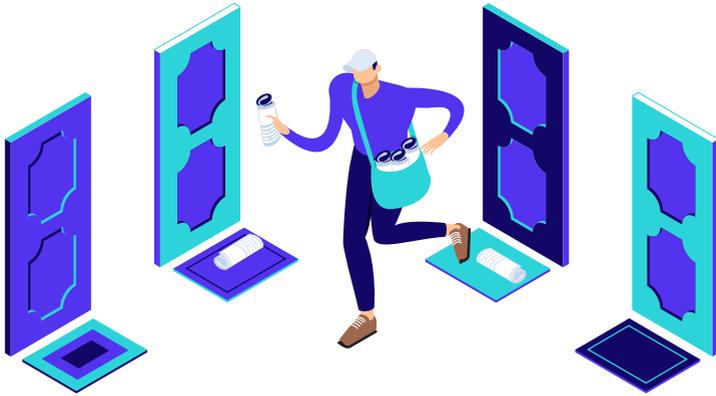
social-warfare	18,9 sec
wordpress-seo	624 ms
social-warfare-pro	449 ms
generatepress	425 ms
wp-show-posts	389 ms
aaawp	356 ms
gp-premium	229 ms
nogluten	160 ms
imagify	115 ms
wp-review	107 ms

Top 5 plugins and themes  
by response time



CAPITOLO 11:

# Utilizzare un Content Delivery Network (CDN)



**CDN è l'abbreviazione di content delivery network. Si tratta di una rete di server (noti anche come POP) situati in tutto il mondo. Sono progettati per ospitare e consegnare copie dei contenuti statici (e talvolta dinamici) del vostro sito WordPress, come immagini, CSS, JavaScript e flussi video.**

Prima di tutto, non bisogna confondere un CDN con il vostro host WordPress. Sono servizi completamente separati. Un CDN non sostituisce il vostro provider di hosting, ma è piuttosto un strumento aggiuntivo per aumentare la velocità del vostro sito. Sebbene il nostro hosting Kinsta sia già velocissimo, un CDN può rendere il vostro sito ancora più veloce.

## Come Funziona un CDN

Come funziona esattamente un CDN? Ad esempio, quando il vostro sito web è ospitato su Kinsta, è necessario scegliere una località fisica del data center, ad esempio Stati Uniti, Europa, Asia-Pacifico o Sud America.

Supponiamo che voi scegliate US Central. Questo significa che il vostro sito web si trova fisicamente su un “server host” a Council Bluffs, Iowa. Quando le persone in Europa visitano il vostro sito web, ci vorrà più tempo per caricarlo rispetto a qualcuno che lo visita da Dallas, TX.

Perché? Perché i dati devono percorrere una distanza maggiore. Questo fenomeno è conosciuto come latenza. La latenza si riferisce al tempo e/o al ritardo che si verifica nella trasmissione dei dati attraverso un network. Maggiore la distanza, maggiore la latenza.

## Tipi di CDN

Ci sono due diversi tipi di content delivery network:

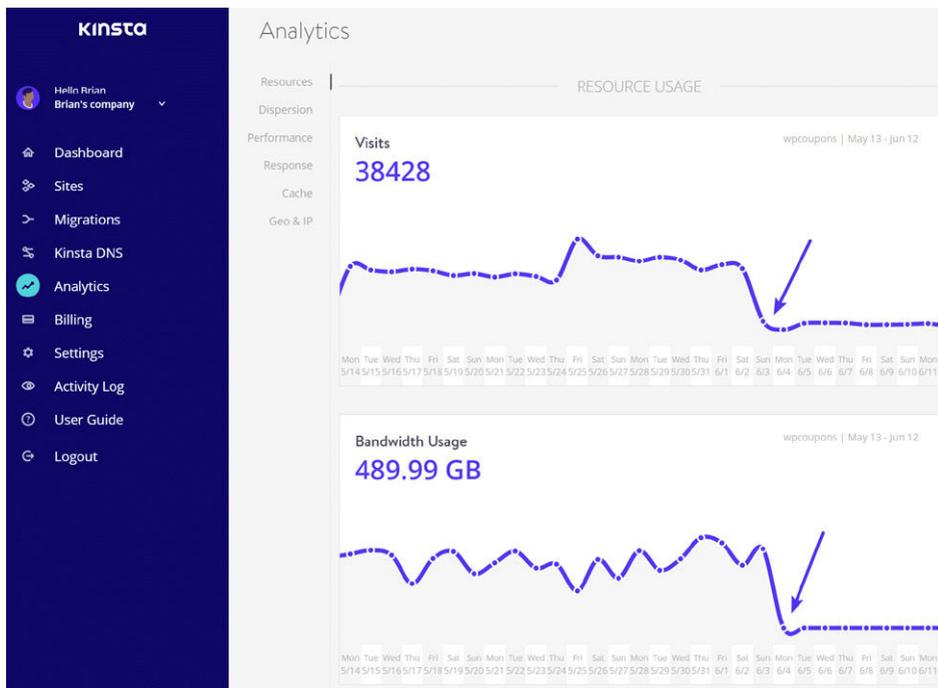
1. CDN Pull Tradizionale
2. CDN a Proxy Inverso

I CDN pull tradizionali memorizzano nella cache una copia di tutti i vostri contenuti e media, ma una richiesta dal client viene comunque inoltrata direttamente al vostro provider di hosting. KeyCDN e CDN77 sono esempi di CDN tradizionale.

Un CDN a proxy inverso è leggermente diverso. Sebbene si comporti come un CDN, intercetta tutte le richieste in arrivo e funge da server intermedio tra il client e il vostro host. Cloudflare e Sucuri sono esempi di CDN a proxy inverso. Questo è uno dei motivi per cui dovete indirizzare il vostro DNS direttamente verso questi provider invece che verso il vostro host.

Il vantaggio di questi ultimi è che agiscono come server intermedio, in grado di fornire forti firewall di applicazioni web che possono contribuire ad evitare che traffico malevolo colpisca il vostro sito WordPress e/o il provider di hosting. Una punto di caduta di questo tipo di CDN è che si presenta con un piccolo extra di costo in termini di prestazioni, rispetto a un tradizionale CDN pull. Ma, grazie alle prestazioni aggiuntive e alle funzionalità legate alla sicurezza, questo costo potrebbe essere considerato trascurabile.

Di seguito è riportato un esempio di quello che è successo dopo aver abilitato Sucuri sul sito di un cliente. Come potete vedere, ha avuto un impatto considerevole sulla quantità di traffico malevolo che stava arrivando al sito. Alla fine, questi tipi di servizi possono aiutarvi a risparmiare sui costi di hosting.



## Speed Test del CDN

In precedenza abbiamo parlato degli enormi vantaggi del caching di WordPress. Bene, anche il caching del CDN è super potente. Questo perché i CDN hanno in genere più server collocati in un maggior numero di località rispetto ai provider di hosting. Ciò significa che possono memorizzare nella cache tutte le vostre risorse (immagini, JS, CSS) in località più vicine ai vostri visitatori e servirli alla velocità della luce.

Facciamo alcuni test rapidi per vedere quanto sia più veloce il vostro sito con un CDN.

## Senza CDN

Il nostro sito Web di prova è ospitato su Kinsta e si trova fisicamente nel data center dell'Iowa, negli Stati Uniti. Abbiamo prima eseguito cinque test di velocità su Pingdom

(CDN non abilitato) e abbiamo fatto la media. Importante: stiamo utilizzando la località Europa – Regno Unito – Londra su Pingdom, per dimostrare la vera potenza di un CDN. Il tempo di caricamento totale è stato **1.03 s**.



Performance grade <b>B 89</b>	Page size <b>231.0 KB</b>
Load time <b>1.03 s</b>	Requests <b>40</b>

## Con CDN

Abbiamo quindi abilitato il nostro CDN e abbiamo eseguito cinque ulteriori test di velocità su Pingdom. Il nostro tempo di caricamento totale è stato di **585 ms** dalla località di prova di Pingdom Europa – Regno Unito – Londra. Quindi, utilizzando il CDN, siamo riusciti a **ridurre i tempi di caricamento della pagina del 43,2%**! È enorme.

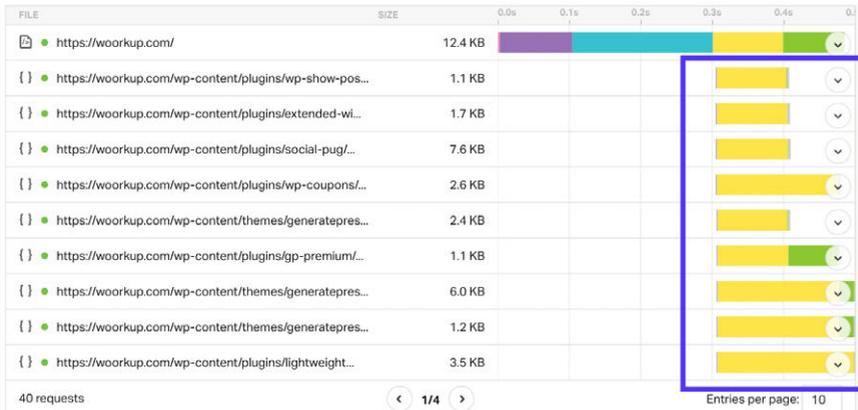


Performance grade <b>B 89</b>	Page size <b>230.6 KB</b>
Load time <b>585 ms</b>	Requests <b>40</b>

La ragione di questa drastica differenza è dovuta al fatto che il CDN ha un centro dati a Londra. Ciò significa che tutte le risorse sono memorizzate nella cache in quella località e sono pronte per essere servite con una latenza minima.

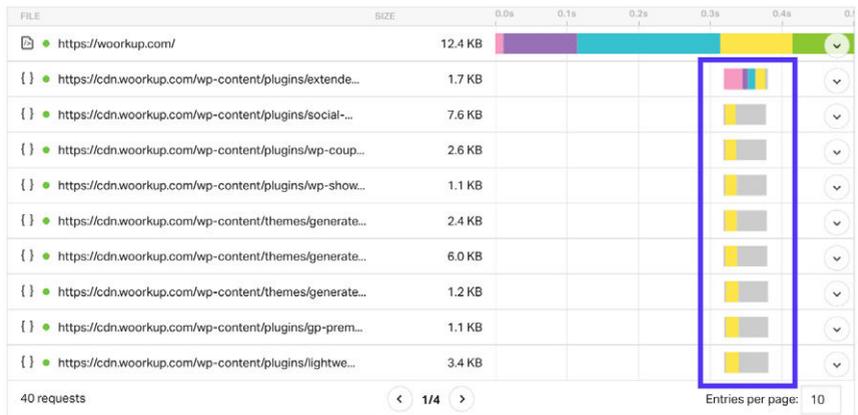
## TTFB Senza CDN

Ricordate che la barra gialla in Pingdom rappresenta il tempo di attesa, che è il tempo del primo byte (TTFB). Nei nostri test di velocità senza il CDN, il TTFB medio sulle risorse era di circa **98 ms**.



## TTFB con CDN

Una volta abilitato il CDN, il TTFB medio sulle risorse è sceso a una media di **15 ms**. Quindi, utilizzando un CDN, il nostro TTFB **medio è sceso dell'84,69%**. Ciò è principalmente dovuto al fatto che le risorse venivano servite direttamente dalla cache del CDN.



*Un CDN ha ridotto i tempi di caricamento della pagina del 43,2%! Scopri perché dovresti usarne uno.*

## Come Abilitare un CDN

Abilitare un CDN sul vostro sito WordPress non deve necessariamente essere difficile, anzi, è abbastanza facile! Seguite questi passaggi.

### Passo 1

Scegliete un provider di CDN e iscrivetevi al loro servizio. Questi servizi sono in genere fatturati su base mensile o in base all'utilizzo dei dati. La maggior parte dei provider avrà un calcolatore per stimare i costi.

- Ogni provider CDN dovrebbe avere della documentazione sul proprio sito per darvi un aiuto iniziale.
- Nel nostro sito [Kinsta.com](https://kinsta.com), abbiamo dei tutorial approfonditi su come installare Cloudflare e Sucuri.

### Passo 2

Se state utilizzando un CDN pull tradizionale, potete utilizzare plugin gratuiti come CDN Enabler (disponibile su [WordPress.org](https://WordPress.org)), WP Rocket (disponibile su [wp-rocket.me](https://wp-rocket.me)), o Perfmatters (disponibile su [perfmatters.io](https://perfmatters.io)) per integrarlo con il vostro sito WordPress. Questi plugin collegano automaticamente le vostre risorse al CDN. Non c'è bisogno di lavoro da parte vostra per mettere i vostri contenuti sul CDN; è tutto pronto! I CDN a Proxy Inverso in genere non richiedono alcun plugin, sebbene a volte vengano forniti per abilitare funzionalità aggiuntive.

- Dashboard
- K Kinsta Cache
- Posts
- Media
- Forms
- Pages
- Downloads
- Jilt
- Knowledge Base
- Appearance
- Plugins

## Perfmatters Settings

Options
CDN
Google Analytics
Extras
License
Support

### CDN

*CDN options that allow you to rewrite your site URLs with your CDN URLs.*

---

**Enable CDN Rewrite** ?

**CDN URL** ?

**Included Directories** ?

**CDN Exclusions** ?

Abilitare il CDN in WordPress con Perfmatters

## Come Abilitare il CDN di Kinsta

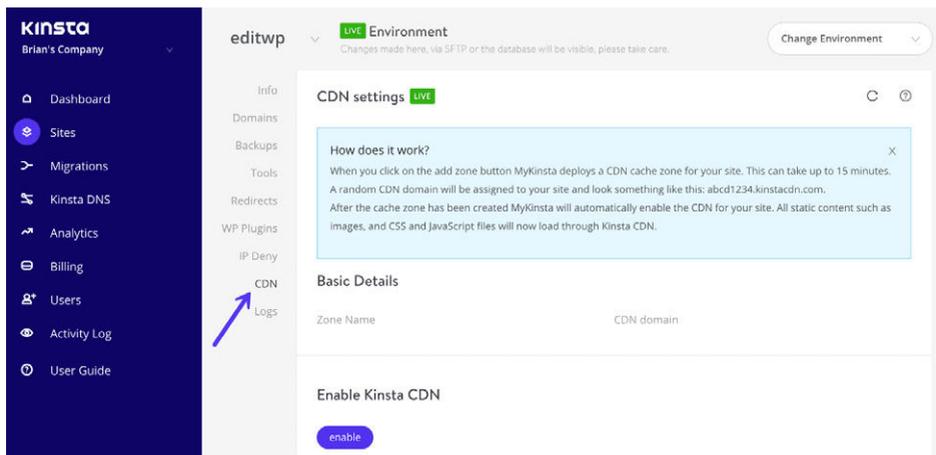
Vi sono piaciuti i test di velocità del CDN qui sopra? In quei test stavamo usando KeyCDN. La buona notizia è che il nostro CDN di Kinsta è alimentato da KeyCDN. È una rete di distribuzione di contenuti abilitata per HTTP/2 e IPv6 con 34 località, in grado di mettere il turbo alle vostre risorse e media in tutto il mondo. Le regioni attualmente servite sono America, Sud America, Europa, Africa, Asia e Australia.



Per i clienti di Kinsta, offriamo larghezza di banda CDN gratuita su tutti i nostri piani di hosting. Potete abilitare il CDN di Kinsta in due semplici passaggi.

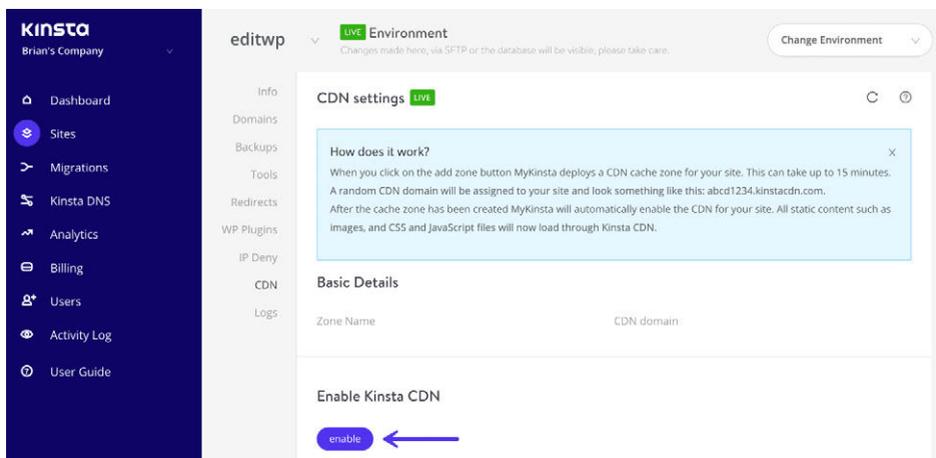
# Passo 1

Innanzitutto, accedete al cruscotto di MyKinsta. Cliccate sul vostro sito e poi sulla scheda Kinsta CDN.



# Passo 2

Quindi fate clic su “Abilita il CDN di Kinsta”. Dopo alcuni minuti, il CDN viene distribuito automaticamente e le risorse verranno pubblicate dalla cache in tutto il mondo. Questo è tutto ciò che c'è da fare.



## Ottimizzazioni Aggiuntive del CDN

Ecco alcune ottimizzazioni aggiuntive del CDN che dovrete provare o su cui dovrete riflettere.

- Se avete molti commenti, sappiate che i gravatar possono generare molte richieste, caricando da [secure.gravatar.com](https://secure.gravatar.com). Potete, invece, caricare i gravatar dal vostro CDN. Lo facciamo anche noi sul sito web di Kinsta. Se siete clienti di Kinsta, vi basta contattare il nostro team di supporto per avere la configurazione necessaria.
- Potete ospitare i vostri web font personali, i font di Google, ed anche la vostra favicon sul vostro CDN. I link ai tutorial su come fare sono disponibili nel nostro sito web all'indirizzo <https://kinsta.com/speed-up-wordpress>.

