



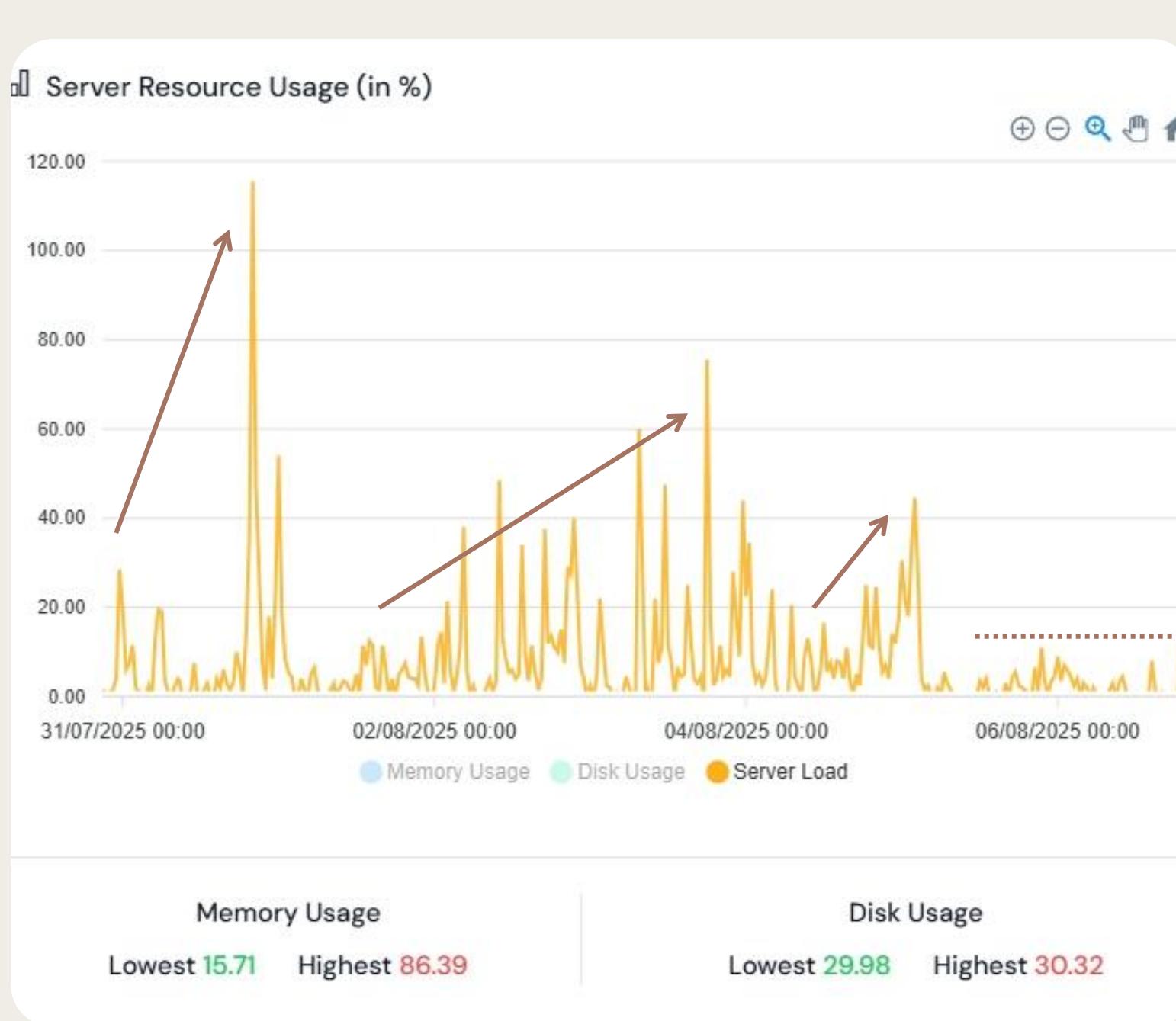
小主機也能飛： WordPress × Cloudflare 實戰架構與除錯心法

WordCamp Taiwan 2025 | Steven Tseng



Agenda

- 01 我的網站怎麼又慢了？
- 02 三層快取的分工合作
- 03 別讓快取被「好心」的規則拖慢
- 04 怎麼知道快取有沒有幫上忙？
- 05 同一套思路，解更多問題
- 06 網站最佳化不是一次性的事



