



CPV Lab Pro

TrueCache

Data Caching for High Volume Users

Up-to-date Online Documentation at:

<https://doc.cpvlab.pro/#/TrueCache>

Copyright © 2019 CPV Lab Pro (cpvlab.pro)

*Reproduction or sharing of this guide is strictly prohibited without
the authorized consent from cpvlab.pro.*

TrueCache – Data Caching for High Volume Users

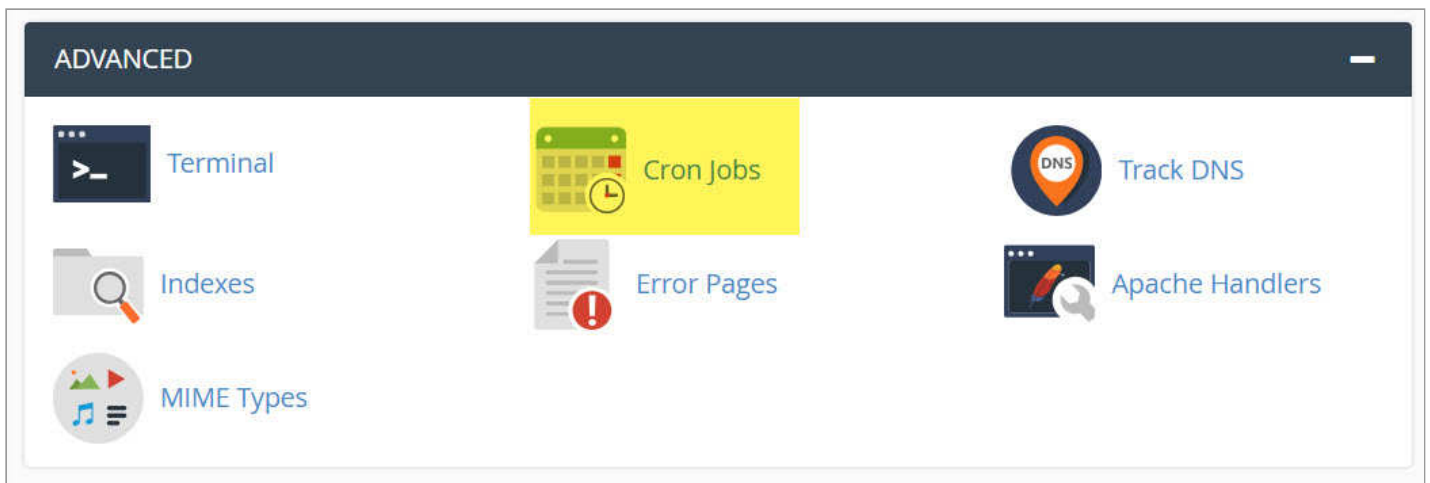
- **TrueCache** – is used for Data Caching by High Volume Users
- This feature addresses one of the biggest requests from Users: **More Volume** and **Fast Stats** within the CPV Lab Pro Interface.
- **Caching of data** in CPV Lab Pro means:
 - aggregating visitor details (clicks) and keeping them in a format that will allow **reports/stats to load faster** and that the “clicks” table will have a reduced size.
 - A CRON job is also setup to remove old records from the “clicks” table, to keep the DB clean and under control.
- CPV Lab Pro instances with small traffic figures should not use caching because it won't bring advantages in this case and will add an extra server load. Caching is effective for CPV Lab Pro Users that run **high volume traffic**.

Tips: if you have less than 500,000-700,000 clicks in the “clicks” table and the CPV Lab Pro interface and Stats pages are quick and responsive, you most likely won't see a direct benefit from caching.

Setting up the CRON Job

- The **Cron job** consists of a PHP page (`cron-cache.php`) that will be called periodically by the server and will perform the caching processes.

1. Login into cPanel and find the ‘Cron jobs’ page



- Enter the Cron job details in this page. The recommended interval for the job to run is every 10-15 minutes or more.

2. Then enter the command

- Enter the following command:

```
cd /home/<server-name>/public_html/cpvlabpro/; php cron-cache.php
```

- Or by an alternate command if the command above doesn't function properly on your server:

```
php /home/<server-name>/public_html/cpvlabpro/cron-cache.php
```

Make sure either of the commands you use from above, include the full path to the location of the “**cron-wurfl.php**” file on your server. This will vary based on your server configuration. Also replace **<server-name>** with the correct Server Name based on your server configuration.

A sample setup for the Cron job to run at 15 minutes over the hour, every hour (Recommended):

Add New Cron Job

Common Settings

-- Common Settings --

Minute:

*/15

Once Per Fifteen Minutes(*/15)

Hour:

*

Every Hour (*)

Day:

*

Every Day (*)

Month:

*

Every Month (*)

Weekday:

*

Every Day (*)

Command:

cd /home/<server-name>/public_html/cpvlabpro/; php cron-cache.php

Add New Cron Job

- Click the 'Add New Cron Job' button and the new Cron job will appear in the jobs list.