CAPITOLO 12:

# Scaricare Media e Email Quando È Necessario



**Tutto ciò che genera una richiesta ha un impatto sulle prestazioni del vostro sito in un modo o nell'altro. Per i siti che ospitano centinaia di migliaia di file o media di grandi dimensioni, potrebbe essere saggio scaricarli del tutto. L'offloading è una cosa diversa dal servirli tramite un CDN. Con un CDN i dati originali risiedono ancora sul vostro host, il CDN ne ha semplicemente più copie.**

Quando la cache delle risorse scade, il CDN richiede al vostro host le ultime copie dei file. I CDN hanno lo scopo di memorizzare i file nella cache per lunghi periodi di tempo. Ma, dato che hanno così tanti POP, potrebbero essere generate molte nuove query man mano che la cache scade nelle diverse regioni.

Quando si scaricano file o contenuti multimediali, vuol dire che si sta effettivamente spostando la loro posizione fisica originale dal vostro provider di hosting. Quindi, sebbene possa sembrare che i file siano serviti dal vostro sito, si trovano in realtà completamente da un'altra parte. Oltre a ridurre le query verso l'host, lo scopo principale dell'offloading è ovviamente quello di risparmiare spazio su disco.

## Scaricare i Media su Amazon S3

Una delle soluzioni di offload più popolari è Amazon S3. Si tratta di un servizio di archiviazione e fa parte dei molti prodotti di Amazon Web Services. Generalmente viene utilizzato

per siti di grandi dimensioni che richiedono backup aggiuntivi o servono file di grandi dimensioni (download, software, video, giochi, file audio, PDF, ecc.). Amazon ha una comprovata esperienza di affidabilità e, grazie alla sua enorme infrastruttura, può proporre costi di archiviazione molto bassi. Tra i clienti di S3 ci sono Netflix, Airbnb, SmugMug, Nasdaq, ecc.



Dato che si occupano interamente dell'archiviazione di massa, potete quasi star certi che i prezzi saranno più economici rispetto al vostro host WordPress. Scaricare i media su AWS può essere un ottimo modo per risparmiare denaro ed è gratuito per il primo anno (fino a 5 GB di spazio di archiviazione). Inoltre, dato che le richieste dei vostri file multimediali vengono servite direttamente da Amazon, si riduce il carico sul vostro sito WordPress, il che significa tempi di caricamento più rapidi.

Per questo consigliamo di utilizzare il plugin WP Offload Media Lite for Amazon S3 (disponibile su WordPress.org). Il plugin crea automaticamente una copia dei file del vostro sito WordPress su Amazon S3 quando vengono caricati nella libreria media uploaded to the media library. Potete anche utilizzare un CDN con questi media per ottenere il massimo da entrambi.

## Scaricare i Media su Google Cloud Storage



Un'altra soluzione di offloading popolare è Google Cloud Storage. Dato che Kinsta è basato su Google Cloud Platform, siamo grandi fan della loro tecnologia e della loro infrastruttura. Grazie alla massiccia infrastruttura di Google e al fatto che si occupano di archiviazione di massa, possono offrire costi di archiviazione molto bassi. Tra i loro clienti ci sono Spotify, Vimeo, Coca-Cola, Philips, Evernote e Motorola.

Consigliamo di usare il plugin gratuito WP-Stateless (disponibile su WordPress.org). Il plugin copia automaticamente i file dal vostro sito WordPress su Google Cloud Storage man mano che questi vengono caricati nella libreria multimediale. Sono disponibili diverse modalità, ad esempio il backup di una copia dei file multimediali o l'eliminazione dei file da WordPress e la loro distribuzione da Google Cloud Storage.

## Scaricare le Email Transazionali e di Marketing

Che vi piaccia o no, le email hanno un notevole impatto sul vostro server e sulle sue risorse. Su alcuni host, in particolare gli host condivisi, fare un uso eccessivo di email potrebbe addirittura farvi sospendere. Questo in particolare diventa un problema per coloro che cercano di inviare email in massa, ed è il motivo per cui esistono provider di email transazionali di terze parti, ed è anche il motivo per cui molti provider di hosting bloccano l'invio di email sulle porte standard. **Noi consigliamo di non utilizzare mai il vostro provider di hosting per le email.**

Se inviate newsletter o email in massa, per ottenere i risultati migliori vi consigliamo sempre le seguenti soluzioni:

- Utilizzate un software di email marketing professionale di terze parti che non fa parte di WordPress
- Utilizzate un provider di servizi di posta elettronica transazionale (HTTP API o SMTP) insieme a WordPress

Tra gli altri vantaggi dei servizi di terze parti ricordiamo:

- Migliore consegna della posta elettronica. Lasciate fare ai provider di posta elettronica ciò che sanno fare meglio!
- Meno possibilità di finire nelle blacklist.
- Potrebbe non essere sempre possibile impostare i record DMARC con il vostro provider di hosting.

## Strumenti di Email Marketing

Some examples of marketing emails include newsletters, product and feature announcements, sales, event invitations, onboarding reminders, etc. Here are a few email marketing tools we recommend:

- **MailChimp** – Da Kinsta usiamo Mailchimp.
- **MailerLite**
- **Drip**

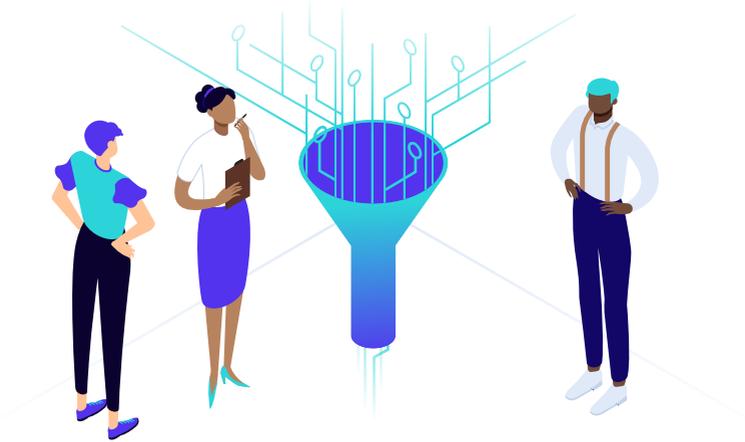
## Servizi di Email Transazionale

Esempi di email transazionali sono le ricevute di acquisto da WooCommerce o EDD, le notifiche di creazione degli account, le notifiche di spedizione, i messaggi di errore delle app, i reset delle password, ecc. Se siete clienti di Kinsta, sappiate che noi ci affidiamo ad un provider SMTP di terze parti per garantire un'elevato tasso di consegna. Ma, con l'aumentare dei volumi, consigliamo sempre di sganciare il servizio dal sito. Ecco alcuni servizi di posta elettronica transazionale:

- **SendGrid** – Da Kinsta usiamo SendGrid.
- **Mailgun**
- **SparkPost**

CAPITOLO 13:

# Come Trovare Strozzature e Plugin Lenti



**Ora vi forniremo alcuni suggerimenti per individuare le strozzature sul vostro sito WordPress e vi diremo cosa fare al riguardo.**

## **Utilizzare New Relic per Identificare Plugin e Query sul Database Lenti**

Sul mercato ci sono diversi ottimi strumenti che possono aiutarvi ad individuare plugin lenti e query sul database la cui esecuzione richiede molto tempo. Da Kinsta siamo grandi fan di New Relic e lo utilizziamo quotidianamente. Si tratta di uno strumento di monitoraggio PHP che fornisce statistiche dettagliate sul rendimento del vostro sito web.

Se siete clienti di Kinsta, potete persino aggiungere la vostra chiave di licenza New Relic nel cruscotto MyKinsta.

The screenshot shows the Kinsta control panel for 'Brian's Company'. The left sidebar contains navigation options like Dashboard, Sites, Migrations, Kinsta DNS, Analytics, Billing, Users, Activity Log, and User Guide. The main content area is titled 'editwp Environment' and contains several tool cards:

- Clear Site Cache:** A card with a blue lightning bolt icon and a 'Clear Cache' button.
- SSL Certificate:** A card with a green padlock icon, a 'Let's Encrypt' button, and a 'Modify' dropdown.
- PHP Engine:** A card with a blue document icon and a 'PHP 7.2' button with a 'Modify' dropdown.
- Restart PHP:** A card with a blue circular arrow icon and a 'Restart PHP' button.
- New Relic Monitoring:** A card with a blue monitor icon, a 'Monitoring Disabled' status, and an 'Enable Monitoring' button. A blue arrow points to this button.
- Search And Replace:** A card with a blue magnifying glass icon and a 'Search And Replace' button.

Tuttavia, utilizzate New Relic con attenzione poiché influisce sulle prestazioni del sito, in quanto aggiunge componenti JavaScript al vostro sito web. Vi consigliamo di attivarlo quando dovete risolvere problemi di prestazioni e poi disattivarlo.

## Individuare i Plugin Lenti

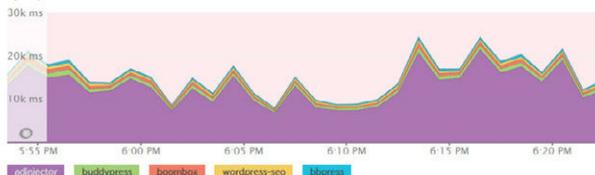
Quando un plugin di WordPress causa un rallentamento generale, i sintomi variano a seconda dell'attività svolta dal plugin. Tuttavia, in molti casi scoprirete che un plugin lento avrà effetto su ogni pagina di un sito WordPress. Nel caso del sito riportato nell'immagine sottostante, è stata osservata una lentezza generale su ogni pagina di front-end. Ecco cosa ci dice New Relic sulle prestazioni dei plugin presenti nel sito.

TIME PICKER  
Last 30 minutes ending now

Sort by Most time consuming

adinjector	390 sec
buddypress	24.4 sec
boombox	17.6 sec
wordpress-seo	15.6 sec
bbpress	12.6 sec
easy-social-share-buttons3	4.34 sec
zombily	4.22 sec

Top 5 plugins and themes  
by response time



Potete subito vedere che il plugin adinjector impiega un tempo superiore di oltre 15 volte rispetto al plugin successivo.

Quando vedete dati come questi, si può essere tentati di eliminare immediatamente il plugin in quanto mal progettato o in qualche modo inefficace. Sebbene a volte sia così, non è sempre il caso. L'errata configurazione del plugin, la lentezza del database o le risorse esterne lente a rispondere possono far sì che il plugin impieghi tempi prolungati.

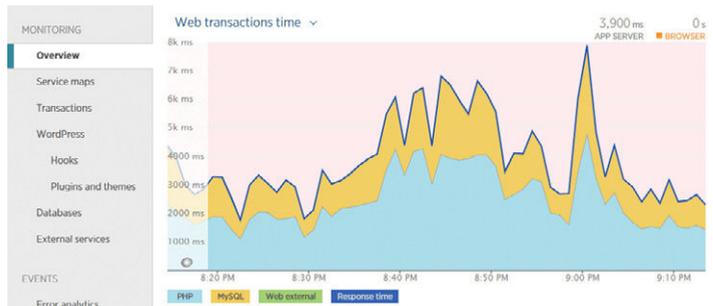
Quindi, quando vedete un plugin che risponde lentamente, è una buona idea controllare altre schermate di New Relic per trovare altre informazioni. Le transazioni, i database e le risorse esterne devono essere controllati prima di decidere che la disattivazione del plugin sia la migliore o unica via da seguire.

## Lentezza Generale Causata da un Database Sovraccarico

Un database non ottimizzato può causare una lentezza generale in un sito WordPress. In precedenza abbiamo esaminato molte delle cose da fare per risolvere questo problema. In New Relic, la lentezza del database si presenterà molto probabilmente in due modi:

- In primo luogo, vedrete nella panoramica un'enorme quantità di attività MySQL.
- In secondo luogo, nella scheda Database vedrete una o più tabelle impiegare molto tempo.

Nella schermata panoramica, un sito con un database in difficoltà potrebbe essere simile a questo:



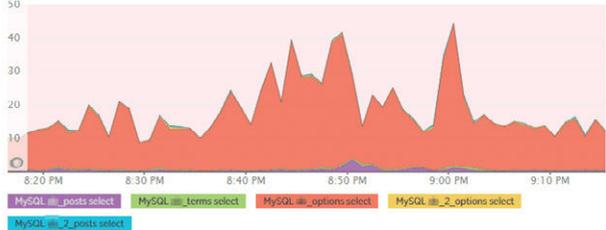
Per una migliore prospettiva delle tabelle o delle query del database che causano problemi, date un'occhiata alla scheda Database.

SORT BY  
Most time consuming

MySQL options select	17.5 s
MySQL posts select	0.461 s
MySQL _2_options select	0.289 s
MySQL terms select	0.257 s
MySQL _2_posts select	0.11 s
MySQL _postmeta select	0.078 s
MySQL _options update	0.0467 s
MySQL _termmeta select	0.0348 s

### MySQL overview

Top database operations by time consumed



La scheda Database evidenzierà la tabella e il tipo di query che impiega più tempo. Se selezionate una delle voci nell'elenco, potrete visualizzarne i dettagli, incluse alcune query di esempio.

SORT BY  
Most time consuming

MySQL options select	17.5 s
MySQL posts select	0.461 s
MySQL _2_options select	0.289 s
MySQL terms select	0.257 s
MySQL _2_posts select	0.11 s
MySQL _postmeta select	0.078 s
MySQL _options update	0.0467 s
MySQL termmeta select	0.0348 s
MySQL _2_options update	0.0309 s
MySQL other	0.0281 s
MySQL _sitemeta select	0.0115 s

### MySQL options select

Slow query trace 2,510 ms /archive-product  
9:14 PM MAX TIME ACTION

**Query**  
`SELECT option_name, option_value FROM wp_options WHERE autoload = ?`

**Database instance**  
[redacted]

**Database name**  
[redacted]

**Stack trace**

```
- mysql_query called at /www/html/wp-content/themes/2013/public/wp-includes/wp-db.php (1887)
-db: do_query called at /www/html/wp-content/themes/2013/public/wp-includes/wp-db.php (1775)
in wpdb::query called at /www/html/wp-content/themes/2013/public/wp-includes/wp-db.php (2398)
-iget_results called at /www/html/wp-content/themes/2013/public/wp-includes/option.php (189)
-!options called at /www/html/wp-content/themes/2013/public/wp-includes/functions.php (1364)
-!blog_installed called at /www/html/wp-content/themes/2013/public/wp-includes/load.php (832)
in wp_not_installed called at /www/html/wp-content/themes/2013/public/wp-settings.php (140)
in require_once called at /www/html/wp-content/themes/2013/public/wp-config.php (65)
in require_once called at /www/html/wp-content/themes/2013/public/wp-load.php (37)
in require_once called at /www/html/wp-content/themes/2013/public/wp-blog-header.php (13)
in require called at /www/html/wp-content/themes/2013/public/index.php (17)
```

In questo caso, i dati a disposizione puntano il dito sui dati a caricamento automatico della tabella `wp_options`. Ricordate? Abbiamo analizzato questo punto in precedenza. Una rapida analisi della tabella `wp_options` conferma che quasi 250 MB di dati vengono caricati automaticamente, rendendo questo sito un ovvio candidato alla manutenzione e all'ottimizzazione del database.

Non perdetevi il nostro tutorial approfondito su come utilizzare New Relic per risolvere i problemi di prestazioni sul vostro sito WordPress.

## Il Plugin Gratuito Query Monitor

Potete anche utilizzare il plugin gratuito di WordPress Query Monitor. Utilizzatelo per identificare ed eseguire il debug di query lente, chiamate AJAX, richieste API REST e molto altro. Inoltre, il plugin riporta i dettagli del sito, come dipendenze e dipendenti degli script, hook di WordPress attivati durante la generazione della pagina, dati dell'ambiente di hosting, tag di query condizionali incontrati dalla pagina corrente e molto altro.