我的網站怎麼又慢了？

情境

- 後台發文要等半天
- 頁面載入速度變慢
- 主機資源耗用顯著提升

WHY

- 安裝了快取外掛，速度也沒改善？
- 快取命中率 (HIT) 大幅降低？

三層快取的分工合作 - 用餐廳備餐來比喻

桌邊保溫盒 (瀏覽器快取) - 就在客人面前，伸手就有

適合放「常見的小菜」：CSS、JS、Logo、已看過的圖片與頁面

特色：最快、容量最小、由瀏覽器的 Cache-Control 規則決定能放多久

外場出餐檯 (CDN/Cloudflare 邊緣快取) - 開在客人家附近的分店，離客人最近

適合放「熱門主菜」：首頁、文章頁、列表頁、靜態資源

特色：速度快、覆蓋廣，可用 TTL/規則控管；缺菜時才回廚房補貨

廚房內場 (主機端/WordPress 快取)

- 「成品」 = 頁面快取 (HTML 靜態化 / FastCGI / Disk cache)
- 「切好料」 = 物件快取 (Redis/Memcached，幫 PHP/DB 減少存取與計算)

特色：比「現點現做」快非常多；若連成品都沒有，才會「照食譜」讓 PHP + SQL 現做

三層快取的分工合作 - 用餐廳備餐來比喻

桌邊保溫盒 (瀏覽器快取)

客人點一道菜

先檢查桌邊保溫盒

外場出餐檯 (CDN/Cloudflare 邊緣快取)

去外場出餐檯取餐

1. 並在桌邊保溫盒保留

廚房內場 (主機端/WordPress 快取)

回中央廚房取餐/備餐

1. 有現成品就直接裝盤

2. 沒有成品，主廚按照食譜現做餐點

做完同時分裝，補內場、補外場、桌邊也留一份

使用者發出請求

瀏覽器有現成就「秒上」

從 CDN 摷取

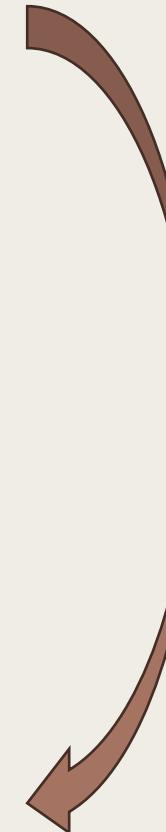
1. 並保留一份在瀏覽器快取

從頁面外取擷取

1. Cache HIT → 摷取成功

2. Cache Not HIT → 存取主機

完成後 → 頁面快取/CDN/瀏覽器快取



三層快取的分工合作 - 用餐廳備餐來比喻

桌邊保溫盒 (瀏覽器快取)

客人點一道菜

先檢查桌邊保溫盒

外場出餐檯 (CDN/Cloudflare 邊緣快取)

去外場出餐檯取餐

1. 並在桌邊保溫盒保留

廚房內場 (主機端/WordPress 快取)

回中央廚房取餐/備餐

1. 有現成品就直接裝盤

2. 沒有成品，主廚按照食譜現做餐點

做完同時分裝，補內場、補外場、桌邊也留一份

使用者發出請求

瀏覽器有現成就「秒上」

從 CDN 摷取

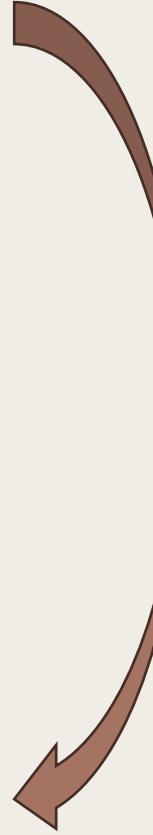
1. 並保留一份在瀏覽器快取

從頁面外取擷取

1. Cache HIT → 摷取成功

2. Cache Not HIT → **存取主機**

完成後 → 頁面快取/CDN/瀏覽器快取



三層快取的分工合作 - 用餐廳備餐來比喻

命中越前面成本越低

→ 省傳輸、省 CPU/DB 資源

→ 小主機也能幹大事

客人點一道菜
先檢查桌邊保溫盒
去外場出餐檯取餐
1. 並在桌邊保溫盒保留
回中央廚房取餐/備餐
1. 有現成品就直接裝盤
2. 沒有成品，主廚按照食譜現做餐點
做完同時分裝，補內場、補外場、桌邊也留一份

