



WP-CRON, CZYLI JAK PORADZIĆ SOBIE Z ZADANIAMI WORDPRESS'A

Posted on 22 kwietnia 2014 by Maciej Cybulski

WordPress ma swój harmonogram zadań WP-Cron, który jednak często nie jest w 100% wydajnym i zadowalającym rozwiązaniem. Jak zatem to poprawić? Z pomocą może przyjść nam cron systemowy...



Categories: [aktualności](#), [instalacja](#), [porady](#), [programowanie](#), [tuning](#)

W WordPressie standardowo zaimplementowano funkcjonalność harmonogramu, który nazywany jest też cronem, co nawiązuje do harmonogramu zadań w systemach unixopochodnych. Działa on na tej samej zasadzie jak np. harmonogram zadań w MS Windows. Co do zasady cron daje nam możliwość uruchomienia jakiegoś zadania o określonej porze, lub co określony interwał czasu. Dzięki temu np. możemy zaplanować publikację wpisu, pobieranie tweetów itp. No właśnie... zasada jest prosta, natomiast niedogodności mogą zacząć się przy analizowaniu szczegółów.

WP-Cron - czyli nie ma róży bez kolców

WordPressowy cron wykorzystywany jest przede wszystkim do sprawdzania aktualizacji wtyczek, motywów czy samego WordPressa, ale też co publikowania postów, pingbacków. Również niektóre wtyczki, które wykonują określone zadania cyklicznie, wykorzystują wp-crona do tego celu.

WP-Cron działa poprawnie, czyli tak jak można by tego od niego oczekiwać, jedynie na stronach które są często odwiedzane przez Internautów. Pomysłodawcy i programiści wordpressowego crona, aby uprościć podstawową konfigurację WordPressa, postanowili że harmonogram zadań WP będzie wywoływany za każdym razem gdy ktoś będzie odwiedzał stronę.

Co to znaczy dla nas?

Ano jeśli np. mamy na dobę powiedzmy dla prostszego rachunku 6 odwiedzin, czyli średnio co cztery godziny (24/6) to znaczy że uruchomienie danego zadania może opóźnić się nam średnio o te 4h. Powiedzmy że zaplanowaliśmy publikację wpisu o 10:00 a wejście na stronę było o 9:50 i później o godz 12:00. Efektem czego jest to że wpis pojawił się na stronie dopiero po pierwszym wejściu internauty po planowanej godzinie publikacji czyli... zamiast o 10:00 wpis pojawił się o 12:00.

Stąd właściwie wp-cron jest takim pseudo harmonogramem zadań. Bo jeśli np. przez dobę nikt nie zajrzy na naszą stronę to zaplanowane zadania będą miały opóźnienie przynajmniej o tę dobę. Co zatem z wtyczkami realizującymi publikację wpisów, archiwizację czy choćby publikowanie informacji na socialmediach?

Owszem wykorzystanie wp-crona upraszcza znacznie konfigurację środowiska pracy WordPress'a, jednak właśnie kosztem pewnej losowości w jego działaniu.

Można zaradzić tym bolączkom poprzez odłączenie sterowania wp-cronem od jądra WordPress'a i "przypięcie" wyzwalania wordpressowego crona do harmonogramu zadań na serwerze (crontab). Brzmi skomplikowanie, jednak w dalszej części będę się starał pokazać że nie jest to aż tak bardzo skomplikowane jak brzmi:)

Korzyści z crona systemowego

Co tak naprawdę daje nam podpięcie wp-crona do harmonogramu zadań systemu w którym pracuje nasza strona?

1. Daje nam pewność że o konkretnej porze dane zadanie będzie wykonane. Choćby publikacje

wpisów, czy znacznie ważniejsze np. wykonanie backupu bazy czy całej strony. Niezależnie od tego czy ktoś na stronę wejdzie czy nie zadanie się wykona o zadanej porze.

2. W standardowej konfiguracji WordPress'a każde otwarcie strony wyzwala wp-crona i zadania do niego wpisane., co może powodować chwilowe zwiększenie obciążenia serwera i wolniejsze ładowanie stron, szczególnie gdy w harmonogramie znajduje się zadanie które wykonuje się długo (np. archiwizacja wykorzystująca wysyłanie archiwum np. na serwery Amazona).

Co zatem musimy zrobić? - Optymalizacja WP-Crona

Od czego zacząć? Na początek powinniśmy wyłączyć standardowe działanie wp-crona. Najprostszym sposobem jest umieszczenie w pliku wp-config.php poniższego kodu

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

Wyłącza on całkowicie wyzwalanie wordpressowego cron, czyli nie będzie już uruchamiania cron po każdym wejściu internauty na stronę.

Plik wp-config.php znajdziemy w głównym katalogu naszej strony na serwerze.