Query Monitor				
Page generation time	Peak memory usage	Database query time	Database queries	Object cache
0.0321 0.0% of 0s limit	2,048 kB 0.0% of 16,777,216 kB limit	0.0022	SELECT: 3	99.9% hit rate External object cache: true

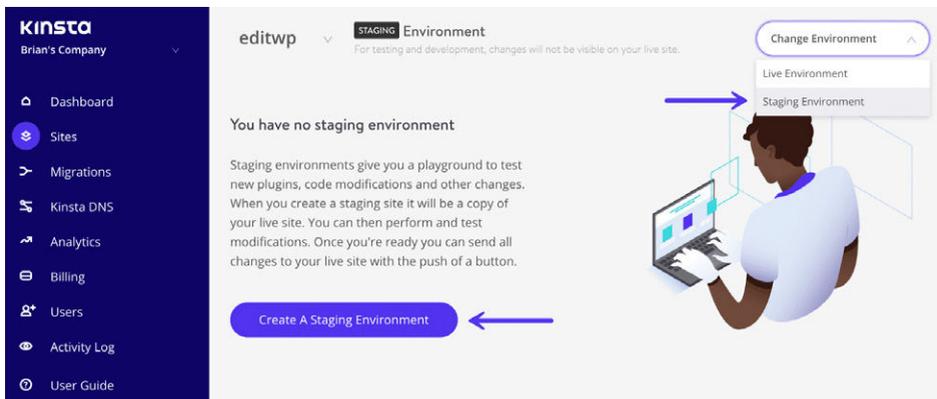
  

\$wpdb Queries				
Query	Caller	Component	Rows	Time
1 SELECT user_id, meta_key, meta_value FROM wp_usermeta WHERE user_id IN (1) ORDER BY umeta_id ASC	update_meta_cache() wp-includes/meta.php:826	+ Plugin: coming-soon-builder	39	0.0004
2 SELECT wp_posts.* FROM wp_posts WHERE 1=1 AND (wp_posts.ID = '2') AND wp_posts.post_type = 'page' ORDER BY wp_posts.post_date DESC	WP_Query->get_posts() wp-includes/query.php:3655	+ Core	1	0.0017
3 SELECT * FROM wp_posts WHERE (post_type = 'page' AND post_status = 'publish') AND post_parent = 2 ORDER BY wp_posts.post_title ASC LIMIT 0,1	get_pages() wp-includes/post.php:4598	+ Core	0	0.0001
Total Queries: 3				0.0022

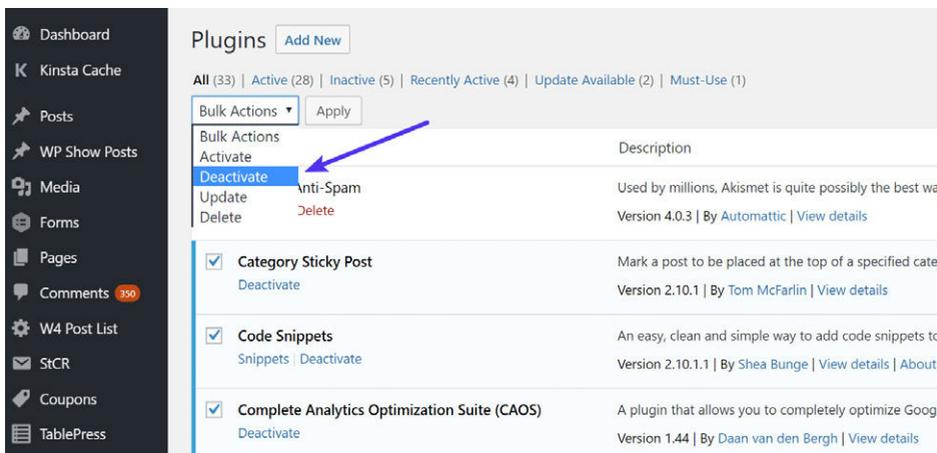
Il plugin è stato sviluppato da John Blackbourn, un committer del core di WordPress, al momento sviluppatore presso Human Made e precedentemente impiegato da WordPress.com VIP – in altre parole, qualcuno che conosce bene WordPress. Query Monitor è stato aggiunto alla directory dei plugin di WordPress nel 2013 e attualmente vanta oltre 10.000 installazioni attive, un numero impressionante per un plugin di sviluppo. La valutazione degli utenti, cinque stelle su cinque, spiega la sua popolarità tra gli sviluppatori.

# Utilizzare Siti di Staging Senza Toccare la Produzione

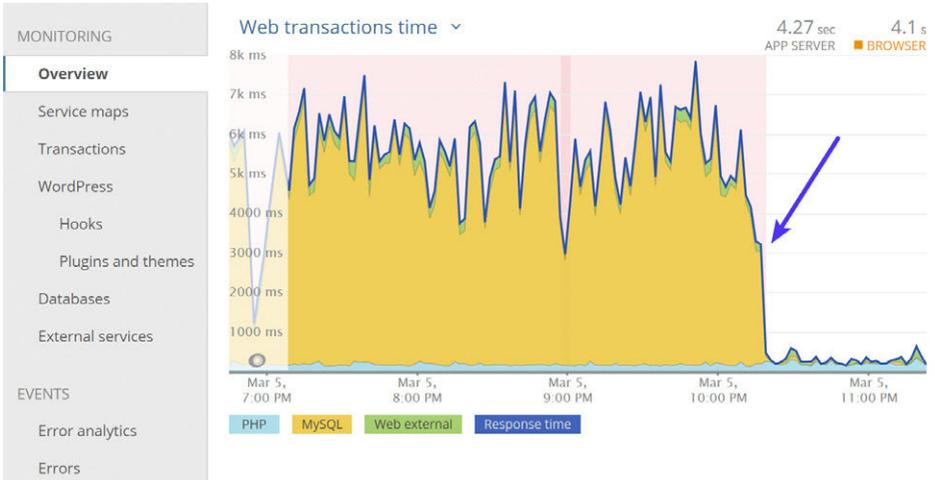
Non sappiamo cosa faremmo senza ambienti di staging. Possono essere inestimabili quando bisogna risolvere problemi di prestazioni. Fortunatamente, Kinsta dispone di ambienti di staging ad un clic. Se il vostro host WordPress non offre ambienti di staging, potreste anche usare un plugin come WP Staging, anche se non è così semplice da usare.



Dopo aver installato e avviato un sito di staging, la prima cosa che potete fare è disabilitare tutti i vostri plugin. Dato che questa è una copia del vostro sito live, non dovete preoccuparvi di rompere nulla. È di gran lunga uno dei modi più semplici per restringere i problemi. Basta andare su Plugin, selezionarli tutti e scegliere “Disattiva” dalle opzioni di gruppo.



Sarà, quindi, possibile monitorare i tempi di risposta in New Relic o Query Monitor e vedere cosa succede. Nell'esempio che segue, i tempi di risposta sono immediatamente tornati alla normalità, quindi era chiaro che uno dei plugin stava generando un'anomalia. Potete, quindi, riattivarli uno a uno, ripetendo la stessa procedura fino a trovare il colpevole.



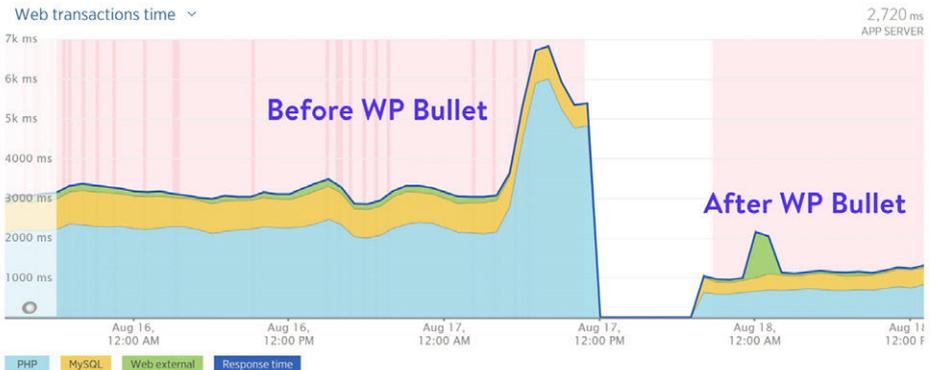
Ecco un esempio di cosa è successo quando abbiamo abilitato il plugin che stava causando il problema. I tempi di caricamento (tempi di transazione web) sono immediatamente ritornati ai livelli precedenti.



Cosa dovrete fare dopo aver trovato il plugin che causa il rallentamento? Ecco cosa consigliamo:

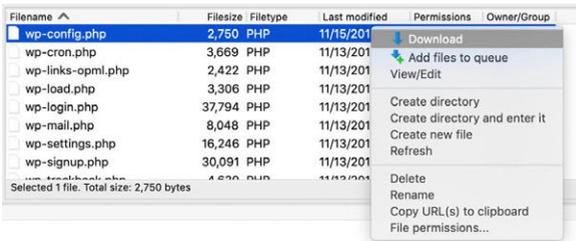
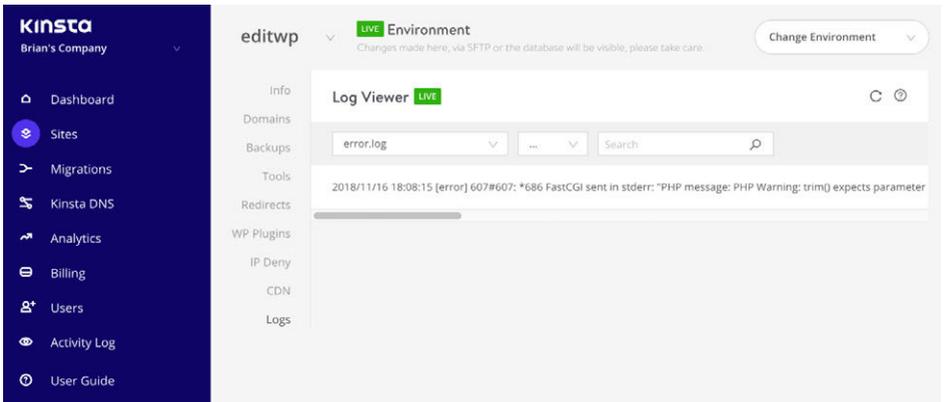
1. Aggiornate i vostri temi e plugin all'ultima versione, se non lo avete già fatto.
2. Contattate lo sviluppatore del plugin o del tema e chiedete assistenza.
3. Cercate un plugin alternativo che fornisca la stessa funzionalità.
4. Nel caso in cui il problema fosse generato dalla versione di PHP, cambiate il motore PHP ad una versione inferiore per verificare se il tema o il plugin riprende a funzionare.

Potreste anche assumere uno sviluppatore WordPress per risolvere il problema. Se questo è legato alle prestazioni, dovremmo dare una voce a Mike Andreason di WP Bullet. Mike è uno sviluppatore full-time di Codeable, specializzato nell'ottimizzazione delle prestazioni, che, qui da Kinsta, ha aiutato molti clienti con installazioni complesse a portare il loro sito ad un livello superiore.



## Controllate i Log degli Errori

Controllare i log degli errori non è mai divertente, ma può dire molto sui problemi di prestazioni generati dai plugin di WordPress. Se siete clienti di Kinsta, potete facilmente visualizzare registro degli errori, registro della cache e registro degli accessi direttamente dal cruscotto di MyKinsta.



Potete anche abilitare i log degli errori aggiungendo del codice al vostro file `wp-config.php`. Innanzitutto, vi consigliamo di connettervi al vostro sito tramite SFTP. Quindi scaricate il vostro file `wp-config.php` in modo da poterlo modificare.

Nota: fate sempre un backup di questo file prima di cominciare!

Cercate la riga che dice `/* Finito, interrompere le modifiche! Buon blogging. */ e, subito prima, aggiungete quanto segue (come si vede più sotto):`

```
define( 'WP_DEBUG', true );
```

```

37
38
39
40 define( 'WP_DEBUG', true );
41
42 /* That's all, stop editing! Happy blogging. */
43
44 /** Absolute path to the WordPress directory. */
45 if ( !defined( 'ABSPATH' ) )
46     define( 'ABSPATH', dirname( __FILE__ ) . '/' );
47
48 /** Sets up WordPress vars and included files. */
49 require_once( ABSPATH . 'wp-settings.php' );

```

Se il codice qui sopra è già nel vostro file `wp-config.php` ma è impostato su “false”, basta semplicemente cambiarlo in “true”. Questo abiliterà la modalità di debug. Nota: visualizzerete anche avvisi ed errori nel pannello di amministrazione di WordPress.

Potete anche abilitare il registro di debug in modo che invii tutti gli errori ad un file aggiungendo il seguente codice subito dopo la riga `WP_DEBUG` (come illustrato di seguito):

```
define( 'WP_DEBUG_LOG', true );
```

```
38
39
40 define( 'WP_DEBUG', true );
41 define( 'WP_DEBUG_LOG', true );
42
43 /* That's all, stop editing! Happy blogging. */
44
45 /** Absolute path to the WordPress directory. */
46 if ( !defined('ABSPATH') )
47     define('ABSPATH', dirname(__FILE__) . '/');
48
49 /** Sets up WordPress vars and included files. */
50 require_once(ABSPATH . 'wp-settings.php');
```

Salvate le modifiche e caricate nuovamente il file sul server. Gli errori saranno così registrati nel file `debug.log` all'interno della cartella `/wp-content/`. Se, per qualche motivo, non vedete questo file, potete sempre crearne uno.

## Utilizzate MyKinsta Analytics

Se siete clienti di Kinsta, potete sfruttare le informazioni sulle performance che abbiamo incorporato nel nostro tool MyKinsta Analytics.

Nella sezione Monitoraggio delle prestazioni, è possibile visualizzare il tempo medio di risposta PHP + MySQL, il throughput PHP, l'utilizzo di AJAX, i principali tempi medi e i principali tempi massimi di upstream.





# Principali Tempi Medi di Risposta PHP + MySQL

Questo elenco mostra i tempi medi di risposta più alti generati da PHP e MySQL. Questi numeri possono essere picchi unici, quindi è consigliabile confrontare questa lista con i “Principali tempi massimi di Upstream”.

PATH	REQUESTS	TIME
/wp-admin/update.php?action=upload-plugin	1	2.19 s
/wp-cron.php?server_triggered_cronjob	93	1.97 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca	1	1.47 s
/wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s=	2	1.37 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974	1	1.07 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca	1	0.91 s
/wp-cron.php?doing_wp_cron=1542348999.8110280036926269531250	1	0.67 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=113bcb7711	1	0.53 s
/wp-admin/plugins.php	3	0.49 s
/wp-admin/plugin-install.php	1	0.48 s

## Principali Tempi Massimi di Upstream

Il tempo di upstream è il tempo totale impiegato da NGINX (e dai server upstream) per elaborare una richiesta e inviare una risposta. Il tempo viene misurato in secondi, con una precisione al millisecondo.

PATH	REQUESTS	TIME
/wp-cron.php?server_triggered_cronjob	93	180.00 s
/wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s=	2	2.60 s
/wp-admin/update.php?action=upload-plugin	1	2.19 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca	1	1.47 s
/hello-world/	5	1.46 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974	1	1.07 s
/	83	0.93 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca	1	0.91 s
/wp-admin/plugins.php	3	0.85 s
/wp-cron.php?doing_wp_cron=1542348999.8110280036926269531250	1	0.67 s

## Il Vostro Sito Potrebbe Essere Stato Violato

Se avete difficoltà ad individuare la causa di un problema di prestazioni, è molto probabile che il vostro sito sia stato violato, infettato da malware o sottoposto a un attacco DDoS. Ciò può influire sulla velocità del vostro sito e anche sulla reattività della dashboard di WordPress. In questi casi, consigliamo quanto segue:

1. Implementate un server proxy e WAF come Cloudflare o Sucuri.
2. Bloccate gli indirizzi IP malevoli utilizzando i servizi di cui sopra. Se siete clienti di Kinsta, potete anche bloccare gli indirizzi IP dal nostro cruscotto MyKinsta.
3. Potete anche implementare il blocco geografico. Alcuni paesi generano una gran quantità di traffico malevolo. Se siete sotto attacco, potreste dover bloccare l'intero paese, temporaneamente o definitivamente.

## Risoluzione dei Problemi con i Codici di Errore (Codici di Stato HTTP)

I codici di stato HTTP sono come una breve nota del server che viene aggiunta nella parte superiore di una pagina web. Non fa parte della pagina web, ma è un messaggio emesso dal server che vi fa di sapere cosa è successo quando la richiesta di visualizzare la pagina è stata ricevuta. Questi codici possono essere inestimabili nella risoluzione dei problemi!

Sebbene ci siano oltre 40 diversi codici di stato, qui di seguito riportiamo quelli con cui gli utenti di WordPress sono alle prese più di frequente. Abbiamo diversi tutorial sui codici di errore su kinsta.com, con le istruzioni necessarie per risolvere ognuno di questi errori:

**429: “Troppe richieste.”** Generato dal server quando l'utente ha inviato troppe richieste in un dato intervallo di tempo (limitazione di quota). Questo a volte può essere generato da bot o script che tentano di accedere al vostro sito. In questo caso, potreste provare a cambiare il vostro URL di accesso a WordPress.

### 429 Too Many Requests

---

nginx

**500: “Si è verificato un errore sul server e non è stato possibile completare la richiesta”.** Un codice generico che significa semplicemente “errore interno del server”. Qualcosa è andato storto sul server e la risorsa richiesta non è stata consegnata. Questo codice è in genere generato da plugin di terze parti, codice PHP difettoso o anche dall'interruzione della connessione al database.

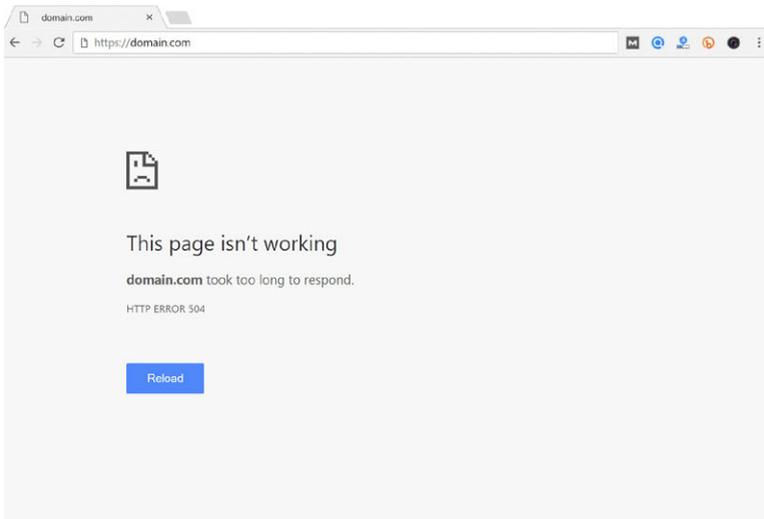
### Error establishing a database connection

**502: “Bad Gateway.”** Questo codice di errore indica in genere che un server ha ricevuto una risposta non valida da un altro. A volte una query o una richiesta impiega troppo tempo e quindi viene cancellata o uccisa dal server e la connessione al database viene interrotta.