使用者發出請求

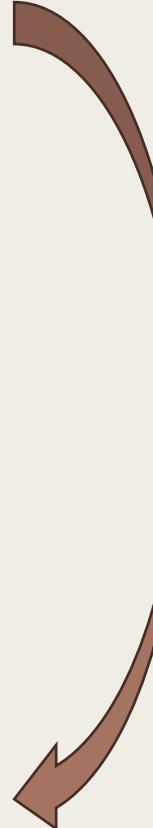
瀏覽器有現成就「秒上」

從 CDN 摷取

1. 並保留一份在瀏覽器快取

從頁面外取擷取

1. Cache HIT → 摷取成功
2. Cache Not HIT → 存取主機
完成後 → 頁面快取/CDN/瀏覽器快取



別讓快取被「好心」的規則拖慢

很多快取外掛會自動幫你排除登入、購物車、語言切換、參數...

這些看起來很好心，但有時太過好心反而讓網站變慢、故障。

真正的最佳化，是懂得哪些該排除、哪些可以快取，把資源用在刀口上。

別讓快取被「好心」的規則拖慢

哪些菜「不適合久放」(應該避開快取)

- 個人化與動態內容：
購物車、結帳、會員中心、表單提交 (含驗證碼/Token)、後台頁面、API 回應 (端點路徑)、站內搜尋等。
- Query Loop 參數：
WordPress 核心/各外掛都有，如 SureCart 商品子頁面/結帳頁面、TutorLMS 課程類別篩選、講師頁面等。

哪些能快就快？(免得廣告一下，流量全部擊中主機)

- 追蹤參數 (行銷/廣告類)：
Google、Facebook、X(Twitter)、Microsoft、Mailchimp、IG、美安、UTM、LINE 等。

別讓快取被「好心」的規則拖慢



快取等級

決定 Cloudflare 應該快取您的網站上的多少靜態內容。增加快取可加速網頁載入。

此設定上次變更是在 幾秒前

Cloudflare 的 [快取等級](#) 是什麼？

您可以設定 Cloudflare 的 CDN 將 [靜態內容](#) 依下列等級加入快取：

- 無查詢字串：僅在無查詢字串時從快取傳遞檔案。
- 忽略查詢字串：無論查詢字串，向所有人都傳遞相同資源。
- 標準：依查詢字串變化，每次供應不同資源。

推薦使用標準等級。[進一步瞭解快取等級](#)。

備註：Cloudflare 預設並不快取 HTML 內容。若有需要，可建立 Cache Rules 來 [快取靜態 HTML 內容](#)。

API ▶ 說明 ▾

○ 無查詢字串
○ 忽略查詢字串 (選中)
● 標準

誤區 (Cloudflare)

- 靜態內容 → 忽略查詢字串
 - 導致外掛改版後資源不更新
 - *.css?ver=1 與 *.css?ver=2 被視為同一份

建議

- 用快取規則對 HTML 忽略查詢字串
- 靜態內容保留「依查詢字串變化，每次供應不同資源」的標準方法

別讓快取被「好心」的規則拖慢

Skip Caching for These Cookies

Pages won't be cached when these cookie patterns are detected. One pattern per line. Supports regex patterns (uses preg_grep for matching).

```
^wordpress_logged_in_
^wp-  
^comment_
^woocommerce_
```

誤區 (Super Page Cache)

- not http.cookie contains “wp-”
 - 很多外掛、佈景主題都會有這類 cookie (即使未登入)
 - 例如 wp-dark-mode、wp-settings-* 等
 - 只要命中，就會略過快取

建議

- 用精準的登入/購物/評論/個人化 cookie
 - 例如 wordpress_sec_*、wordpressuser_*、edd_items_in_cart 等

忽略查詢字串
不受查詢字串影響，向所有人都傳遞相同資源

排序查詢字串
將具有相同查詢參數的請求視為相同，無論這些查詢參數的順序如何。

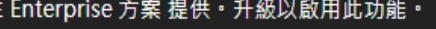
查詢字串 (選用)

所有查詢字串參數

所有查詢字串參數  但下列除外:

無查詢參數，但下列除外:

忽略查詢字串

標頭  僅在 Enterprise 方案 提供。升級以啟用此功能。
定義哪些標頭將注入快取索引鍵 [進一步瞭解](#) 

包括標頭和所選值
符合標頭中的所有值，在快取索引鍵中僅包括符合的值。對於受限制  標頭，您需要包括 1 至 10 個值。

Ignore marketing parameters 

Significantly increases cache hit rate by ignoring common marketing and tracking parameters in URLs, treating them as the same page for caching purposes.

