

WebbMotell - ISPConfig

Vilken cache-plugin för WordPress skall jag använda?

Våra web-servrar erbjuder inbyggd server-side caching som en del av vår infrastruktur. Då är det rimligt att fråga sig om ett separat cache-plugin verkligen behövs.

Det finns webbhotell som menar att cache-plugins är överflödiga eftersom servern redan hanterar cachning. Men räcker det verkligen i praktiken?

Vad är server-side cache?

När en förfrågan når en server måste WordPress köra en mängd PHP-skript och MySQL-frågor.

Varje gång du öppnar en sida eller ett inlägg måste minst 27 MySQL-frågor köras. 27 är standardantalet frågor – men det kan vara 50 eller 100 beroende på vilket tema du använder och hur många plugins som är installerade. Varje fråga kan ta några millisekunder att bearbeta, beroende på frågans komplexitet och mängden data i databasen.

När man summerar alla dessa PHP- och MySQL-exekveringar kan TTFB (Time to First Byte) landa på runt en sekund – eller till och med mer.

Server-side caching eliminerar denna process genom att lagra kopior av det genererade HTML-innehållet på disk eller i minnet. Detta minskar TTFB och minskar belastningen på serverresurserna, eftersom servern endast behöver leverera den redan genererade filen.

Våra web-servrar erbjuder APCu/Redis/Memcached/Memcache och dessutom OPcache, vilket vi kommer till senare.

Om du cachar HTML-sidor via Cloudflare räknas det tekniskt sett också som en form av server-side caching.

Vad gör cache-plugins som server-side caching inte klarar av?

De flesta cache-plugins genererar statiska HTML-sidor på liknande sätt som server-side caching. Men cache-plugins erbjuder dessutom flera front-end-optimeringar som servern i sig inte hanterar.

Här är några exempel:

WebbMotell - ISPConfig

- **Generering av critical path CSS** – Förbättrar FCP (First Contentful Paint), vilket numera är en rankingfaktor i Googles sökresultat.
- **Fördröjd inläsning av JavaScript och CSS** – Elimineras render-blocking resurser genom att ladda dem asynkront.
- **Minifiering av CSS och JavaScript** – Minskar filstorleken och därmed laddningstiden.
- **Lazy loading** – Fördröjer inläsning av bilder, videor, inbäddat innehåll m.m. tills de syns i vyporten.

Det finns många fler funktioner beroende på vilket cache-plugin du använder.

Så, behöver man ett cache-plugin även om server-side caching redan finns? Svaret är: JA!

Server-side caching är utmärkt för att avlasta servern och minska TTFB, men det räcker inte hela vägen när det gäller prestandaoptimering ur ett front-end-perspektiv. Ett bra cache-plugin kan göra skillnad när det gäller användarupplevelse och Core Web Vitals.

Jag rekommenderar **FlyingPress** för cachning, då det kombinerar kraftfull sidcachning med avancerade optimeringsfunktioner som lazy loading, preloading, minifiering, critical CSS och mycket mer – allt med låg påverkan på serverresurserna.

Om du redan har server-side caching via ditt webbhotell – grymt. Men komplettera det med ett välkonfigurerat cache-plugin för att maximera både hastighet och användarupplevelse.

Notera också att det behövs plugins för att aktivera server-side cachning för hemsidan, dock med ett undantag.

OPCache

Läs mer om OPCache i denna artikel [Vad är OPCache?](#). Denna är aktiverad på våra webbservrar och behöver ingen plugin för att den skall användas.

Jämförelse mellan olika plugins

Här kommer en tekniskt inriktad jämförelse mellan **FlyingPress**, **WP Rocket**, **Surge** och **W3 Total Cache**, med fokus på prestanda, funktionalitet och användarvänlighet:

⚙ 1. FlyingPress

+ Fördelar:

WebbMotell - ISPConfig

- Allt-i-ett-lösning med modern kodbas och hög prestanda.
- Generering av **Critical CSS** (per sida), **lazy loading, preloading, minifiering, defer av JS/CSS**.
- Inbyggt stöd för **font-display: swap, image optimization** och **cache invalidation**.
- Extremt lätt att konfigurera – kräver minimalt med teknisk kunskap.
- Fungerar väl tillsammans med server-side caching och CDN som Cloudflare eller Bunny.net.

- Nackdelar:

- Premiumplugin – ingen gratisversion.
- Inte lika modulärt som W3TC om du vill ha granular kontroll.

Passar bäst för: Prestandafokuserade WordPress-sajter som vill ha kraftfull optimering utan krångel.

⚙ 2. WP Rocket

+ Fördelar:

- Välkänt, moget plugin med stort community.
- Enkel att använda – bra UI.
- Bra stöd för cache, preloading, minifiering, delay JS execution, m.m.
- Integrerar fint med CDN, inklusive Cloudflare.

- Nackdelar:

- Premium – ingen gratisversion.
- Saknar vissa avancerade optimeringsfunktioner som bildoptimering (kräver tillägg).
- Critical CSS är globalt, inte per sida (inte lika exakt som FlyingPress).