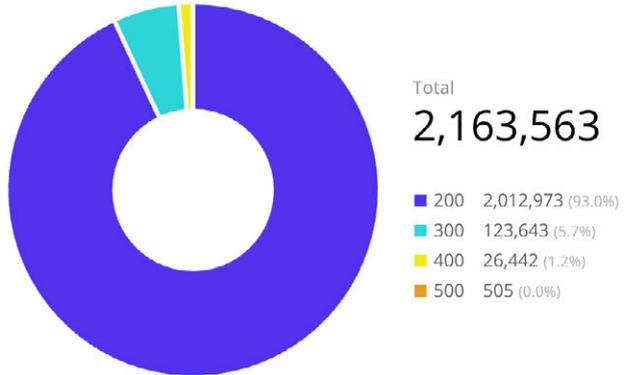
**503: “Il server non è disponibile per gestire questa richiesta in questo momento”.** La richiesta non può essere completata al momento. Questo codice può essere restituito da un server sovraccarico che non è in grado di gestire ulteriori richieste.

**504: “Il server, che agisce da gateway, è scaduto in attesa della risposta di un altro server”.** Questo codice è restituito quando ci sono due server coinvolti nell’elaborazione di una richiesta e il primo server va in timeout in attesa della risposta del secondo server.



Potete anche approfondire questi codici di risposta HTTP nel nostro MyKinsta Analytics. Il nostro report di suddivisione dei codici di risposta consente di visualizzare una panoramica della distribuzione dei codici di stato HTTP serviti per le risorse richieste.

## Suddivisione Codici Risposta



Il report sulle statistiche delle risposte consente di visualizzare il numero totale di redirect che si verificano, errori, percentuali di successo e tasso di errore. In genere ogni sito WordPress presenta un piccolo tasso di errore; ma questo è del tutto normale.

## Statistiche Risposte

**513,037**  
Redirects

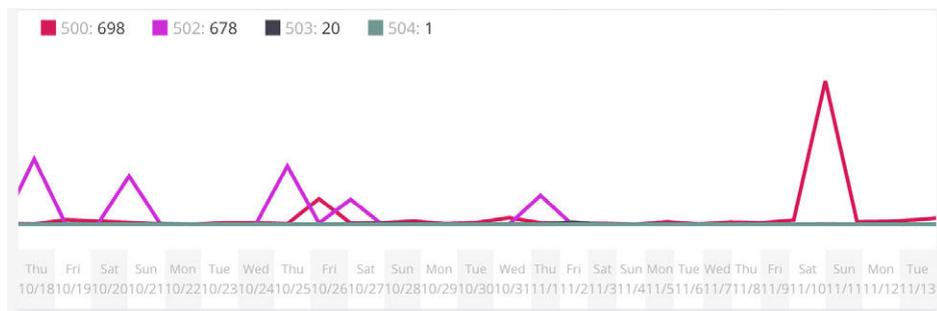
**119,796**  
Errors

**99.3%**  
Success Rate

**0.7%**  
Error Ratio

Esistono quindi report di suddivisione per ciascun tipo di codice di errore, ad esempio errori 500, errori 400, redirect, ecc.

## Suddivisione Errori 500



CAPITOLO 14:

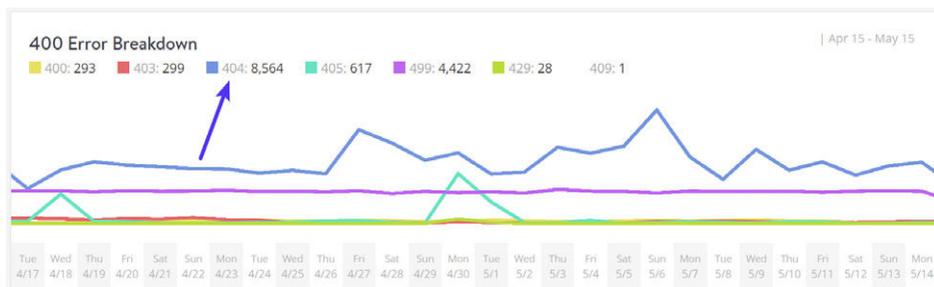
# Raccomandazioni sull'Ottimizzazione del Back-End



Ora ci addentreremo nei metodi per accelerare WordPress ottimizzando il back-end. Il back-end in genere riguarda tutto ciò che viene gestito interamente dal server, come PHP, header HTTP della cache, compressione GZIP, ecc.

## Create una Pagina 404 Leggera

Abbiamo visto in prima persona che siti altamente dinamici generano in genere molti errori 404. Il vostro sito web potrebbe generarne più di quanto pensiate! Il nostro tool di analisi di MyKinsta può aiutarvi a determinarne il numero esatto (come mostrato di seguito).



La ragione per cui questi errori sono dannosi è che un gran numero di pagine 404 richiede molte risorse. Per un sito WordPress altamente dinamico, vi consigliamo di evitare una pagina 404 pesante. Create un semplice template 404 che eviti di interrogare ulteriormente il database, se possibile. Ovviamente, dedicate un po' di tempo a correggere gli errori 404 poiché non si tratta solo di un uso intensivo delle risorse, ma è anche negativo per la user experience.

## Aumentate i PHP Worker

Forse non avete mai sentito parlare di PHP worker potrebbe. Questi costituiscono il modo in cui molti host, compreso Kinsta, gestiscono la limitazione delle richieste (invece che stabilire limiti di CPU o di RAM, che è in genere la soluzione adottata dai provider di hosting condiviso).

I PHP worker determinano **il numero di richieste simultanee che il vostro sito può gestire in un dato momento**. Per dirla in modo semplice, ogni richiesta non presente nella cache per il vostro sito web è gestita da un PHP worker. Ad esempio, se ci sono 4 richieste che arrivano al vostro sito nello stesso istante e il vostro sito ha 2 PHP worker, due di queste richieste verranno elaborate mentre le altre due dovranno attendere in coda fino a quando le prime due non avranno concluso l'elaborazione.

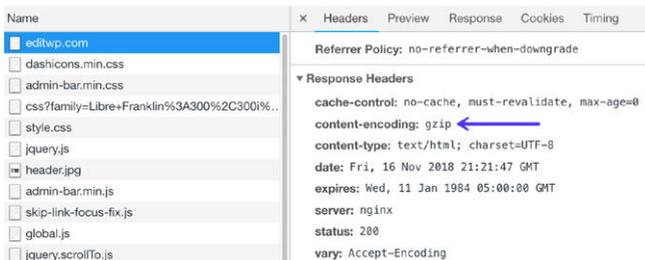
Ricordate che abbiamo detto in precedenza che uno dei maggiori problemi dei siti WordPress ad iscrizione sono tutte quelle richieste non presenti nella cache. Questo è il motivo per cui i PHP worker diventano molto importanti: devono fare del lavoro per ogni richiesta. Pertanto, questi siti in genere richiedono un maggior numero di PHP worker per fare in modo che ogni richiesta venga elaborata senza ritardi e sia completata correttamente.

Cosa succede se andate continuamente oltre il numero di PHP worker disponibili? In sostanza, la coda inizia ad annullare le richieste più vecchie, le quali potrebbero causare errori 500 sul sito. Ciascuno dei piani di hosting di Kinsta dispone di un numero predefinito di PHP worker. Se avete difficoltà nel fare una stima del fabbisogno di PHP worker del vostro sito, potete sempre chattare con il nostro team di vendita o di supporto.

## Utilizzate la Compressione GZIP

GZIP è un formato di file e un'applicazione software utilizzata per la compressione e la decompressione dei file. La compressione GZIP è abilitata sul lato server e consente una maggiore riduzione delle dimensioni di file HTML, fogli di stile e file JavaScript.

Quando un browser visita un sito web, verifica se GZIP è attivo sul server, controllando se esiste l'header HTTP `content-encoding: gzip`. Se viene rilevata questa intestazione, allora serve i file compressi e in dimensioni minori. In caso contrario, serve i file non compressi. Se non avete abilitato GZIP, molto probabilmente riceverete avvisi ed errori negli strumenti di test della velocità come Google PageSpeed Insights e GTmetrix.



L'abilitazione della compressione GZIP può aiutarvi a ridurre le dimensioni della vostra pagina web, cosa che riduce in modo significativo il tempo necessario a scaricare la risorsa, riduce l'utilizzo dei dati per il client e il tempo necessario al rendering delle pagine. La compressione GZIP è normalmente standard nella maggior parte dei provider di hosting, anche se in realtà non ci sorprendiamo più di nulla.

## Abilitate la Protezione dagli Hotlink

Il concetto di hotlinking è piuttosto semplice. Trovate un'immagine da qualche parte su internet e utilizzate l'URL dell'immagine direttamente sul vostro sito. Questa immagine sarà visualizzata sul vostro sito web ma sarà servita dalla posizione originale. È molto conveniente per chi usa l'hotlink, ma in realtà è un furto perché utilizza le risorse del sito linkato. È come se dovessimo entrare nella nostra macchina e andare via con la benzina che abbiamo risucchiato dalla macchina del nostro vicino.

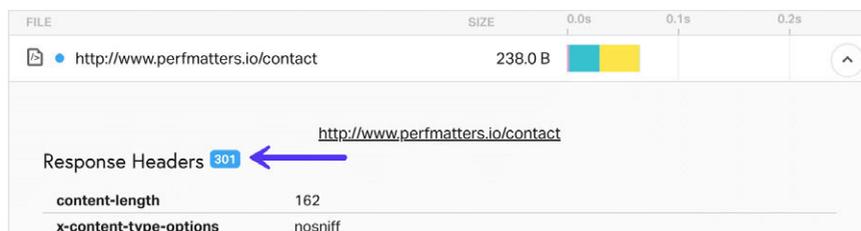
*L'hotlinking è come andare in giro in macchina con la benzina risucchiata dalla macchina del vostro vicino.*

L'hotlinking può costituire un **enorme drenaggio di risorse per il server di destinazione**. Immaginate di essere su un host WordPress condiviso e l'Huffington Post si collega improvvisamente alle vostre immagini. Il vostro sito potrebbe passare da un paio di centinaia di query all'ora a duecentomila. Ciò potrebbe comportare addirittura la sospensione del vostro account di hosting. Questo è un buon motivo non solo per utilizzare un host dalle alte prestazioni (che può gestire picchi a singhiozzo come questo), ma anche per abilitare la protezione dagli hotlink.

## Riducete al Minimo i Reindirizzamenti e Aggiungeteli a Livello di Server

Dovete sempre prestare attenzione al numero di redirect. I reindirizzamenti semplici come un singolo redirect 301, da HTTP a HTTPS o da www a non-www (e viceversa) vanno bene. E molte di volte sono necessari in alcune aree del vostro sito web. Tuttavia, ognuno ha un peso sulle prestazioni del vostro sito. E se iniziate ad accumulare redirect uno sopra l'altro, è importante che vi rendiate anche conto di come questi influiscono sul vostro sito. Questo vale per i redirect di pagine e post, per i redirect delle immagini, e per tutto il resto.

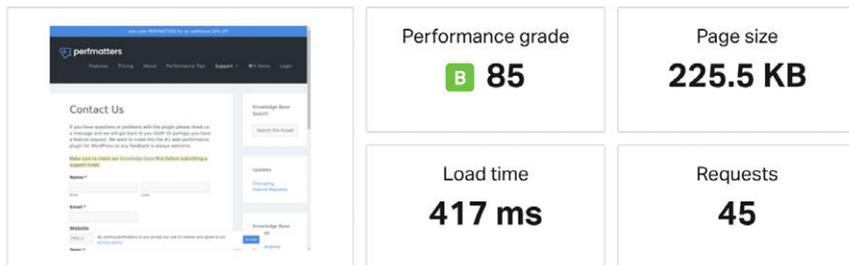
Un redirect genererà un 301 o 302 sullo stato dell'header della risposta.



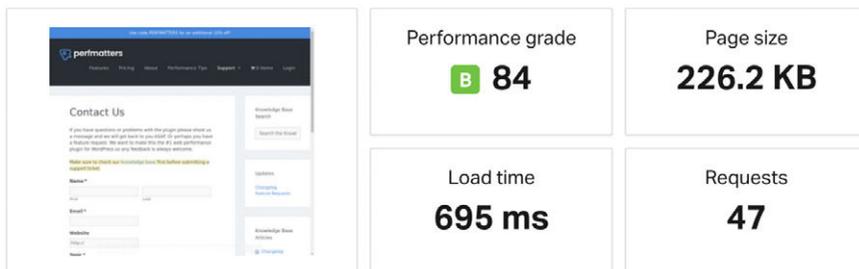
The screenshot shows a browser's developer tools interface. At the top, the address bar displays the URL `http://www.perfmatters.io/contact` with a size of 238.0 B. Below the address bar, the 'Response Headers' section is expanded, showing a status code of 301. A blue arrow points from the '301' status code to the URL `http://www.perfmatters.io/contact` in the header list. The header list includes:

Header	Value
content-length	162
x-content-type-options	nosniff

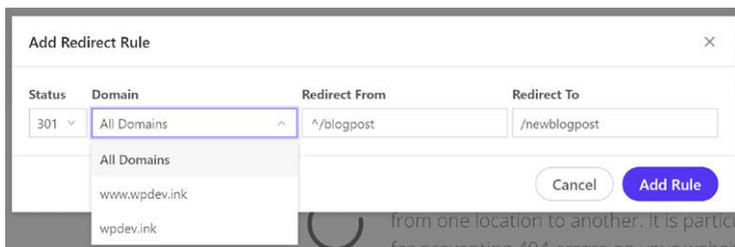
Quanto influiscono i redirect sul vostro sito? Facciamo un piccolo test. Prima di tutto, abbiamo eseguito uno speed test sulla nostra pagina dei contatti: <https://perfmatters.io/contact/>. Come potete vedere qui sotto, abbiamo ottenuto un tempo di caricamento totale di **417 ms**.



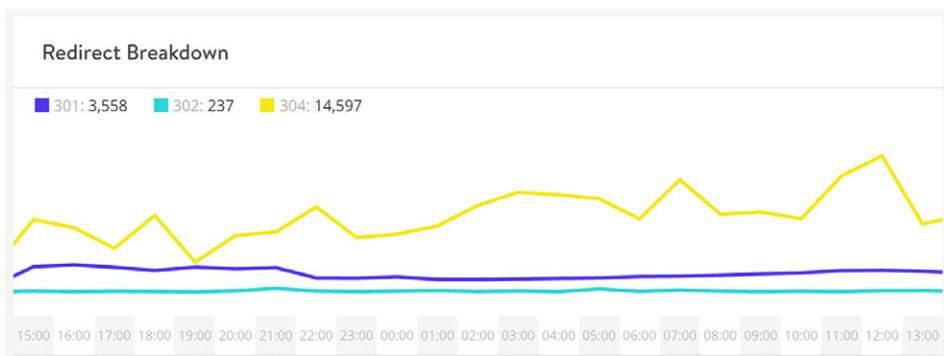
Quindi abbiamo modificato leggermente l'URL ed eseguito un altro speed test per vedere l'impatto di più redirect. <http://www.perfmatters.io/contact>. Come potete vedere, per essere caricata la stessa pagina ora richiede **695 ms**. È un aumento del 66%. Wow!



L'utilizzo di plugin di WordPress gratuiti per implementare i reindirizzamenti può a volte causare problemi di prestazioni poiché molti di essi utilizzano la funzione `wp_redirect`, che richiede l'esecuzione di codice e risorse aggiuntive. Alcuni plugin aggiungono anche dati a caricamento automatico alla tabella `wp_options`, cosa che aumenta le dimensioni del database. Quello che bisognerebbe fare è aggiungere i redirect a livello di server. Ed è quello che consentiamo di fare in MyKinsta con il nostro strumento di gestione delle regole di reindirizzamento.



Nel nostro strumento di analisi di MyKinsta potete vedere una suddivisione completa del numero di reindirizzamenti che vengono registrati sui vostri siti. Qui di seguito si vede il numero totale di redirect 301, 302 e 304.



## Tenete Sotto Controllo i Cron Job

I CRON job (WP-Cron) vengono utilizzati per pianificare attività ripetitive per il vostro sito WordPress. Purtroppo, nel tempo possono sfuggire al controllo e causare problemi di prestazioni. Potete utilizzare il plugin gratuito WP Control per effettuare una verifica di tutti i Cron Job attivi sul vostro sito.

Abbiamo inoltre riscontrato problemi di prestazioni nel gestore dei Cron integrato in WordPress: WP-Cron. Se un sito non ha abbastanza PHP worker, può accadere che arrivi una richiesta, WordPress genera il cron, ma il cron deve aspettare il worker, e quindi rimane lì in attesa. Un approccio migliore è quello di disabilitare WP-Cron e utilizzare invece il cron di sistema. Questa soluzione è consigliata anche nel manuale ufficiale dei Plugin.

Per disabilitare WP-Cron, aggiungete l'istruzione che segue al vostro file `wp-config.php`, subito prima della riga che dice "Finito, interrompere le modifiche! Buon blogging".  
Nota: questo disabiliterà l'esecuzione di WP-Cron al caricamento della pagina, ma non quando lo invocate direttamente tramite `wp-cron.php`.

```
define('DISABLE_WP_CRON', true);
```

```
60  /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66  $table_prefix = 'wp_';
67
68  define('DISABLE_WP_CRON', true); ←
69
70  /* That's all, stop editing! Happy blogging. */
71
72  /** Absolute path to the WordPress directory. */
73  if ( ! defined( 'ABSPATH' ) )
74      define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76  /** Sets up WordPress vars and included files. */
77  require_once ABSPATH . 'wp-settings.php';
78
```

Line 68, Column 33

Dovrete quindi pianificare `wp-cron.php` dal vostro server. Se siete clienti di Kinsta, i cron di sistema sono già abilitati di default e vengono eseguiti ogni 15 minuti. Potete anche aumentarne la frequenza contattando il nostro team di supporto. Se avete familiarità con SSH, potete gestire i cron del server dalla riga di comando

## Aggiungete gli Header Cache-Control ed Expires (Determinano la Durata della Cache)

Ad ogni script del vostro sito WordPress deve essere assegnato un header HTTP cache (o dovrebbe). Questo **stabilisce quando scade la cache sul file**. Se non è assegnato questo header, accertatevi che il vostro host WordPress abbia impostato correttamente le intestazioni `cache-control` e `expires`. Se non lo fate, probabilmente vedrete avvisi sulla necessità di aggiungere header `expires` o l'avviso di sfruttare il caching del browser negli strumenti di test della velocità.