別讓快取被「好心」的規則拖慢

忽略行銷/廣告類參數容易嗎？

- 有付費就容易
 - Cloudflare Enterprise 方案
(Kinsta / FlyingCDN 等)
 - Super Page Cache 外掛 Pro 方案

不花錢的方法？

- 繞過外掛的限制，自己設定快取規則 (Cache Rules)
- Cloudflare Workers 修改快取標頭
- Cronjob 快取預熱

別讓快取被「好心」的規則拖慢

繞過外掛的限制，自己設定快取規則 (Cache Rules)

- Cloudflare 快取規則功能強大
- Super Page Cache 外掛降低進入障礙

可以大幅改善行銷/廣告類參數的快取 MISS 問題

但無法解決偶爾發生的略過快取問題：

x-wp-cf-super-cache-disabled-reason = Bypass Query Var

The screenshot shows the 'Cache Rules' section of the WP Super Page Cache settings. At the top, there is a toggle switch for '忽略查詢字串' (Ignore query string) which is turned on (green). A red arrow points to this switch. Below it is a section for '排序查詢字串' (Sort query string) which is turned off (grey). The '查詢字串 (選用)' (Query string (optional)) section contains four radio button options: '所有查詢字串參數' (All query string parameters), '所有查詢字串參數, 但下列除外:' (All query string parameters, except for the following:), '無查詢參數, 但下列除外:' (No query parameters, except for the following:), and '忽略查詢字串' (Ignore query string). The '忽略查詢字串' option is selected (radio button is filled). A tooltip at the bottom of this section states: '僅在 Enterprise 方案 提供。升級以啟用此功能。' (Only available in the Enterprise plan. Upgrade to enable this feature.).

順序	名稱	已啟用
1	SPCFC Cache Rule → Cache Eligible Requests (主機名稱, Cookie, URI, URI 路徑)	<input checked="" type="checkbox"/>
2	[DO NOT EDIT] WP Super Page Cache Plugin rules for (主機名稱, Cookie, URI, URI 路徑)	<input type="checkbox"/>

別讓快取被「好心」的規則拖慢

運用 Cloudflare Workers 修改快取標頭

- Cloudflare 邊緣「忽略查詢字串」vs SPC 在原點的判斷不一致
- 在打到原點之前先清理掉追蹤參數
 - 不影響功能參數 (如 ?lang=、?variant=、?products)
 - 不改使用者網址、不轉址，GA/Pixel 能正常運作
 - 以乾淨 URL 當 cacheKey

1. 只處理 text/html

2. 清除追蹤參數

```
[ "utm_source", "utm_medium", "utm_campaign", "utm_term", "utm_content", "utm_id", "utm_source_platform", "utm_creative_format", "utm_marketing_tactic", "gclid", "dclid", "msclkid", "fbclid", "wbraid", "gbraids", "igshid", "mc_eid", "mc_cid", "openExternalBrowser", "twclid", "twclkid", "gad_source" ];
```

情境：訪客從 FB 點連結過來 (帶 fbclid)

1. 請求進入 Cloudflare → 先進 Worker
 - Worker 偵測到 fbclid，建立「乾淨 URL」(原網址移除追蹤參數)
 - 以 cacheKey = 乾淨 URL 發出 fetch (前端網址仍含 fbclid)
2. 邊緣快取查找 (以 cacheKey)
 - HIT : Cloudflare 直接回應，不打原點
 - MISS : Cloudflare 回源一次取得頁面，並以 cacheKey 寫入邊緣快取
3. 後續所有訪客 (帶參數或不帶參數)，因 cacheKey 一致，統一命中一份邊緣快取、HIT 率提升

怎麼知道快取有沒有幫上忙？

X-Wp-Cf-Super-Cache

- cache – SPC 使用中，頁面快取已啟用
- no-cache – SPC 使用中，頁面快取已停用

X-Wp-Cf-Super-Cache	no-cache
X-Wp-Cf-Super-Cache-Cache-Control	no-store, no-cache, must-revalidate, max-age=0
X-Wp-Cf-Super-Cache-Disabled-Reason	Logged In
X-Wp-Spc-Disk-Cache	BYPASS