Passar bäst för: Användare som vill ha en välpolerad, stabil lösning och som inte har extremt avancerade behov.

WebbMotell - ISPConfig

⚙ 3. Surge

⚠ Endast relevant för statiska WordPress-sajter.

+ Fördelar:

- Superlätt plugin, utvecklat av WordPress core contributors.
- Ger extremt låg TTFB via inbyggd full page static cache direkt i WordPress.
- Kräver nästan ingen konfiguration – installera och kör.

- Nackdelar:

- Ingen CSS/JS-optimering, inget stöd för lazy loading, preloading eller liknande.
- Ingen kontroll över cache-uppdatering.
- Inte lämpligt för dynamiska sajter med inloggade användare eller WooCommerce.

Passar bäst för: Bloggar eller statiska innehållssajter där man prioriterar snabbhet med minimal overhead.

⚙ 4. W3 Total Cache (W3TC)

+ Fördelar:

- Extremt konfigurerbart – stödjer object cache, page cache, browser cache, CDN, fragment caching, etc.
- Kan integreras med Redis, Memcached, APCu.
- Stöd för både disk och RAM-baserad cache.
- Har en bra guide för vilka inställningar som är bäst för just denna site.
- Gratis

- Nackdelar:

WebbMotell - ISPConfig

- Många funktioner överlappar med vad hosting-leverantörer redan hanterar (t.ex. via server-side caching).
- Frontend-optimeringar (CSS/JS minifiering etc.) är inte lika kraftfulla eller moderna som i FlyingPress eller WP Rocket.

Passar bäst för: Den som inte vill betala för typ **FlyingPress** får här en plugin som hanterar alla aspekter på cachning. Behövs inga flera plugins om man väljer denna. Passar både den teknisk avancerade liksom nybörjaren.

Sammanfattning

Plugin	Enkelhet	Prestanda	Funktioner	Bäst för
FlyingPress	□□□□	□□□□□	□□□□□	Högprestanda, enkel användning
WP Rocket	□□□□	□□□□	□□□□	Stabil allroundlösning
Surge	□□□□□	□□□□□	□	Statiska sajter
W3 Total Cache	□□□□□	□□□□	□□□□□	Komplett, granular kontroll, gratis

Vilken plugin skall jag använda för att komplettera flyingpress så att jag får object cache också?

Bra fråga! FlyingPress hanterar **frontend-optimering** (HTML, CSS, JS, bilder, lazy loading osv), men **object cache** – alltså caching av databasfrågor och PHP-objekt – kräver en separat lösning.

Här är de **bästa alternativen för object cache** som fungerar perfekt **tillsammans med FlyingPress**:

1. [Object Cache Pro](#) (premium)

WebbMotell - ISPConfig

Rekommenderas starkt för Redis-baserad object cache.

- **Byggt för Redis** – supersnabb, stabil och har stöd för taggar, preloading, metrics m.m.
- Integreras väl med WooCommerce, Multisite och andra tunga WordPress-installationer.
- Ger **debugging & statistik via WordPress admin**.
- Kräver Redis-server (kan köras lokalt eller via t.ex. Redis på Docker).

2. [Redis Object Cache](#) (gratis)

Open source, officiellt plugin från teamet bakom Redis Object Cache.

- Enkel att installera och konfigurera.
- Klarar tunga sajter bra, men saknar avancerad funktionalitet (ingen statistik eller taggar).
- Kompatibel med FlyingPress, WooCommerce och Multisite.

Sammanfattning

Plugin	Pris	Kräver Redis	Stöd för statistik	Prestanda
Object Cache Pro	\$\$\$	☐	☐ (admin + graf)	☐☐☐☐
Redis Object Cache	Gratis	☐	☐	☐☐☐

Vår rekommendation

Är ni beredd att betala då är [Flying Press](#) absolut bäst. Utöver denna behöver man en plugin för **Object Cache** enligt ovan. Vilken är en fråga för plånboken.

Notera att W3TC inte kan användas i kombination med FlyingPress, eftersom de båda försöker göra frontend-optimering – det blir konflikt.

Annars är gratis-versionen av [W3 Total Cache](#) det bästa alternativet. Det är komplett och enkelt att ställa in med sin interaktiva **Setup Guide**. Välj den metod, ingen/disk/redis/memcached/APC, som W3TC rekommenderar när ni kör **Setup**

Sida 6 / 7

WebbMotell - ISPConfig

Guide.

Notera dock att cachning kan ställa till det, t.ex. vid staging. Det är därför bra att försöka minimera användningen av cache. Ser man att det bara ger en minimal förbättring på några procent, då är det bättre att låta det vara ocachat.

Referenser

- [best-wordpress-cache-plugins/](#)
- [Do I need a Cache Plugin with Server Side Cache \(FastCGI\)?](#)
-

Unikt lösnings-ID: #1060

Av: : Admin

Senast uppdaterad: 2025-03-23 11:27