Mentre l'header `cache-control` attiva il caching dal lato client e imposta la durata massima di una risorsa, l'header `expires` viene utilizzato per stabilire un momento preciso nel tempo a partire dal quale la risorsa non è più valida. Sebbene le intestazioni possano essere utilizzate insieme, non è necessario aggiungerle entrambe. `cache-control` è più recente ed è di solito il metodo consigliato.

Kinsta aggiunge automaticamente le intestazioni HTTP della cache su tutte le richieste del server e, se utilizzate un CDN, molto probabilmente anche quest'ultimo aggiungerà queste intestazioni.

Name	Headers	Preview	Response	Timing
editwp.com				
admin-bar.min.js?ver=5.0-beta3-43867				
skip-link-focus-fix.js?ver=20151215				
wp-embed.min.js?ver=5.0-beta3-43867				
be9a4e9519964b933dea6723d480c95a?s=52&d=mm&r=g				
be9a4e9519964b933dea6723d480c95a?s=128&d=mm&r=g				
wp-emoji-release.min.js?ver=5.0-beta3-43867				
data:application/fo...				
url-metrics				
tpc-check.html				
postmessage.js				
tpc-check-add-cookie.js				

12 requests | 6.5 KB transferred | Finish: 1.19 s | DOMContentLoaded: 452 ms | ...

▼ Response Headers

- access-control-allow-origin: \*
- cache-control: max-age=315360000
- content-encoding: gzip
- content-type: application/javascript; charset=UTF-8
- date: Wed, 07 Nov 2018 03:17:34 GMT
- etag: W/"5bdc7194-2efa"
- expires: Thu, 31 Dec 2037 23:55:55 GMT
- last-modified: Fri, 02 Nov 2018 15:47:32 GMT
- server: nginx
- status: 200
- vary: Accept-Encoding
- x-content-type-options: nosniff

Se al vostro server mancano le intestazioni, potete aggiungerle manualmente.

## Aggiungere l'Header Cache-Control in Nginx

Potete aggiungere gli header `cache-control` in Nginx inserendo il codice che segue alla posizione o al blocco del file di configurazione del server.

```
location ~* \.(js|css|png|jpg|jpeg|gif|svg|ico)$ {
    expires 30d;
    add_header Cache-Control "public, no-transform";
}
```

## Aggiungere l'Header Expires in Nginx

Potete aggiungere gli header `expires` in Nginx aggiungendo quanto segue al vostro blocco del server. In questo esempio, potete vedere come specificare tempi di scadenza diversi in base ai tipi di file.

```
location ~* \.(jpg|jpeg|gif|png|svg)$ {
    expires 365d;
}

location ~* \.(pdf|css|html|js|swf)$ {
    expires 2d;
}
```

## Aggiungere l'Header Cache-Control in Apache

Potete aggiungere gli header `cache-control` in Apache inserendo il codice che segue nel vostro file `.htaccess`. Possono essere aggiunti dei frammenti di codice nella parte superiore o inferiore del file (prima di `#BEGIN WordPress` o dopo `#END WordPress`).

```
<filesMatch ".(ico|pdf|flv|jpg|jpeg|png|gif|svg|js|css|swf)$">
Header set Cache-Control "max-age=84600, public"
</filesMatch>
```

## Aggiungere l'Header Expires in Apache

Potete aggiungere gli header `expires` in Apache inserendo il codice che segue nel vostro file `.htaccess`.

```
## EXPIRES HEADER CACHING ##
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access 1 year"
```

```

ExpiresByType image/jpeg "access 1 year"
ExpiresByType image/gif "access 1 year"
ExpiresByType image/png "access 1 year"
ExpiresByType image/svg "access 1 year"
ExpiresByType text/css "access 1 month"
ExpiresByType application/pdf "access 1 month"
ExpiresByType application/javascript "access 1 month"
ExpiresByType application/x-javascript "access 1 month"
ExpiresByType application/x-shockwave-flash "access 1 month"
ExpiresByType image/x-icon "access 1 year"
ExpiresDefault "access 2 days"
</IfModule>

## EXPIRES HEADER CACHING ##

```

È importante notare che è possibile aggiungere le intestazioni HTTP della cache solo sulle risorse che risiedono sul proprio server. Se ricevete un avviso al riguardo, forse dovrete sfruttare il caching del browser su una richiesta di terze parti. Non c'è null'altro che possiate fare, dato che non avete il controllo delle richieste sul loro server. Tra queste risorse ricordiamo lo script di Google Analytics e i pixel di marketing, come Facebook e Twitter.

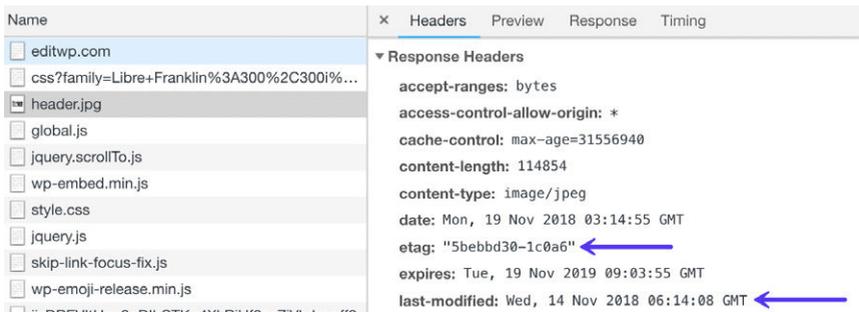
Se state cercando di risolvere questo problema con lo script di Google Analytics, potete ospitarlo localmente o sul vostro CDN (anche se questa soluzione non è supportata ufficialmente) con un plugin come Perfmatters (disponibile all'indirizzo [perfmatters.io](https://perfmatters.io)) o WP Rocket (disponibile all'indirizzo [wp-rocket.me](https://wp-rocket.me)).

## Aggiungere gli Header Last-Modified e ETag (Convalida della Cache)

Dopo questi, abbiamo altri due set di intestazioni, `last-modified` e `etag`.

Mentre gli header `cache-control` ed `expires` aiutano il browser a determinare **se il file è stato modificato** dall'ultima volta in cui è stato richiesto (o piuttosto convalidano la

cache), gli header `last-modified` ed `ETag` **convalidano e impostano la lunghezza della cache** e devono essere inclusi in ogni risposta del server di origine. Se questi header non sono impostati correttamente, potreste visualizzare un avviso che vi avverte di “Specificare un validatore della cache”.



Queste intestazioni dovrebbero essere inserite su ogni risposta del server di origine, poiché entrambe **convalidano e impostano la durata della cache**. Se le intestazioni non vengono trovate, sarà generata ogni volta una nuova richiesta per la risorsa, il che aumenta il carico sul server. L'utilizzo delle intestazioni di caching garantisce che le richieste successive non debbano essere caricate dal server, risparmiando così larghezza di banda e migliorando le prestazioni per l'utente.

Kinsta aggiunge automaticamente le intestazioni descritte a tutte le richieste del server e, se utilizzate un CDN, probabilmente anche questo aggiungerà le intestazioni. Proprio come avviene con `cache-control` ed `expires`, non è possibile impostare manualmente queste intestazioni HTTP su risorse esterne

## L'Header Last-Modified

L'header **last-modified** viene generalmente inviato automaticamente dal server. È un'intestazione che **generalmente non è necessario aggiungere manualmente**. Viene inviata per vedere se il file nella cache del browser è stato modificato dall'ultima volta in cui è stato richiesto. Potete visualizzare la richiesta dell'header in Pingdom o utilizzare Chrome DevTools per vedere il valore dell'intestazione `last-modified`.

## L'Header ETag

L'header **ETag** è molto simile all'intestazione `last-modified`. Anche questo viene utilizzato per convalidare la cache di un file. Se avete in esecuzione Apache 2.4 o versione successiva, l'header ETag viene aggiunto automaticamente utilizzando la direttiva `FileETag`. E per quel che riguarda NGINX, l'header ETag è stata abilitato di default dal 2016.

È possibile abilitare manualmente l'header ETag in Nginx utilizzando il seguente codice.

```
etag on
```

## Aggiungere un Header Vary: Accept-Encoding

L'header `vary: Accept-Encoding` deve essere inserito in ogni risposta del server di origine, in quanto indica al browser se il client può gestire le versioni compresse del contenuto. Se questo non è impostato correttamente, potreste visualizzare un avviso "Specify a Vary: Accept-Encoding Header".

Ad esempio, supponiamo che abbiate un vecchio browser senza compressione gzip e un browser moderno. Se non utilizzate l'header `vary: Accept-Encoding`, il vostro server o il vostro CDN potrebbero memorizzare nella cache la versione non compressa e consegnarla per errore al browser moderno, il che a sua volta danneggerebbe le prestazioni del vostro sito WordPress. Utilizzando questa intestazione si può essere certi che il server o il CDN forniscano la versione appropriata.

Name	×	Headers	Preview	Response	Timing
editwp.com					
css?family=Libre+Franklin%3A300%2C300i%...					
header.jpg					
global.js					
jquery.scrollTo.js					
wp-embed.min.js					
style.css					
jquery.js					
skip-link-focus-fix.js					

×	Headers	Preview	Response	Timing
	▼ Response Headers			
	<b>content-encoding:</b> gzip			
	<b>content-type:</b> text/html; charset=UTF-8			
	<b>date:</b> Mon, 19 Nov 2018 03:14:57 GMT			
	<b>server:</b> nginx			
	<b>status:</b> 200			
	<b>vary:</b> Accept-Encoding ←			
	<b>x-content-type-options:</b> nosniff			
	<b>x-kinsta-cache:</b> HIT			

Kinsta aggiunge automaticamente le intestazioni a tutte le richieste del server, e se utilizzate un CDN, molto probabilmente le aggiungerà anche quest'ultimo. Proprio come con le altre intestazioni della cache di cui abbiamo parlato sopra, non è possibile impostare manualmente questa intestazione su risorse esterne.

## Aggiungere l'Header Vary: Accept-Encoding in Apache

Potete aggiungere l'header vary: `Accept-Encoding` in Apache aggiungendo quanto segue al vostro file `.htaccess`.

```
<IfModule mod_headers.c>
  <FilesMatch "(.js|css|xml|gz|html)$">
    Header append Vary: Accept-Encoding
  </FilesMatch>
</IfModule>
```

## Aggiungere l'Header Vary: Accept-Encoding in Nginx

Potete aggiungere l'header vary: `Accept-Encoding` in Nginx aggiungendo il seguente codice al vostro file di configurazione. Tutti i file di configurazione di Nginx si trovano nella directory `/etc/nginx/`. Il file di configurazione principale è `/etc/nginx/nginx.conf`.

```
gzip_vary on
```

## Modificare il Limite di Memoria di WordPress in wp-config.php

Come affermato nel Codex di WordPress, con la versione 2.5 di WordPress, l'opzione `WP_MEMORY_LIMIT` consente di specificare la quantità massima di memoria che può essere consumata da PHP. Questa impostazione potrebbe essere necessaria nel caso in cui venga visualizzato un messaggio come "Allowed memory size of xxxxxx bytes exhausted".

Di default, WordPress tenterà di aumentare la memoria allocata a PHP a 40 MB per un singolo sito e 64 MB per il multisite. I limiti di memoria sono impostati nel file `./wp-includes/default-constants.php`, alle righe 32 – 44 (fonte).

Anche il vostro provider di hosting imposta sul server il `memory_limit` di PHP, ma tratta di due cose diverse. Da Kinsta impostiamo il valore predefinito di `memory_limit` a 256M. Se vi imbattete nell'errore della memoria esaurita, potete provare ad aumentare il limite di memoria PHP in WordPress.

Aggiungete quanto segue al vostro file `wp-config.php`, appena prima della riga che dice “Finito, interrompere le modifiche! Buon blogging”.

```
define( 'WP_MEMORY_LIMIT', '256M' );
```

Jan Reilink ha pubblicato un ottimo post in cui descrive in modo più dettagliato il problema dei limiti di memoria di WordPress. Offre anche una variante del codice che potreste utilizzare. Invece di assegnare il valore manualmente, potete impostarlo allo stesso valore `memory_limit` di PHP.

```
define( 'WP_MEMORY_LIMIT', ini_get( 'memory_limit' ) );
```

CAPITOLO 15:

# Consigli sull'Ottimizzazione del Front-End e Sui Servizi Esterni



**Ora analizzeremo alcune soluzioni per accelerare WordPress ottimizzando il front-end. Il front-end in genere riguarda tutto ciò che viene gestito interamente dal browser lato client, come CSS, JavaScript, immagini, ecc. Ciò riguarda anche l'analisi dei servizi esterni caricati sul vostro sito e il loro impatto sul tempo di caricamento complessivo.**

Per quel che riguarda l'ottimizzazione del front-end, ecco due degli obiettivi più importanti che dovrete porvi:

- **Ridurre le dimensioni complessive della pagina web.** La dimensione dei vostri file CSS, JavaScript e file immagine sono importanti. Un sito web da 4 MB in genere viene caricato molto più lentamente di un sito da 1 MB. Tuttavia, Paul Calvano ha pubblicato un ottimo articolo sull'impatto del peso della pagina sui tempi di caricamento e su quanto sia importante che vi accertiate che non sia l'unica cosa che monitorate, in quanto, a volte, può essere fuorviante.
- **Ridurre le richieste HTTP e i servizi esterni.** Con HTTP/2 è possibile inviare contemporaneamente più richieste e risposte utilizzando una singola connessione TCP. Sebbene ciò sia ottimo per le prestazioni, la riduzione delle richieste HTTP può comunque contribuire ad accelerare il vostro sito WordPress. E questo include anche la riduzione del numero totale di richieste e servizi esterni. Si tratta di cose come risoluzione del DNS, connessioni TLS e latenza di rete.

# Primo Speed Test sul Sito WordPress Come Punto di Partenza

Quando si va ad ottimizzare il front-end di un sito, è sempre bene iniziare con un punto di riferimento. Questo di solito significa che dovete eseguire un test di velocità. Ci sono moltissimi modi per farlo. Nel nostro sito [web.kinsta.com](http://web.kinsta.com), abbiamo delle guide approfondite su come utilizzare correttamente Pingdom e GTmetrix.

URL:  Test from:  [START TEST](#)

Your Results:

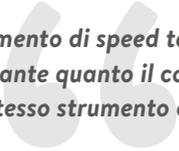
[DOWNLOAD HAR](#) [SHARE RESULT](#)

Performance grade	<b>B 88</b>	Page size	<b>243.0 KB</b>
Load time	<b>398 ms</b>	Requests	<b>36</b>

Ecco alcune cose da tenere a mente quando si effettuano i test sulla velocità:

## 1. Scegliete uno Strumento e Tenetevelo Stretto

Noi siamo grandi fan di Pingdom, GTmetrix, WebPageTest, PageSpeed Insights e Chrome DevTools. Tuttavia, non importa tanto quale strumento di speed test utilizzate. L'importante è utilizzarlo in modo coerente. Ognuno di questi strumenti fornisce un modo diverso di misurare la velocità, quindi sceglierne e usate sempre lo stesso durante tutta la fase di testing e ottimizzazione. Anche Google dice di sceglierne solo uno.



*Lo strumento di speed test che scegli non è tanto importante quanto il continuare ad utilizzare lo stesso strumento durante tutti i test.*

## **2. Non Fatevi Ossessionare dal Punteggio Perfetto**

Molti strumenti come Google PageSpeed Insights assegnano un certo punteggio di velocità o di prestazioni. È importante ricordare che il punteggio non conta tanto quanto la velocità del sito e delle prestazioni percepite dall'utente. Il punteggio è lì per aiutarvi a valutare quello che state facendo. Ma l'ossessione per un perfetto 100/100, o in alcuni casi un valutazione di A, potrebbe comportare solo una perdita di tempo. E i siti più grandi, con un sacco di script e annunci esterni, non otterranno mai un punteggio perfetto, ed è una cosa assolutamente normale.

## **3. La Location del Test È Importante**

La località scelta quando si effettua un test di velocità conta molto. Come abbiamo visto in una sezione precedente, il motivo è che molto dipende dalla posizione del data center. TTFB, latenza di rete, tutto entra in gioco. Quindi, provate il vostro sito sia da una posizione vicina al vostro data center, che da una posizione lontana. Questo vi aiuterà anche a vedere quale impatto può avere un CDN sul vostro sito WordPress.

## **4. Provate Più Volte per Neutralizzare gli Effetti della Cache**

Come abbiamo visto prima nella sezione sulla cache, se la cache è stata cancellata di recente o è scaduta sul vostro host WordPress o sul CDN, registrerà un "MISS" sull'intestazione HTTP. Questo significa che il vostro sito o la vostra risorsa non vengono serviti dalla cache.