X-Wp-Cf-Super-Cache-Active

- 1 – 頁面已存放在 Cloudflare 快取中

X-Wp-Cf-Super-Cache-Active	1
X-Wp-Cf-Super-Cache-Cache-Control	s-maxage=31536000, max-age=60
X-Wp-Spc-Disk-Cache	DISABLED

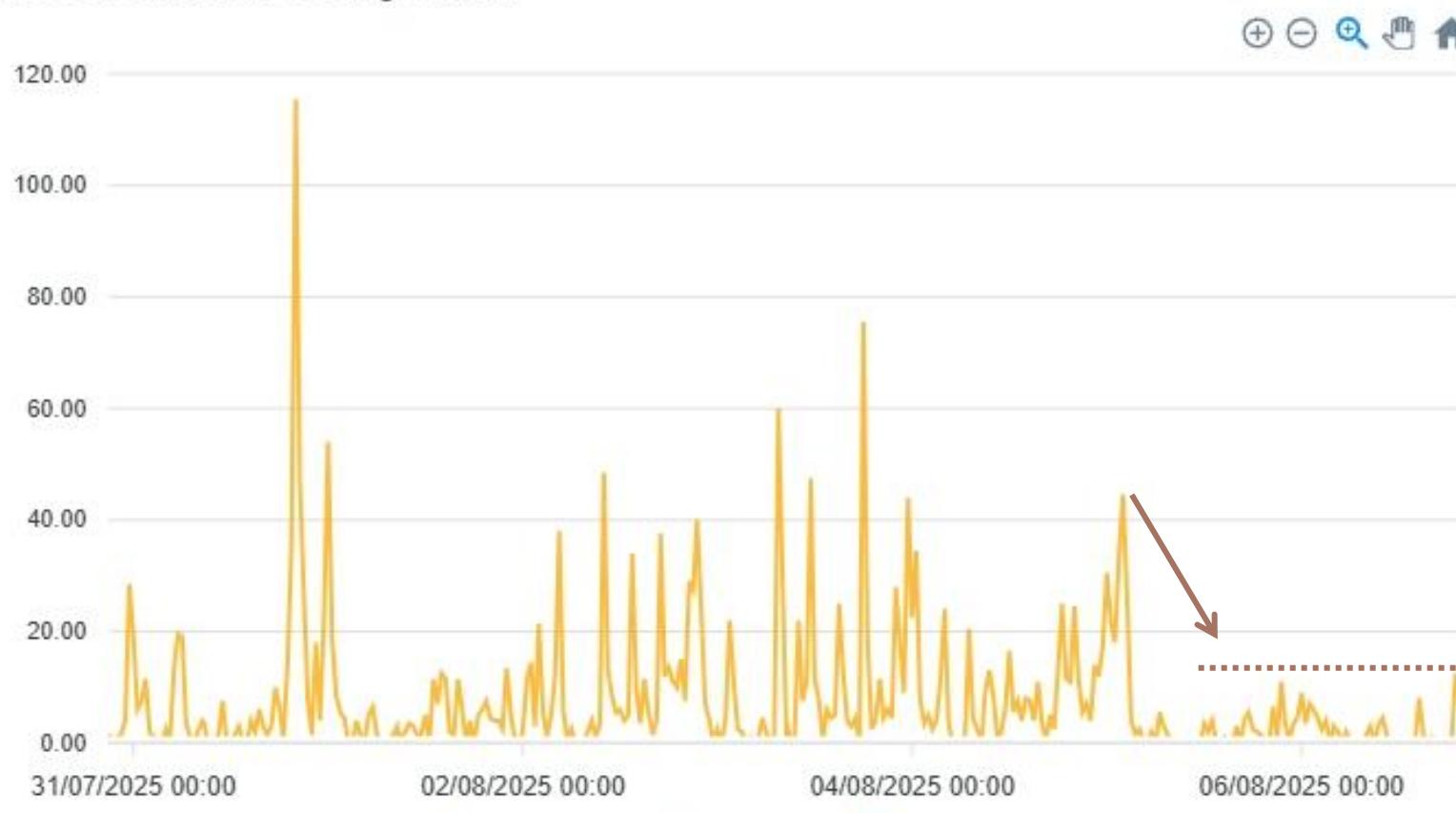
Cf-Cache-Status

- HIT – 該頁面已從快取中傳回
- MISS – 快取中未找到該頁面，請刷新頁面
- BYPASS – 該頁面被 SPC 排除
- EXPIRED – 已被緩存，但快取已過期
- DYNAMIC – 頁面中有動態內容，不使用快取