Następnym etapem będzie "podpięcie" naszego wp-crona pod systemowy harmonogram zadań czyli crontaba. W tym celu najlepiej przyjrzeć się panelowi klienta na naszym hostignu, gdyż prawdopodobnie będzie zawierał coś co nazywa się harmonogramem zadań, lub skontaktować się z administratorem jeśli nie ma możliwości samodzielnego dodawania zadań do systemowego cron.

Na różnych hostingach różnie to wygląda np. w Zenboxie nazwane jest to "Zadania cykliczne" . W formularzu należy uzupełnić informacje o godzinie uruchamiania zadania, lub wskazanie cyklu. Jako polecenia cron należy dodać komendę:

```
wget http://twojadomena/wp-cron.php /dev/null 2>&1
```



lub podać względną ścieżkę do pliku wp-cron.php (w zależności od konfiguracji środowiska).

Częstotliwość wyzwalania wp-crona zależy tylko od nas. Jednak wskazane jest kierowanie się zdrowym rozsądkiem :) Osobiście proponuję wywołanie nie mniej niż raz na godzinę, choć wszystko zależy od tego jakie pluginy mamy zainstalowane na stronie.

No ok mamy harmonogram WP podpięty pod systemowy harmonogram i... co dalej? Przecież system w tym przypadku zarządza wyzwalaniem harmonogramu WordPress'a. Należałoby jeszcze jakoś do wp-crona mieć możliwość dodawania zadań. Zatem idźmy dalej...

Jak dodać zadanie do harmonogramu WordPress?

Dodawanie nowych zadań jest względnie proste, choć pewnie nie dla początkującego użytkownika.

Należy w katalogu z motywem (wp-content/themes/) odnaleźć plik functions.php

Tutaj będziemy dodawać nowe funkcjonalności do naszej strony, ale też zadania dla wp-crona, czyli napiszemy funkcję wp którą ma wykonać cron oraz dodamy ją jako akcję.

Jako przykładowym rozwiązaniem posłużymy się kodem który będzie wysyłał co określony czas maila na wskazany adres. (Czyli np. poinformuje nas o wykonaniu zadań harmonogramu). Kod ten może być umieszczony jako element pliku functions.php ale też może funkcjonować jako oddzielna wtyczka (o sposobach umieszczania kodu w tych dwóch miejscach, zaletach i wadach danego rozwiązania napiszę niebawem)

```
function moja_funkcja() {  
    wp_mail( 'adres@odbiorcy.pl', 'Temat maila', 'Treść maila' );  
}  
add_action( 'moj_akcja', 'moja_funkcja' );
```

Czyli mamy już zadanie które ma wp-cron wykonać. Teraz czas na dopisanie zadania do harmonogramu.

```
if ( !wp_next_scheduled('moja_akcja') ) {  
    wp_schedule_event( current_time( 'timestamp' ), 'hourly',  
    'moja_akcja', 'argumenty_akcji' );  
    // Podstawowe przedziały czasowe: hourly, daily and twicedaily
```

```
}
```

Usuwanie zadań

Bardzo często popełniany jest błąd, który zaśmieca nam harmonogram zadań. Np. podczas odinstalowywania pluginu nie usuwamy zadania jego dotyczącego z harmonogramu. Usunięcie zadania jest w sumie bardzo proste. Wystarczy zastosować poniższy kod:

```
wp_clear_scheduled_hook( 'moja_akcja' );  
delete_action( 'moj_akcja', 'moja_funkcja' );
```

Inne opcje czasowe

W WordPressie mamy standardowo 3 przedziały czasowe: hourly, daily and twicedaily czyli co godzinę, raz i dwa razy dziennie. Możemy też ustalić sobie swoje dodatkowe przedziały.

```
function cron_time_intervals( $param ) {  
    return array( 'two_hours' => array(  
        'interval' => 7200, // sekundy  
        'display' => 'two_hours'  
    ) );  
}  
add_filter( 'cron_schedules', 'cron_time_intervals' );
```

To tylko kilka z wielu możliwości które daje nam świadome ingerowanie z środowisko serwerowe i konfigurację WordPress'a. Zapraszam do dyskusji i własnych prób i eksperymentów :)

Comments



Łukasz - 2018-06-15 09:25:58

Super, zamiast sprawdzać w doc, dostałem wszystkie informacje na tacy. Dzięki



Maciej C - 2017-12-03 10:28:16

Cron jest w stanie zrealizować wykonanie każdego polecenia które da się zapisać jako polecenie systemu. Zatem jeśli jesteś w stanie wywołać odpowiedni program z wp import all z poziomu wiersza poleceń (php) to praktycznie wszystko jesteś w stanie wrzucić do harmonogramu crona.



NaPeK - 2017-09-26 13:38:58

Świetny artykuł, da się jakoś podpiąć do crona darmowy wp import all ?