FILE	SIZE	0.0s	0.1s	0.2s
<a href="https://perfmatters.io/">https://perfmatters.io/</a>	9.6 KB			
<a href="https://perfmatters.io/">https://perfmatters.io/</a>				
Response Headers <span style="background-color: #c8e6c9; padding: 2px;">200</span>				
<b>status</b>	200			
<b>x-content-type-options</b>	nosniff			
<b>content-encoding</b>	gzip			
<b>set-cookie</b>	PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1...			
<b>x-kinsta-cache</b>	MISS			
<b>expires</b>	Thu, 19 Nov 1981 08:52:00 GMT			

Per valutare correttamente la velocità di tutto il vostro sito, dovete fare in modo che tutto carichi dalla cache, cioè che la pagina iniziale e tutte le risorse registrino un “HIT”. A volte è necessario eseguire lo speed test più volte, e quindi fare una media.

FILE	SIZE	0.0s	0.1s	0.2s
<a href="https://perfmatters.io/">https://perfmatters.io/</a>	9.6 KB			
<a href="https://perfmatters.io/">https://perfmatters.io/</a>				
Response Headers <span style="background-color: #c8e6c9; padding: 2px;">200</span>				
<b>status</b>	200			
<b>x-content-type-options</b>	nosniff			
<b>content-encoding</b>	gzip			
<b>set-cookie</b>	PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1...			
<b>x-kinsta-cache</b>	HIT			
<b>expires</b>	Thu, 19 Nov 1981 08:52:00 GMT			

Ora passiamo ad alcune ottimizzazioni del front-end di WordPress.

## Rimuovete le Query String

Un avvertimento o una raccomandazione che gli utenti vedono di frequente negli strumenti di speed test è che è necessario rimuovere le query string. Di cosa si tratta? In pratica funziona così: i file CSS e JavaScript di solito hanno la versione del file alla fine dei loro URL, come <https://domain.com/file.min.css?ver=4.5.3>.

Alcuni server e server proxy non sono in grado di memorizzare le query string. Quindi, rimuovendoli, si può migliorare il caching.

Potete rimuovere le query string aggiungendo manualmente il codice che segue al file `functions.php` del vostro tema.

```
function remove_query_strings() {  
    if(!is_admin()) {  
        add_filter('script_loader_src', 'remove_query_strings_split', 15);  
        add_filter('style_loader_src', 'remove_query_strings_split', 15);  
    }  
}  
  
function remove_query_strings_split($src){  
    $output = preg_split("/(&ver|\?ver)/", $src);  
    return $output[0];  
}  
  
add_action('init', 'remove_query_strings');
```

Oppure potete utilizzare un plugin premium come Perfmatters (disponibile su [perfmatters.io](https://perfmatters.io)) per rimuovere le query string con un solo clic.

## Con le Query String

Ecco un esempio di script che caricano le query string.

Name	Status
 editwp.com	200
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600%2C600i...	200
 style.css?ver=4.9.8	200
 jquery.js?ver=1.12.4	200
 header.jpg	200
 skip-link-focus-fix.js?ver=1.0	200
 global.js?ver=1.0	200
 jquery.scrollTo.js?ver=2.1.2	200
 jizDREVItHgc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200

## Senza le Query String

Name	Status
 editwp.com	200
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600%2C600i...	200
 style.css ←	200
 jquery.js	200
 header.jpg	200
 skip-link-focus-fix.js	200
 global.js	200
 jquery.scrollTo.js	200
 jizDREVItHgc8qDIbSTKq4XkRiUf2zcZiVbJ.woff2	200

Ecco un esempio di script cui sono state rimosse le query string.

Tuttavia, prima di eliminare le query string dal vostro sito, è importante sapere perché vengono utilizzate. Le versioni dei file vengono in genere aggiunte dagli sviluppatori di WordPress per evitare problemi di caching.

Ad esempio, se uno sviluppatore di plugin invia un aggiornamento e cambia `style.css` da `?ver=4.6` a `?ver=4.7`, questo sarà considerato come un URL completamente nuovo e non sarà memorizzato nella cache. Se rimuovete le query string e aggiornate un plugin, potrebbe essere mantenuta la versione memorizzata nella cache. In alcuni casi, questo potrebbe rovinare l'aspetto del sito fino alla scadenza della risorsa presente nella cache, o al suo completo svuotamento.

Inoltre, anche alcuni CDN possono memorizzare nella cache le query string. Il CDN di Kinsta può farlo e lo fa di default. Quindi, se siete un cliente di Kinsta, sappiate che le query string sono già memorizzate nella cache sui vostri asset.

## Eliminate JavaScript e CSS che Bloccano la Visualizzazione della Pagina

Potrebbe essere visualizzato un avviso relativo a JavaScript e CSS che bloccano il rendering quando i file impediscono alla pagina di caricare il più rapidamente possibile. Specifici JS e CSS sono a volte condizionali, cioè non sono necessari per visualizzare i contenuti above-the-fold. È possibile impedire che queste risorse blocchino il rendering utilizzando gli attributi `async` e `defer`.

Opportunity	Estimated Savings
1 Eliminate render-blocking resources	 2.24 s ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more](#).

Per eliminare JavaScript e CSS che bloccano il rendering, dovete procedere come segue:

## Rimuovete JS dal Critical Rendering Path

Per spostare JavaScript dal Critical Rendering Path vengono di solito utilizzati gli attributi `defer` o `async` agli elementi HTML script che invocano le risorse JavaScript.

- L'**attributo `async`** dice al browser di iniziare subito a scaricare la risorsa senza rallentare il parsing HTML. Una volta che la risorsa è disponibile, il parsing HTML viene messo in pausa in modo che la risorsa possa essere caricata.
- L'**attributo `defer`** dice al browser di attendere il download della risorsa fino al completamento del parsing HTML. Una volta che il browser ha finito con l'HTML, scaricherà e eseguirà tutti gli script posticipati nell'ordine in cui compaiono nel documento.

## Ottimizzate la Consegna delle Risorse CSS

Ottimizzare la consegna di CSS in sostanza significa che dovete capire come far sì che non blocchi il rendering.

- Identificate gli stili necessari per il rendering dei contenuti above-the-fold e consegnate quegli stili in linea con l'HTML.
- Utilizzate CSS in modo condizionale sui dispositivi, cioè solo quando è necessario.
- Caricate il CSS rimanente in modo asincrono.

Fare tutto questo può a volte essere un po' complicato e richiede sicuramente qualche aggiustamento a seconda degli script caricati sul vostro sito. Ecco alcuni plugin di WordPress che possono essere di aiuto:

- Autoptimize (gratuito, disponibile su WordPress.org)
- Async JavaScript (gratuito, disponibile su WordPress.org)
- Hummingbird (gratuito, disponibile su WordPress.org)

Per una spiegazione più dettagliata e approfondita, vi consigliamo di leggere il nostro post sull'eliminazione delle risorse JavaScript e CSS che bloccano il rendering.

## Combinare CSS e JavaScript esterni in WordPress

L'avviso che consiglia di combinare i CSS esterni viene in genere visualizzato quando si utilizza un CDN perché si ospitano i file CSS su un dominio esterno, come ad esempio `cdn.domain.com`. In passato, un modo rapido per risolvere questo problema era quello di concatenare i file CSS o combinarli in modo che venissero caricati in un'unica richiesta.

Tuttavia, se si opera su HTTPS con un provider che supporta HTTP/2, questo avviso non è più rilevante come un tempo. Con HTTP/2, infatti, è possibile caricare più file CSS in parallelo su una singola connessione. E oltre l'86% dei browser supporta HTTP/2.

Ma questo non significa necessariamente che questa ottimizzazione sia completamente obsoleta. In alcuni casi, abbiamo visto che accelera ancora i siti WordPress. Dipende dalla dimensione e dal numero dei file. Per questo, è una ottimizzazione che consigliamo comunque di testare sui vostri siti.

Uno dei metodi più semplici per combinare i file CSS e JavaScript esterni è quello di utilizzare il plugin gratuito Autoptimize. Dopo averli combinati, vedrete un file "autoptimize\_XXXX.css" o "autoptimize\_XXXX.js". Il plugin supporta anche il caricamento dal vostro CDN. È qualcosa che potete fare anche con il plugin WP Rocket (disponibile su [wp-rocket.me](https://wp-rocket.me)).

Name	Status	Domain
 editwp.com	200	editwp.com
 autooptimize_23d90d8a1eaacef4363d9e02c50ae363.css ←	200	mk0editwpn6...
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600...	200	fonts.googlea...
 jquery.js?ver=1.12.4	200	mk0editwpn6...
 header.jpg	200	mk0editwpn6...
 autooptimize_3fde0f2c0341cdc5afd7cc0c0e403f92.js ←	200	mk0editwpn6...
<input type="checkbox"/> jizDREVItHgc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200	fonts.gstatic.c...
<input type="checkbox"/> jizAREVItHgc8qDlbSTKq4XkRi24_Sl0q1vjitOh.woff2	200	fonts.gstatic.c...
<input type="checkbox"/> jizAREVItHgc8qDlbSTKq4XkRi20-Sl0q1vjitOh.woff2	200	fonts.gstatic.c...

## Minimizzate le Risorse HTML, CSS e JavaScript

Possiamo ridurre la quantità di dati che il browser deve scaricare minimizzando le risorse HTML, CSS e JavaScript. La minification è il processo di rimozione di caratteri non necessari, come commenti e spazi bianchi, dal codice sorgente. Questi caratteri sono estremamente utili per lo sviluppo, ma non servono al browser per il rendering della pagina.

### HTML Non Minimizzato

Ecco un esempio di codice HTML non minimizzato.

```

1 <!DOCTYPE html>
2 <html lang="en-US" class="no-js no-svg">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="profile" href="http://gmpg.org/xfn/11">
7
8 <script>(function(html){html.className = html.className.replace(/\bno-
9 js\b/, 'js')})(document.documentElement);</script>
10 <title>editwp &#8211; Just another WordPress site</title>
<link rel='dns-prefetch' href='//fonts.googleapis.com' />
```

# HTML Minimizzato

Ecco invece un esempio di codice HTML minimizzato.

```
1 <!DOCTYPE html><html lang="en-US" class="no-js no-svg"><head><meta
  charset="UTF-8"><meta name="viewport" content="width=device-width,
  initial-scale=1"><link rel="profile" href="http://gmpg.org/xfn/11">
  <script>(function(html){html.className = html.className.replace(/\bno-
  js\b/, 'js')})(document.documentElement);</script> <link type="text/css"
  media="all" href="https://mk0editwpn6qbfrow5o9.kinstacdn.com/wp-
  content/cache/autoptimize/css/autoptimize_499850c34eb7b4e7cbd9d8b41b2a5
  7d6.css" rel="stylesheet" /><style type="text/css"
  media="print">#wpadminbar{display:none}</style><style type="text/css"
  media="screen">html{margin-top:32px !important}* html body{margin-
```

Potete utilizzare uno dei plugin gratuiti Autoptimize o WP Rocket per ridurre facilmente i vostri file.

## Utilizzate Domini Senza Cookie

In genere, quando si servono contenuti come immagini, JavaScript, CSS, non vi è alcun motivo che questi vengano accompagnati da un cookie HTTP, in quanto questo crea un sovraccarico inutile. Una volta che il server imposta un cookie per un determinato dominio, tutte le successive richieste HTTP per quel dominio devono includere il cookie. Questo avviso si vede di solito su siti con un numero elevato di richieste.

Molte volte potete ignorare questo avviso poiché i nuovi protocolli come HTTP/2 ne riducono l'importanza. Il costo di una nuova connessione è in genere maggiore rispetto alla trasmissione di tutto sulla stessa connessione.

Un modo semplice per eliminare questo avviso è quello di utilizzare un provider CDN che può ignorare i cookie, come anche eliminare i cookie che impediscono del tutto al client di ricevere l'header di risposta Set-Cookie. KeyCDN è un provider CDN che offre questa funzionalità. Di default, potete vedere se le seguenti due opzioni sono abilitate. È un'alternativa semplice, che permette di evitare di spostare e configurare il proprio sito in modo che consegni risorse statiche da un sottodominio separato.

### Cache Cookies \*

By default, files with cookies are not cacheable. However, enabling this option will ignore the presence of cookies and therefore force the edge servers to cache these files.

### Strip Cookies \*

This feature strips the cookies received from the origin server. The client will not receive the `Set-Cookie` response header. **It is highly recommended to enable this directive if you enable Cache Cookies.**

Se utilizzate Cloudflare, non potete disabilitare i cookie sulle risorse servite attraverso la loro rete. CloudFlare inserisce il proprio cookie di sicurezza nell'intestazione. Ripetiamo, questi cookie sono molto piccoli e le conseguenze sulle prestazioni sono minime. Ma se utilizzate CloudFlare, non c'è modo di aggirare questo avviso.

Un secondo modo per aggirare l'avviso è riconfigurare il sito WordPress in modo che fornisca le risorse statiche da un nuovo dominio o sottodominio.

## Disabilitate gli Embed in WordPress

Quando hanno rilasciato WordPress 4.4, hanno unito la funzione oEmbed al core. Questo permette agli utenti di incorporare video, tweet e molte altre risorse sui propri siti semplicemente incollando un URL che WordPress converte automaticamente in un embed e fornisce un'anteprima live nell'editor visuale. Con l'aggiornamento, WordPress stesso è diventato un provider oEmbed.

Questa funzione è utile in molti casi e potreste anche decidere di mantenerla attiva.

Tuttavia, ciò significa anche che viene generata una richiesta HTTP aggiuntiva sul vostro sito WordPress per caricare il file `wp-embed.min.js`.

Name	Status	Domain
<input type="checkbox"/> editwp.com	200	editwp.com
<input type="checkbox"/> global.js	200	mk0editwpn...
<input type="checkbox"/> css?family=Libre+Franklin%3A300%2C300%2C400%2C400...	200	fonts.google...
<input type="checkbox"/> jquery.scrollTo.js	200	mk0editwpn...
<input checked="" type="checkbox"/> wp-embed.min.js ←	200	mk0editwpn...
<input type="checkbox"/> style.css	200	mk0editwpn...
<input type="checkbox"/> jquery.js	200	mk0editwpn...
<input checked="" type="checkbox"/> header.jpg	200	mk0editwpn...
<input type="checkbox"/> skip-link-focus-fix.js	200	mk0editwpn...
<input type="checkbox"/> wp-emoji-release.min.js	200	editwp.com

E questo file viene caricato in tutto il sito. Sebbene il file sia solo di 1,7 KB, cose come queste si sommano nel tempo. La sola richiesta è a volte più ingombrante della dimensione di download del contenuto.

Ci sono un paio di modi per disabilitare gli embed in WordPress. Lo potete fare con uno dei seguenti plugin:

- Disable Embeds (gratuito, disponibile su WordPress.org)
- Perfmatters (disponibile all'indirizzo perfmatters.io)

In alternativa, potete disattivare gli embed tramite codice. Per la sua lunghezza, lo troverete nel nostro sito all'indirizzo <https://kinsta.com/knowledgebase/disable-embeds-wordpress/>

## Disabilitare gli Emoji in WordPress

In modo simile agli embed, in WordPress 4.2 è stato aggiunto il supporto per gli emoji nel core per i browser più vecchi. Il problema maggiore di questa modifica è che viene generata una richiesta HTTP aggiuntiva sul vostro sito WordPress per caricare il file `wp-emoji-release.min.js`.

E questo file viene caricato in tutto il sito. Sebbene il file sia solo di 10,5 KB, è inutile se non utilizzate emoji sul vostro sito.

Name	Status	Domain
<input type="checkbox"/> editwp.com	200	editwp.com
<input type="checkbox"/> global.js	200	mk0editwpn...
<input type="checkbox"/> css?family=Libre+Franklin%3A300%2C300%2C400%2C400...	200	fonts.google...
<input type="checkbox"/> jquery.scrollTo.js	200	mk0editwpn...
<input type="checkbox"/> wp-embed.min.js	200	mk0editwpn...
<input type="checkbox"/> style.css	200	mk0editwpn...
<input type="checkbox"/> jquery.js	200	mk0editwpn...
<input checked="" type="checkbox"/> header.jpg	200	mk0editwpn...
<input type="checkbox"/> skip-link-focus-fix.js	200	mk0editwpn...
<input type="checkbox"/> wp-emoji-release.min.js	200	editwp.com
<input type="checkbox"/> jizDREvItHgc8qDlbSTKq4XkRiUf2zcZVbJ.woff2	200	fonts.gstatic....

Ci sono un paio di modi per disabilitare gli emojis in WordPress. Lo potete fare con uno dei seguenti plugin:

- Disable Emojis (gratuito, disponibile su WordPress.org)
- Perfmatters (disponibile all'indirizzo perfmatters.io)

In alternativa, potete disattivare gli emojis tramite codice. Per la sua lunghezza, lo troverete nel nostro sito all'indirizzo <https://kinsta.com/knowledgebase/disable-emojis-wordpress/>

## Come Velocizzare i Commenti di WordPress o Disabilitarli

Una sezione di commenti molto frequentata in un sito può causare molti problemi di prestazioni. Pensate solo alle risorse che servono per far funzionare i commenti:

- Viene interrogato un database per visualizzare i commenti esistenti.
- Vengono create voci nel database per ogni nuovo commento.
- I commenti e i metadati dei commenti vengono ricevuti ed elaborati dal browser del visitatore.
- Risorse esterne, come i Gravatar, vengono richieste, scaricate e caricate (richiedendo una risoluzione del DNS separata).
- In molti casi, devono essere scaricate ed elaborate risorse JavaScript e jQuery di grandi dimensioni per far sì che il sistema dei commenti funzioni regolarmente.