Current Cron Jobs

Minute	Hour	Day	Month	Weekday	Command	Actions
*/15	*	*	*	*	cd /home/<server-name>/public_html/cpvlabpro/; php cron-cache.php	Edit Delete

Edit Server Setting

- Before enabling caching, increase the "group_concat_max_len" system variable in **MySQL** to a larger value:
 - at least **1048576 (1MB)** for medium volumes of traffic
 - for high traffic, it could be **10485760 (10MB)**
- The maximum allowed value for this variable is 4294967295 (4GB), however you don't need to adjust this setting close to that.
- This is a system variable that should be changed by the web host since it cannot be modified from PhpMyAdmin on most servers.

Configuring Caching

- Caching can be configured based on the server settings and properties using 4 configuration keys from the [“constants-user.php”](#) file from the “lib” folder. If these keys don’t exist in your “constants-user.php” file, you can add them at the end of the file, before the last “}” sign.

Option Name	Default Value	Description
UseDataCaching	false	“true” if caching of data is used; beside this option, you also have to setup the Cron job
CachingMaxItems	10000	The maximum number of clicks processed in a single Cron job execution
CachingDelay	600	The delay in aggregating clicks; the clicks from the last xxx seconds won't be aggregated yet (in seconds)
CachingRemoveTime	2592000	The interval after which clicks are removed from the "clicks" table (after these clicks have been aggregated) - in seconds

A typical caching configuration section from the “constants-user.php” looks like this:

```
// Controls if cached data are used in application ("true" or "false" values); you must also setup the Cache Cron job
$GLOBALS['UseDataCaching'] = "false";
// The maximum number of clicks processed in a single Cron job execution
$GLOBALS['CachingMaxItems'] = 10000;
// The delay in aggregating clicks - clicks from the last xxx seconds won't be aggregated yet - in seconds
// 300 - 5 minutes
// 600 - 10 minutes
// 900 - 15 minutes
// 1800 - 30 minutes
$GLOBALS['CachingDelay'] = 600;
// The interval after which clicks are removed from the "clicks" table (after these clicks have been aggregated) - in seconds
// 86400 - 1 day
// 432000 - 5 days
// 864000 - 10 days
// 1728000 - 20 days
// 2592000 - 30 days
// 7776000 - 90 days
$GLOBALS['CachingRemoveTime'] = 2592000;
// The interval after which conversions cannot be marked in the aggregated data - in seconds
// 86400 - 1 day
// 432000 - 5 days
// 864000 - 10 days
// 1728000 - 20 days
// 2592000 - 30 days
// 7776000 - 90 days
$GLOBALS['CachingRemoveCacheRelation'] = 2592000;

//Controls if the GEO details are pulled by the Cron job; otherwise they are captured by the base.php page
$GLOBALS['UseGeoCron'] = "false";

// The maximum number of clicks geo-encoded in a single Cron job execution
$GLOBALS['GeoMaxItems'] = 20000;
```

The “CachingMaxItems” option

- Should be set to a value that is greater than the number of visitors you usually receive in the time interval between 2 consecutive Cron job executions.

- Setting this option to a large value will slow down the server on the first few Cron job runs if you have a large existing database with non-aggregated data (applies to upgrades because on new installations you won't have any non-aggregated data initially).

The “CachingDelay” option

- Specifies a time interval when clicks won't be aggregated by the Cron job.
- For example:
 - if you run the Cron job every hour and set the “**CachingDelay**” to 600 seconds (10 minutes)
 - then clicks from the last 10 minutes before the Cron job execution, won't be aggregated by this Cron job run.
 - **These clicks** will get aggregated by the **next Cron job execution**, when again the last 10-minutes clicks will be skipped and so on.
- This option is useful in order to aggregate clicks after visitors click to the offer page(s) for optimal caching of data.

The “CachingRemoveTime” option

- Specifies the time interval to keep non-aggregated clicks.
- **The non-aggregated clicks** will still be used by the [Visitors Stats page](#) and the Report Upload page, so if your server has the resources, it's recommended to keep 10, 20, 30, 60 days of non-aggregated clicks in the database together with the aggregated clicks.
- It is important to keep some non-aggregated data in the “clicks” table for Reports and [Visitor Stats](#).
 - If you want to see [Visitors Stats](#) from the last 60 days, then you have to keep 2 months of non-aggregated data in the “clicks” table by configuring the “**CachingRemoveTime**” with a value of at least “5184000” seconds (60 days).
- It's advised to start this with a higher value in order to not remove too much data initially and see how the server handles a larger number of non-aggregated clicks.
- You can set the value for "2592000" (60 days) or even higher using "7776000" seconds (90 days) as the value for "**CachingRemoveTime**"
 - this means you will be able to upload reports for the last 90 days and also see Visitors Stats for the last 90 days.
- Your **Server Admin** can fine tune these settings, as they are server-specific. Each server has its own configuration and settings. You Server Admin will be able to adjust these settings based on your server resources/hardware and your individual needs for your volume of traffic.

- They will check and monitor your Memory and CPU usage while the Caching / CRON Jobs are running in order to see how much extra load it adds and can make adjustments to run the Cron job while not affecting other server processes if the requirements are high.