▼ Response Headers	
Age	23
Alt-Svc	h3=":443"; ma=86400
Cache-Control	s-maxage=31536000, max-age=60
Cf-Cache-Status	HIT

▼ Response Headers	
Alt-Svc	h3=":443"; ma=86400
Cache-Control	s-maxage=31536000, max-age=60
Cf-Cache-Status	DYNAMIC

■ Server Resource Usage (in %)



怎麼知道快取有 沒有幫上忙？

情境

- 後台發文不再卡卡
- 頁面載入速度變快
- 主機資源耗用穩定在一個範圍內

CHECK

- 快取命中率 (HIT) 大幅提高

總計

7.85k

● 由 Cloudflare 緩解

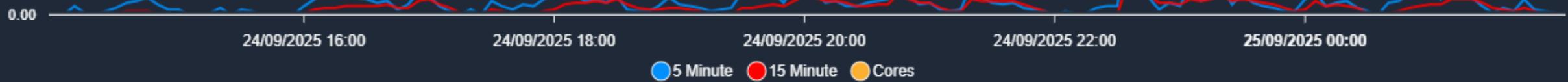
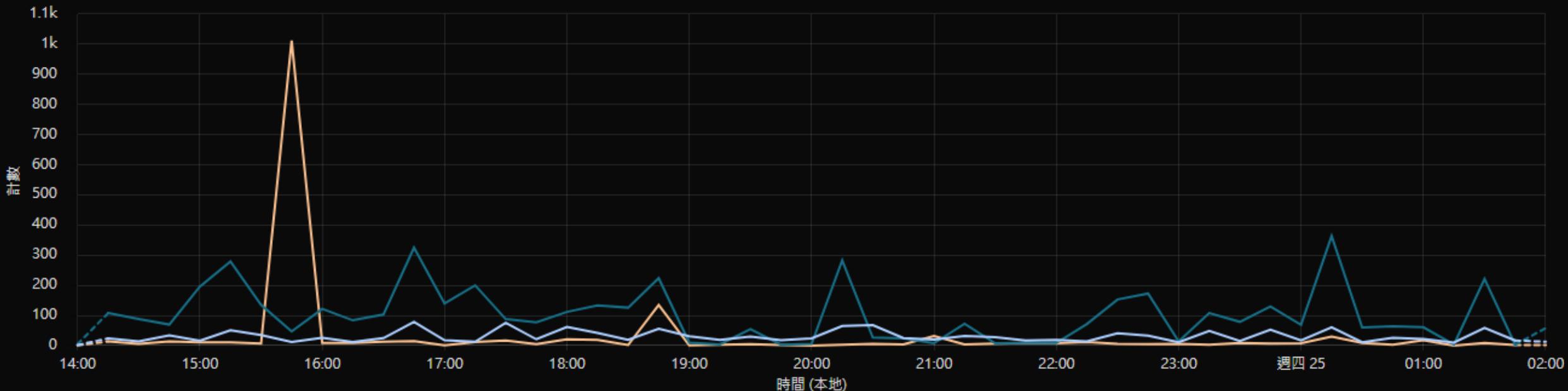
1.55k

● Cloudflare 伺服器提供

4.81k

● 原點伺服器提供

1.49k



同一套思路，解更多問題

錯一個設定，不是混語系就是卡結帳

怎麼做？

- Cloudflare 快取規則
 - 列出需繞過的 Cookie
→ tutor_、comment_、wlmapi、wishlist_reg 等
 - 列出需繞過的頁面
→ cart、checkout、account、dashboard 等
 - 列出需繞過的動態更新頁面 (Query Loop)
→ ?s=、?keyword=、?look_content_only=、?products= 等
 - 列出需繞過的端點
→ ?fluent、?ns_url= 等
- Cloudflare Worker
 - 列出不能繞過快取的追蹤參數

網站最佳化不是一次性的事

- 快取是有限資源下的最佳幫手
- 規則與檢查方法要並行
- 檢查、調整、測試是循環不斷的過程

什麼時候應該重新檢視？

- 內容變了
- 外掛更新
- 流量模式變

例行性檢查

- 每月用無痕視窗測一次 HIT 率、尖峰時段抽測
- 新增大型功能或改版後，跑一遍「檢查 → 調整 → 測試」

Q&A

把你今天的發現回饋給外掛作者或社群。工具會因此變好，你的網站也會更穩。

這就是 **WordPress** 生態圈最有價值的地方。