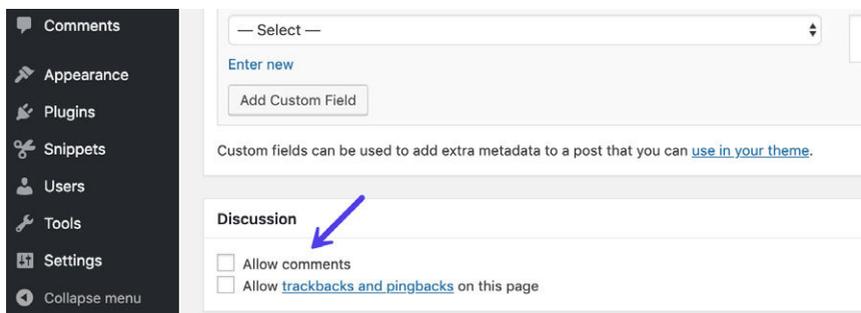
Ecco quattro diverse soluzioni per velocizzare i commenti di WordPress:

### Opzione 1 – Disabilitare i Commenti

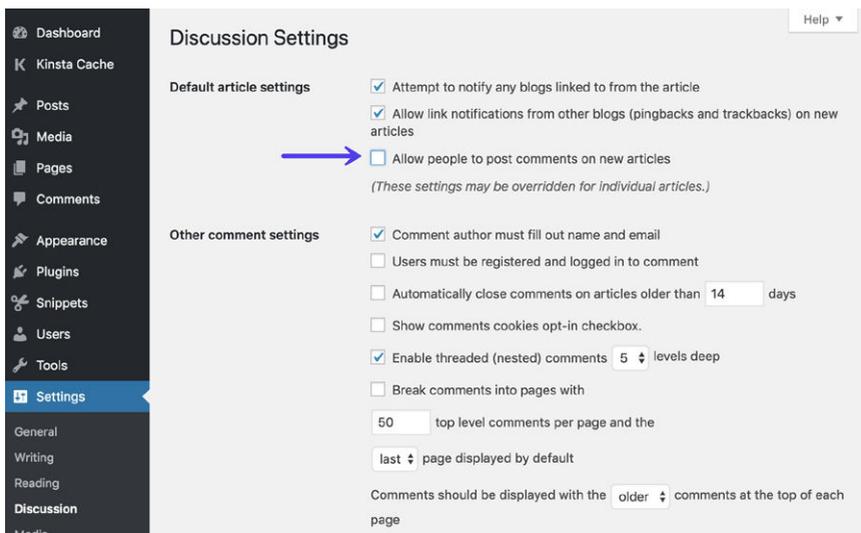
Se il vostro sito non riceve molti commenti e non pensi che apportino alcun valore, potrebbe essere meglio disabilitare del tutto i commenti. Ricordate che i commenti possono avere un impatto sulla vostra SEO, dato che in genere Google li sottopone a scansione come contenuti aggiuntivi nella pagina, pertanto dovrete approvare solo commenti di alta qualità. Date un'occhiata a questi tre semplici metodi per disabilitare i commenti:

## 1. Disabilitare i Commenti con le Opzioni di WordPress

Potete disattivare i commenti individualmente su un post o su una pagina mentre siete in editing, togliendo la spunta all'opzione "Permetti commenti", nella sezione "Discussione". Se non vedete questa sezione, scorrete la pagina verso l'alto e abilitatela nelle "Impostazioni schermata".



Potete anche disabilitare completamente le discussioni. Per farlo, cliccate su "Discussione" nelle Impostazioni della dashboard di WordPress, e quindi deselezionate l'opzione "Consenti la scrittura di commenti per i nuovi articoli". L'inconveniente di questo metodo è che non modifica le impostazioni degli articoli precedenti, ma solo dei nuovi.



Se avete bisogno di un maggior controllo o volete semplicemente disattivare tutti i commenti a livello globale, consigliamo di seguire uno dei metodi qui di seguito.

## 2. Disabilitare i Commenti con un Plugin

Uno dei metodi più semplici per disabilitare i commenti è quello di utilizzare un plugin gratuito come [Disable Comments](#) (disponibile su [WordPress.org](#)). Potete disabilitare i commenti ovunque con un solo clic, oppure disabilitarli su post, pagine o post di tipo media.

## 3. Disabilitare i Commenti con il Codice

Potete anche disabilitare i commenti via codice semplicemente rimuovendo la sezione dei commenti dal vostro tema WordPress.

Vi basta semplicemente sostituire nel vostro file `page.php` il seguente codice:

```
<?php comments_template( '', true ); ?>
```

Con questo codice:

```
<!-- Begin Comment  
<?php comments_template( '', true ); ?>  
End Comment -->
```

Questo rimuove i commenti dalle pagine. Per rimuoverli dai post del blog, cercate il file `single.php` e sostituite il seguente codice:

```
<?php comments_template(); ?>
```

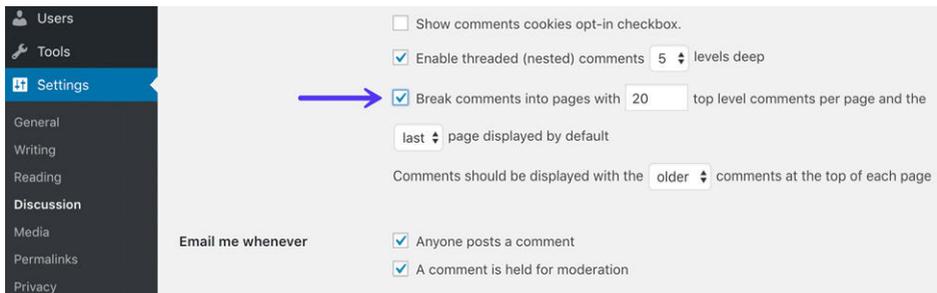
Con questo:

```
<!-- Begin Comment  
<?php comments_template(); ?>  
End Comment →
```

## Opzione 2 – Ottimizzare i Commenti Nativi di WordPress

La seconda opzione è quella di ottimizzare il sistema di commenti nativo di WordPress. Un metodo è quello di ridurre il numero di commenti caricati al caricamento iniziale della pagina.

1. Andate su Impostazioni → Discussione nell'area di amministrazione di WordPress.
2. Cercate la sezione Altre impostazioni commenti.
3. Selezionate la casella di controllo accanto a Dividi i commenti in pagine con, e aggiungete un valore per il numero di commenti che desiderate visualizzare al caricamento iniziale della pagina.



Un'altra opzione è quella di ospitare i Gravatar sul vostro CDN. Questo è l'approccio che abbiamo da Kinsta.

Di default, quando vengono caricati i commenti di WordPress, ogni singolo Gravatar richiede una richiesta HTTP. Quindi, se una pagina viene caricata con commenti da 50 diversi commentatori, saranno necessarie 50 richieste HTTP per scaricare tutti questi Gravatar. Come potete immaginare, questo può influire sulla vostra page speed. Per non parlare, poi, del fatto che abbiamo visto che la ricerca DNS esterna su [gravatar.com](https://www.gravatar.com) è a volte lenta e in alcuni casi va addirittura in timeout.

Se guardate ai Gravatar sul blog di Kinsta, vedrete che caricano da [Kinsta.com](https://www.kinsta.com) (e dal nostro CDN).



Assicuratevi sempre di testare la velocità del sistema di commenti di terze parti che state provando. Date un'occhiata a tutte le singole richieste che genera Disqus (come mostrato sotto). Sebbene la maggior parte di queste richieste venga caricata in modo asincrono, noterete comunque un tempo di caricamento maggiore per Disqus.

## Opzione 4 – Lazy Load dei Commenti

La quarta opzione consiste nel caricare i commenti in differita (“lazy load”) in modo che non rallentino il rendering iniziale della pagina. Ecco un paio di plugin che potreste provare:

- **Lazy Load for Comments** (gratuito, disponibile su WordPress.org): questo plugin vi permette di implementare il lazy load dei commenti nativi di WordPress.
- **Disqus Conditional Load** (gratuito, disponibile su WordPress.org): se volete utilizzare il sistema di commenti di Disqus, questo è un plugin imperdibile per il lazy load dei commenti.

## Disabilitate i Feed RSS di WordPress

Se sul vostro sito WordPress non utilizzate la sezione del blog, potete disabilitare i feed RSS. Sebbene questo non abbia un grande impatto sulle prestazioni, tutto può contribuire a migliorarle. Ed è anche una cosa in meno di cui vi dovrete preoccupare.

Ecco due diversi modi per disabilitare i feed RSS in WordPress:

### 1. Disabilitare i Feed RSS Feed con un Plugin

Il primo metodo per disabilitare un feed RSS WordPress è quello di utilizzare un plugin gratuito come Disable Feeds (disponibile su WordPress.org). Questo plugin disabilita tutti i feed RSS/Atom sul vostro sito WordPress, reindirizzando tutte le richieste.

Potete anche utilizzare un plugin premium come Perfmatters (sviluppato da un membro del team di Kinsta, disponibile su [perfmatters.io](https://perfmatters.io)), che vi consente sia di disabilitare i feed RSS, sia di disabilitare i link dei feed RSS, oltre a fornire altre opzioni di ottimizzazione per il vostro sito WordPress.

## 2. Disabilitare i Feed RSS con il Codice

Se non volete usare un plugin, potete anche disabilitare i feed RSS con il codice. Vi basta inserire il seguente codice nel file `functions.php` del vostro tema.

```
function itsme_disable_feed() {
    wp_die( __( 'No feed available, please visit the <a href="'. esc_url(
        home_url( '/' ) ) ."'>homepage</a>!' ) );
}

add_action( 'do_feed', 'itsme_disable_feed', 1);
add_action( 'do_feed_rdf', 'itsme_disable_feed', 1);
add_action( 'do_feed_rss', 'itsme_disable_feed', 1);
add_action( 'do_feed_rss2', 'itsme_disable_feed', 1);
add_action( 'do_feed_atom', 'itsme_disable_feed', 1);
add_action( 'do_feed_rss2_comments', 'itsme_disable_feed', 1);
add_action( 'do_feed_atom_comments', 'itsme_disable_feed', 1);
```

WordPress genera anche collegamenti ai feed RSS all'interno dell'intestazione della vostra pagina web. Potete fare un ulteriore passo in avanti e rimuovere questi collegamenti dal codice HTML della vostra pagina. Aggiungete il seguente codice nel file `functions.php` del vostro tema.

```
remove_action( 'wp_head', 'feed_links_extra', 3 );
remove_action( 'wp_head', 'feed_links', 2 );
```

# Utilizzate Prefetch e Preconnect

Suggerimenti e direttive sulle risorse come prefetch e preconnect possono costituire un ottimo modo per velocizzare dietro le quinte il funzionamento di WordPress.

## Prefetch

Il prefetch del DNS consente di risolvere i nomi di dominio (eseguire una ricerca DNS in background) prima che un utente faccia clic su un collegamento, cosa che può contribuire a migliorare le prestazioni. È implementato aggiungendo un tag `rel="dns-prefetch"` nell'intestazione del vostro sito WordPress.

```
<link rel="dns-prefetch" href="//domain.com">
```

Cose che si è soliti utilizzare per il prefetching del DNS sono l'URL del CDN, i font di Google, Google Analytics, ecc.

```
<link rel="dns-prefetch" href="//cdn.domain.com/">
<link rel="dns-prefetch" href="//fonts.googleapis.com/">
<link rel="dns-prefetch" href="//www.google-analytics.com">
```

Il prefetch è inoltre supportato dalla maggior parte dei browser moderni. Ci sono vari modi per implementarlo.

1. Potete aggiungere il codice al vostro header usando un plugin gratuito come Head, Footer e Post Injections (disponibile su WordPress.org).
2. Potete aggiungere manualmente il codice al file `header.php` del vostro tema.
3. Implementare il prefetch DNS utilizzando un plugin premium come Perfmatters (disponibile su perfmatters.io). Basta fare clic sulla scheda "Extras" nel plugin Perfmatters e aggiungere i domini.  
Formato: `//domain.tld` (uno per riga)

**Perfmatters Settings**

Options | CDN | Google Analytics | **Extras** | License | Support

### Extras

*Extra options that pertain to Perfmatters plugin functionality.*

Script Manager ?

DNS Prefetch ?

## Preconnect

Il preconnect consente al browser di configurare le connessioni prima di una richiesta HTTP, eliminando la latenza di andata e ritorno e risparmiando tempo per gli utenti.

Preconnect è uno strumento importante nella vostra toolbox di ottimizzazione ... può eliminare molti round-trip costosi dal percorso della vostra richiesta, in alcuni casi riducendo la latenza delle richieste di centinaia o persino migliaia di millisecondi.



Ilya Grigorik

È implementato aggiungendo un tag `rel="preconnect"` nell'header di WordPress.

```
<link rel="preconnect" href="//domain.com">
```

Tra le cose che per cui potreste utilizzare preconnect, ricordiamo l'URL del vostro CDN e i Google Font.

```
<link rel="preconnect" href="https://cdn.domain.com">
<link rel="preconnect" href="https://fonts.gstatic.com">
```

Preconnect è supportato dalla maggior parte dei browser moderni, con l'eccezione di Internet Explorer, Safari, Safari per iOS e Opera Mini. Ci sono vari modi per implementarlo.

1. Potete aggiungere il codice al vostro header usando un plugin gratuito come Head, Footer e Post Injections (disponibile su WordPress.org).
2. Potete aggiungere manualmente il codice al file `header.php` del vostro tema.
3. Implementare il prefetch DNS utilizzando un plugin premium come Perfmatters (disponibile su perfmatters.io). Basta fare clic sulla scheda "Extras" nel plugin Perfmatters e aggiungere i domini. Formato: `//domain.tld` (uno per riga)



## Disabilitare gli Script in Base a Post/Pagina

Un altro metodo da considerare per accelerare WordPress parte dall'analisi approfondita di ogni richiesta che viene caricata su pagine e post. Molto probabilmente finirete per trovare script che vengono caricati in tutto il sito, mentre non dovrebbero esserlo.

Potete utilizzare un plugin premium come Perfmatters (disponibile su perfmatters.io), che dispone di una funzione integrata di gestione degli script ("Script Manager"). Questa vi consente di disabilitare gli script (CSS e JavaScript) in base al post e alla pagina, o anche a livello di sito, con un solo clic. Ripetiamo, questo plugin è sviluppato da un membro del team di Kinsta.

Alcuni esempi di cosa può fare:

- Il popolare plugin Contact Form 7 carica su ogni post e su ogni pagina. Potete facilmente disabilitarlo ovunque con un clic e abilitarlo solo sulla vostra pagina di contatto.
- I plugin di social sharing devono essere caricati solo nei vostri post. Potete facilmente disattivarli ovunque e caricarli solo sui post, o anche sui custom post type.
- Il plugin Table of contents (TOC) viene caricato su ogni pagina e post. Con lo script manager potete facilmente stabilire dove deve essere caricato.
- Se avete effettuato l'aggiornamento a WordPress 5.0 e non state utilizzando l'editor a blocchi Gutenberg, evidentemente utilizzate ancora l'editor classico o un editor di terze parti, ci sono altri due script nel front-end che potreste disattivare: `/wp-includes/css/dist/block-library/style.min.css` e `/wp-includes/css/dist/block-library/theme.min.css`.

## Perché Alcuni Plugin Sono Programmati in Questo Modo?

Vi potreste chiedere come mai tutti gli sviluppatori di plugin non fanno in modo che i loro script vengano caricati solo quando il plugin viene rilevato sulla pagina. Beh, è un po' più complicato di quello che sembra. Ad esempio, un plugin come Contact Form 7 fornisce anche degli shortcode che permettono di posizionare i form ovunque. Potreste, ad esempio, inserire un form in un widget. Con WordPress è molto più difficile recuperare i dati quando si disaccodano gli script, rispetto a quanto avviene per le query effettuate dai metadati delle pagine o dei post.

Molte volte è una questione di usabilità. Minore è la possibilità che un plugin vada in errore, meno ticket di supporto ci saranno. Tuttavia, con tutti i plugin disponibili sul mercato, si trova sempre una soluzione per aggirare questo problema e sviluppare il codice in modo da mantenere le prestazioni. Purtroppo spesso accade che l'elevato numero di download e di utenti renda prioritario sviluppare codice per l'usabilità a discapito delle prestazioni.

# Un Tour dello Script Manager

Ora vi offriamo un piccolo tour dello Script Manager. Dopo aver fatto clic nella barra degli strumenti, verranno presentati tutti gli script caricati sull'URL corrente, sia i file JavaScript che CSS. Avete, quindi, le seguenti opzioni:

- 1. Status On** (impostazione predefinita)
- 2. Status Off:** Disable Everywhere (potete scegliere su quali tipi di post volete attivarlo, insieme all'URL corrente)
- 3. Status Off:** Disable only on current URL (questo è molto utile per l'utilizzo sulla vostra home page)
- 4. Status Off:** Exceptions (URL corrente, tipo di post o archivio)

The screenshot displays the 'Script Manager' interface for the 'perfmatters' plugin. The left sidebar contains navigation options: 'Script Manager' (selected), 'Global View', and 'Settings'. The main content area is titled 'Script Manager' with the subtitle 'Manage scripts loading on the current page.' Below this, there are two sections: 'Extended Widget Options' and 'Social Warfare', each with a status toggle (ON/OFF).

**Extended Widget Options** (Status: ON)

Status	Script	Type	Size
ON	<b>jquery-widgetopts</b> /wp-content/plugins/extended-widget-options/assets/js/jquery.widgetopts.min.js	JS	6.1 KB
ON	<b>widgetopts-styles</b> /wp-content/plugins/extended-widget-options/assets/css/widget-options.css	CSS	13.6 KB

**Social Warfare** (Status: OFF)

Status	Script	Type	Size
OFF	<b>social_warfare_script</b> /wp-content/plugins/social-warfare/js/script.min.js	JS	10 KB
Disabled ⊙ Everywhere ⊙ Current URL			
Exceptions <input type="checkbox"/> Current URL			
Post Types: <input checked="" type="checkbox"/> Posts <input type="checkbox"/> Pages <input type="checkbox"/> Lists <input type="checkbox"/> Coupons			
OFF	<b>social_warfare</b> /wp-content/plugins/social-warfare/css/style.min.css	CSS	45.2 KB
Disabled ⊙ Everywhere ⊙ Current URL			
Exceptions <input type="checkbox"/> Current URL			

È tutto **raggruppato in base al plugin o al nome del tema**. Ciò rende semplicissimo disabilitare un intero plugin in una sola volta. In genere un plugin di WordPress avrà sia un file JavaScript che un file CSS. Un tema WordPress potrebbe avere più di 10 file. Lo Script Manager supporta anche le regex nel caso in cui aveste una struttura degli URL o un'installazione più complesse.