The “GeoMaxItems” option

- To further optimize server sources, this key controls the maximum number of **clicks to be geo-encoded** in a single run of the **Cache Cron job** (this happens only if the UseGeoCron key is set to "true").
- The default value of the **GeoMaxItems** key is **20000 clicks**; this value should always be greater than your **CachingMaxItems** value you have set.

Automatic Clean-Up of Cache Tables

- This option works only when **TrueCache** is enabled and will remove records from the cache tables in order to reduce the database space, but will not actually delete aggregated data, so no clicks will be lost.
- It only removes click associations needed to mark conversions in the database.
 - In many cases, you likely won't have conversions within the same campaign, from Visitors who haven't returned in over 3 months for example. It will be safe to remove those associations after such a time interval.
- In order to configure this option edit the [“constants-user.php” file](#) from the “lib” folder and enter a range of days to keep the click associations. Enter this setting in seconds, in the key named **“CachingRemoveCacheRelation”**, as in the image below:

```
// The interval after which conversions cannot be marked in the aggregated data - in seconds
// 86400 - 1 day
// 432000 - 5 days
// 864000 - 10 days
// 1728000 - 20 days
// 2592000 - 30 days
// 7776000 - 90 days
$GLOBALS['CachingRemoveCacheRelation'] = 2592000;
```

- In the example above:
 - if you have visitors converting from within the same campaign after 3 months, then you would enter the setting as **“7776000”**
 - this will keep the associated data for 90 days then remove it.
- The automatic cleanup is performed, like the blocked clicks cleanup, by enabling at least one of the following 2 CRON jobs:
 - **Wurfl Cron job**

- **Campaigns Cron job**

Additional Automatic Clean-Up Features executed by enabling one of the 2 Cron jobs:

- **Blocked Clicks cleanup** – configured from the Settings page
- **Error Log cleanup** – configured from the Settings page
- **Non-Cookie Details cleanup** – configured from the Settings page
- **Cache Tables Associations cleanup** – configured from the “constants-user.php” file
- **Orphan datatable records cleanup** – without a configuration option since this is a task that should not be removed; it deletes any orphan records from the main tables in the database, records that are not needed anymore and will just take up database space

FAQ about Caching

1. Should I use caching?

If you are tracking high volumes of traffic, and notice any delays when loading stats or reports in the interface, then **Yes!** For Users that don't run high traffic, do not need to enable caching as it won't be required. You also need to have some server knowledge in order to enable and customize caching on your server.

2. How often should I setup the Cache Cron job to run?

- This depends on what you need to see in reports and if you need “real-time” data and the delay in data that you can accept. The default setting is to run the Cron job every hour, but you can lower or increase this interval based on your needs.
- It's very important that you don't set a very small interval that won't allow the previous Cron job execution to complete (for example if the Cron job takes 2 minutes each time to execute and you set the interval between executions to 1 minute).
- For Users with Solid Hosting, 5-15 minutes will work well. However, users with lower performance servers may need a higher time interval to complete the CRON job.

3. Visitors don't appear in Stats - I've tested with the Campaign URL and no visitors appear in the Stats page.

- When the Stats page uses cached data there is a delay before the Cron job runs, so visitors won't appear immediately. You can use the [Visitors Stats page](#) to check real-time traffic.

4. I'm keeping non-aggregated data in the "clicks" table together with aggregated data in the cache tables. Which version of data will be used in reports?

- When caching is enabled (with the "UseDataCaching" configuration option), pages that allow caching will use the aggregated data in order to lower the page load times.
- The Visitors Stats and Reports Upload pages will still use non-aggregated data.

5. Records disappeared from the "clicks" table

- This is normal and it is done in order to keep the "clicks" table size under control. Old visitors' details will be removed from the "clicks" table only after they were aggregated, so nothing is lost because reports can be obtained from the aggregated data.
- You can configure the time to keep clicks in the database with the "CachingRemoveTime" configuration option.

6. I have visitors that convert after a long interval of time (24 hours), should I set the "CachingDelay" configuration option to not aggregate data from the last 24 hours?

- No, that is not necessary. Conversions will be marked correctly in the database even if the click of the visitor the converted was previously aggregated. The conversion will be added to the aggregated records in this case.

7. I want to upload a list of converting subIDs for old visitors that were already aggregated. Will this work?

- Yes, this will work and the uploaded conversions will be marked in the aggregated data. This will work even if the original visitors' details were removed from the "clicks" table.

8. I want to remove a false conversion from the database for a visitor that was already aggregated. Is this possible?

- Yes, you will use the [Stats Management page](#) for this. The conversion and the associated revenue will be removed from the aggregated data.

9. Do I need to keep running the Cron job for the Campaigns page (cron-campaigns.php) if I run the Cron job for caching (cron-cache.php)?

- No, if you have caching enabled, you can Remove the Cron job for the Campaigns page. This process will be completed in the new CRON Job. In fact, the Cron job for the Campaigns page checks if caching is enabled and exits quickly in this case.

You may also find useful:

- [Setup Cron Jobs](#)
-

Last modified 07/21/2018

Copyright © 2019 CPV Lab Pro

[CPV Lab Pro Website](#) [Privacy Policy](#)