Dopo aver selezionato e modificato le impostazioni, ricordatevi di cliccare su “Salva” in basso. Sarà quindi possibile eseguire un test con uno strumento di test della velocità per accertarsi che gli script non vengano più caricati sulla pagina o sul post. Ricordatevi di svuotare prima la cache! E se qualcosa andasse storto sul sito dal punto di vista visuale, potete sempre riattivare il plugin nelle impostazioni per tornare alla normalità.

In uno speed test di woorkup, sono stati in grado di **diminuire i tempi di caricamento complessivi del 20,2%**. Solo nella loro homepage hanno ridotto il numero di richieste HTTP da 46 a 30. Le dimensioni della pagina si sono ridotte da 506,3 KB a 451,6 KB.

## Analisi delle prestazioni di terze parti

Fondamentalmente, tutto ciò che chiamate esternamente dal vostro sito ha degli effetti sul tempo di caricamento. Ciò che rende questo problema ancora più grave è che alcune risorse sono lente solo a intermittenza, rendendo ancora più difficile l'individuazione del problema.

Possiamo considerare un servizio esterno di terze parti come qualsiasi cosa che comunichi con il vostro sito WordPress dal di fuori del vostro server. Ecco alcuni esempi di risorse esterne in cui ci imbattiamo regolarmente:

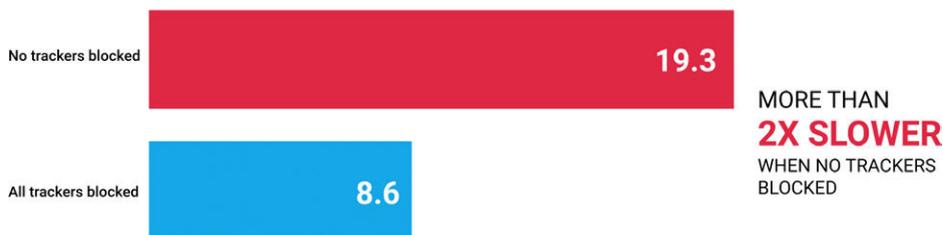
- Piattaforme di social media come Twitter, Facebook e Instagram (widget o pixel di conversione)
- Reti pubblicitarie di terze parti come Google AdSense, Media.net, BuySellAds, Amazon Associates
- Analisi dei siti web e script di monitoraggio come Google Analytics, Crazy Egg, Hotjar, AdRoll

- Strumenti di A/B testing come Optimizely, VWO, Unbounce
- Sistemi di commenti di WordPress come Disqus, Jetpack, Commenti di Facebook
- Backup e strumenti di sicurezza come VaultPress, Sucuri, CodeGuard
- Strumenti di social sharing come SumoMe, HelloBar
- Reti CDN come KeyCDN, Amazon CloudFront, CDN77 e StackPath
- Javascript ospitato esternamente

Quanto influiscono sulle prestazioni questi tracker di terze parti? Nel nostro case study, abbiamo visto che gli script di terze parti avevano **fatto aumentare i tempi di caricamento della pagina dell'86,08%**.

Anche Ghostery ha misurato i primi 500 domini degli Stati Uniti in Alexa, e i risultati sono stati sbalorditivi, anche se, per noi, non sorprendenti. I siti web erano 2 volte più lenti quando nessun tracker era bloccato. Ciò significa che questi script di tracciamento di terze parti sono uno dei principali fattori che contribuiscono a rallentare la velocità di caricamento delle pagine sul web.

## AVERAGE PAGE LOAD TIME WITH vs. WITHOUT TRACKERS

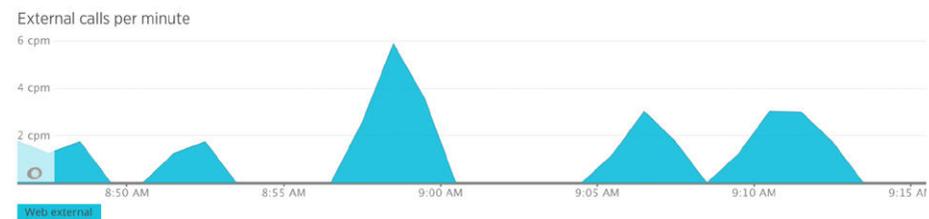
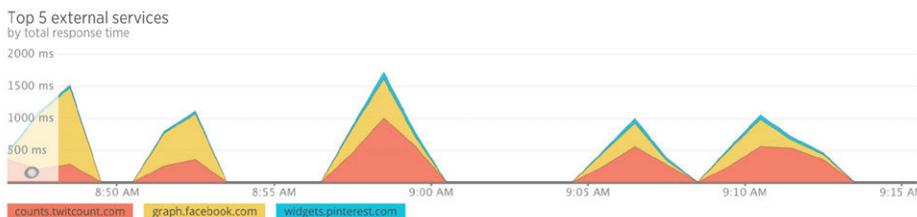


Average page load time (seconds) measured by Ghostery, of the top 500 US domains according to Alexa.

Origine immagine: Ghostery

Dovete stare molto attenti sul vostro sito WordPress. Una sola cattiva chiamata ad un'API di terze parti potrebbe far precipitare il vostro intero sito! Sì, non dovrebbe andare in questo modo, ma in molti casi è proprio così. L'abbiamo visto più volte di quante ne possiamo calcolare.

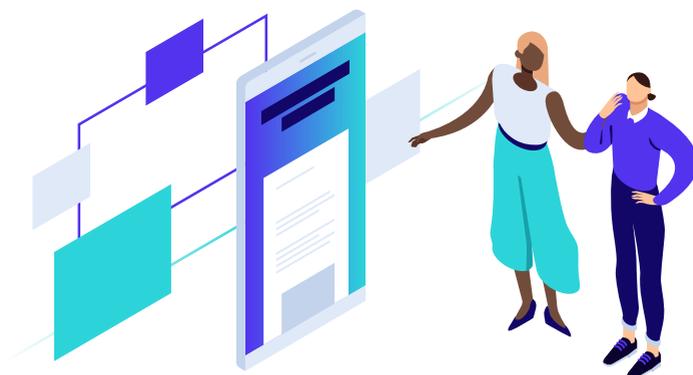
New Relic è uno strumento eccellente per monitorare i servizi esterni nel corso del tempo. Nell'esempio riportato di seguito, possiamo vedere chiamate esterne effettuate su [twitcount.com](https://counts.twitcount.com), [graph.facebook.com](https://graph.facebook.com) e [widgets.pinterest.com](https://widgets.pinterest.com).



È importante che, ogni volta che aggiungete al vostro sito una nuova funzione o un plugin, studiate attentamente le risorse esterne che vengono caricate. Meno ce ne sono, meglio è!

CAPITOLO 16:

# Ottimizzare Sempre con il Mobile-First in Mente



**Google ha iniziato a pubblicare il suo mobile-first index il 26 marzo 2018. In precedenza, i sistemi di scansione, indicizzazione e ranking di Google utilizzavano la versione desktop dei siti web. L'indicizzazione mobile-first indica che Googlebot utilizzerà ora la versione mobile del vostro sito WordPress per l'indicizzazione e il posizionamento. Questo contribuirà a migliorare l'esperienza di ricerca per gli utenti mobili.**

Quando si tratta di ottimizzare il vostro sito per dispositivi mobili, **la velocità è uno dei fattori più importanti su cui concentrarsi.** Questa gioca un ruolo fondamentale in tutto, dalla usabilità alle frequenze di rimbalzo e determina se i potenziali acquirenti torneranno sul vostro sito. La velocità è anche un fattore di landing page per Google Search e Google Ads per le ricerche su dispositivi mobili.

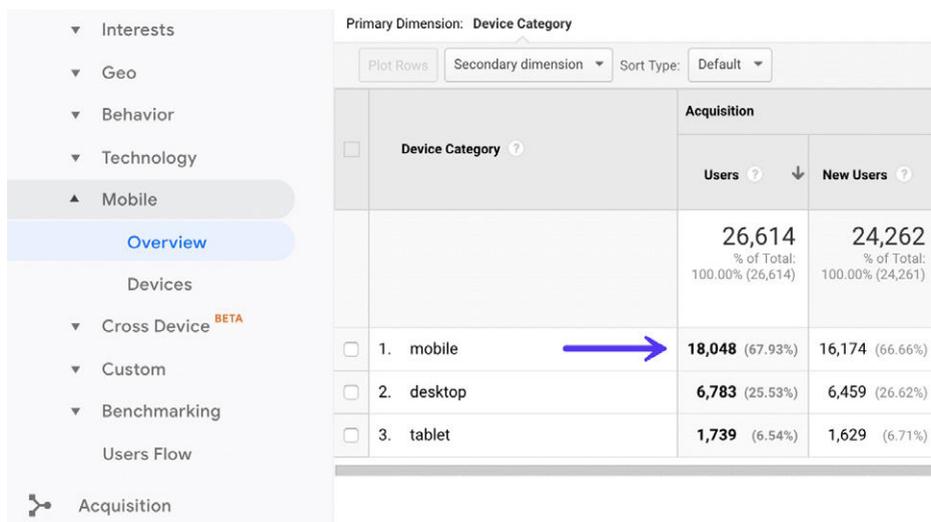
Esperienze mobili negative faranno sì che la maggior parte degli utenti non torni mai più. Secondo l'ultimo rapporto sulla page speed di Google, il tempo medio di caricamento di un sito per dispositivi mobili nel 2018 è stato di 15 secondi. Riuscite a immaginare di aspettare così tanto per caricare una singola pagina? Pazzesco.

Gli utenti chiedono (e meritano) di meglio. Secondo lo stesso rapporto sulla page speed, **il 53% dei visitatori di un sito mobile lascia le pagine che impiegano più di tre miseri secondi per caricare.**

Le esperienze lente su dispositivi mobili non uccidono solo le conversioni. Vi impediscono di avere persino la possibilità di convertire i potenziali clienti. Dato che i tempi di caricamento delle pagine aumentano di pochi secondi, la probabilità che qualcuno rimbalzi aumenta esponenzialmente. Ecco alcune cose da considerare quando si ottimizza per i dispositivi mobili.

## Controllate il Vostro Traffico Mobile

È sempre importante dare un'occhiata alla quantità di traffico mobile che ricevete, in quanto ciò potrebbe spostare un po' le vostre priorità. Potete vedere quanti dispositivi mobili visitano il vostro sito in Google Analytics, alla voce "Pubblico → Dispositivo mobile → Panoramica." Come potete vedere su questo sito, oltre il 67% del traffico proviene da dispositivi mobili. È molto!



Primary Dimension: Device Category

Plot Rows: Secondary dimension Sort Type: Default

Device Category	Acquisition	
	Users	New Users
	26,614 % of Total: 100.00% (26,614)	24,262 % of Total: 100.00% (24,261)
1. mobile	18,048 (67.93%)	16,174 (66.66%)
2. desktop	6,783 (25.53%)	6,459 (26.62%)
3. tablet	1,739 (6.54%)	1,629 (6.71%)

Se siete clienti di Kinsta, potete anche controllare il traffico da dispositivo mobile rispetto al desktop in MyKinsta Analytics. Su questo sito, oltre l'88% del traffico proviene dal desktop. È sempre importante verificare e non solo presumere. Solo perché tutti dicono che le cose vanno verso il mobile, non significa sempre che è vero anche per il vostro sito. Guardate i dati.

## Mobile vs. Desktop

88.9% Desktop 11.1% Mobile



Sat 10/20/10/21 10/22 10/23 10/24 10/25 10/26 10/27 10/28 10/29 10/30 10/31 11/1 11/2 11/3 11/4 11/5 11/6 11/7 11/8 11/9 11/10 11/11 11/12 11/13 11/14 11/15 11/16 11/17 11/18

## Assicuratevi che il Vostro Sito Sia Reattivo

Tested on: Nov 18, 2018 at 11:13 PM

**Page is mobile friendly**

This page is easy to use on a mobile device

Additional resources

- Open site-wide mobile usability report
- Learn more about mobile-friendly pages
- Post comments or questions to our discussion group

SCREENSHOT HTML

Nel 2019, sarà meglio che il vostro sito sia reattivo! Ciò significa che utilizza le media query per ridimensionare gli oggetti automaticamente sui dispositivi mobili. Se non avete ancora provveduto, probabilmente siete già indietro rispetto alla concorrenza. Tutti i temi di WordPress menzionati in precedenza in questo post sono pienamente reattivi e appaiono fantastici su tutti i dispositivi.

Utilizzate lo strumento Mobile-Friendly di Google per effettuare i vostri test e assicuratevi che il vostro sito web soddisfi tutti i requisiti.

## Assicuratevi che srcset Funzioni

In passato era molto importante caricare le immagini in scala e non lasciare che fosse CSS a ridimensionarle. Tuttavia, ora questo non è più così importante dal momento che, a partire da WordPress 4.4, sono supportate le immagini reattive (non ridimensionate da CSS). WordPress crea automaticamente diverse dimensioni di ogni immagine caricata nella libreria multimediale. Includendo le dimensioni disponibili di un'immagine in un attributo srcset, i browser possono ora scegliere di scaricare la dimensione più appropriata e ignorare le altre. Ecco qui sotto un esempio di come appare il vostro codice.

```

```

Con tutti i plugin per immagini di terze parti e le personalizzazioni a disposizione, abbiamo potuto constatare che in molti casi questa funzionalità non opera correttamente. Pertanto, è importante controllare che le immagini ricevano correttamente l'attributo srcset con versioni diverse per dimensioni di schermo diverse. Da ora in avanti, l'ottimizzazione delle immagini sarà sempre importante.

## Google AMP Potrebbe Essere la Soluzione per Voi

Google AMP (Accelerated Mobile Pages Project) è stato lanciato nel mese di ottobre 2015. Il progetto si basa su AMP HTML, un nuovo framework aperto interamente realizzato con tecnologie web esistenti, che consente ai siti web di creare pagine web leggere. Per dirla in modo semplice, offre un modo per pubblicare una versione ridotta della vostra attuale pagina web.

Abbiamo una specie di rapporto di amore e odio con Google AMP, e così succede a molti nella community. Lo abbiamo testato noi stessi e non abbiamo registrato buoni risultati. Tuttavia, questo non significa che non li avrete nemmeno voi. Ogni sito web è diverso e Google AMP viene costantemente migliorato.

Potete iniziare rapidamente ad utilizzare Google AMP sul vostro sito WordPress con uno dei seguenti plugin:

- AMP for WordPress (gratuito, disponibile su WordPress.org)
- AMP for WP (gratuito, disponibile su WordPress.org)

## Riepilogo

Come probabilmente potreste pensare, siamo ossessionati dalla ricerca di tutti i possibili modi in cui si può velocizzare WordPress. Avere un sito veloce aiuta a migliorare il ranking, migliora la possibilità di scansione per i motori di ricerca, migliora i tassi di conversione, aumenta il tempo di permanenza sul sito e diminuisce la frequenza di rimbalzo. Per non parlare del fatto che tutti amano visitare un sito web veloce!

Speriamo che questa guida alla velocità sia stata utile e che riusciate a portare a casa alcune soluzioni e ad applicarle al vostro sito WordPress.

Non dimenticate di dare un'occhiata al nostro programma di affiliazione dell'hosting WordPress all'indirizzo <https://kinsta.com/affiliates>. Potete guadagnare fino a \$500 per ogni referral + 10% di commissioni periodiche mensili. È la migliore proposta di affiliazione che troverete nel settore. Acceleriamo il web insieme!

Per maggiori informazioni su Kinsta, visitare il sito web <https://kinsta.com/it/>.

**Grazie per la lettura!**



**Note**













La velocità del vostro sito WordPress è più importante di quanto potreste pensare. Non importa quale tipo di sito abbiate in esecuzione; sia che si tratti di un blog che di un negozio WooCommerce, i tempi di caricamento della pagina sono sempre direttamente correlati a:

- Miglior ranking delle parole chiave (la velocità è uno dei fattori utilizzati da Google).
- Aumento dei tassi di conversione.
- Miglioramento della capacità di analisi da parte dei motori di ricerca come Google e Bing.
- Tempo medio di permanenza sul sito superiore.
- Diminuzione delle frequenze di rimbalzo.

Per non parlare del fatto che avere un sito web che carica velocemente offre un'esperienza complessiva superiore ai i vostri visitatori.

WordPress ora alimenta oltre un terzo di tutti i siti web su Internet! È fantastico, ma significa anche che ci sono migliaia di temi, plugin e tecnologie diverse che devono coesistere. Per l'utente medio di WordPress, tutto questo può trasformarsi rapidamente in un incubo quando su un sito iniziano a presentarsi delle strozzature. Molti non sanno il perché o, addirittura, non sanno nemmeno da dove iniziare a cercare per risolvere il problema.

Il team di Kinsta si occupa quotidianamente di questi tipi di problemi e ha visto di tutto! In questa guida ci immergeremo in profondità tra le soluzioni utili a velocizzare i siti basati su WordPress. C'è di tutto, dai consigli per principianti alle strategie avanzate, e in tutto questo troverete di certo qualcosa di utile. Questa guida sintetizza oltre 15 anni di esperienza, e tutte le dure lezioni che abbiamo appreso.

**KINSTA**