NAME

DBI::ProfileDumper::Apache - capture DBI profiling data from Apache/mod_perl

SYNOPSIS

Add this line to your *httpd.conf*:

```
PerlSetEnv DBI_PROFILE 2/DBI::ProfileDumper::Apache
```

(If you're using mod_perl2, see "When using mod_perl2" for some additional notes.)

Then restart your server. Access the code you wish to test using a web browser, then shutdown your server. This will create a set of *dbi.prof.** files in your Apache log directory.

Get a profiling report with dbiprof:

```
dbiprof /path/to/your/apache/logs/dbi.prof.*
```

When you're ready to perform another profiling run, delete the old files and start again.

DESCRIPTION

This module interfaces DBI::ProfileDumper to Apache/mod_perl. Using this module you can collect profiling data from mod_perl applications. It works by creating a DBI::ProfileDumper data file for each Apache process. These files are created in your Apache log directory. You can then use the dbiprof utility to analyze the profile files.

USAGE

LOADING THE MODULE

The easiest way to use this module is just to set the DBI_PROFILE environment variable in your *httpd.conf*:

```
PerlSetEnv DBI_PROFILE 2/DBI::ProfileDumper::Apache
```

The DBI will look after loading and using the module when the first DBI handle is created.

It's also possible to use this module by setting the Profile attribute of any DBI handle:

```
$dbh->{Profile} = "2/DBI::ProfileDumper::Apache";
```

See DBI::ProfileDumper for more possibilities, and DBI::Profile for full details of the DBI's profiling mechanism.

WRITING PROFILE DATA

The profile data files will be written to your Apache log directory by default.

The user that the httpd processes run as will need write access to the directory. So, for example, if you're running the child httpds as user 'nobody' and using chronolog to write to the logs directory, then you'll need to change the default.

You can change the destination directory either by specifying a `Dir` value when creating the profile (like `File` in the DBI::ProfileDumper docs), or you can use the `DBI_PROFILE_APACHE_LOG_DIR` env var to change that. For example:

```
PerlSetEnv DBI_PROFILE_APACHE_LOG_DIR /server_root/logs
```

When using mod_perl2

Under mod_perl2 you'll need to either set the `DBI_PROFILE_APACHE_LOG_DIR` env var, or enable the mod_perl2 `GlobalRequest` option, like this:

```
PerlOptions +GlobalRequest
```

to the global config section you're about test with DBI::ProfileDumper::Apache. If you don't do one of those then you'll see messages in your error_log similar to:

```
DBI::ProfileDumper::Apache on_destroy failed: Global $r object is not available. Set:
PerlOptions +GlobalRequest in httpd.conf at ..../DBI/ProfileDumper/Apache.pm line 144
```

Naming the files

The default file name is inherited from DBI::ProfileDumper via the *filename()* method, but DBI::ProfileDumper::Apache appends the parent pid and the current pid, separated by dots, to that name.

Silencing the log

By default a message is written to STDERR (i.e., the apache error_log file) when *flush_to_disk()* is called (either explicitly, or implicitly via DESTROY).

That's usually very useful. If you don't want the log message you can silence it by setting the `Quiet` attribute true.

```
PerlSetEnv DBI_PROFILE 2/DBI::ProfileDumper::Apache/Quiet:1

$dbh->{Profile} = "!Statement/DBI::ProfileDumper/Quiet:1";

$dbh->{Profile} = DBI::ProfileDumper->new(
Path => [ '!Statement' ]
Quiet => 1
);
```

GATHERING PROFILE DATA

Once you have the module loaded, use your application as you normally would. Stop the webserver when your tests are complete. Profile data files will be produced when Apache exits and you'll see something like this in your error_log:

```
DBI::ProfileDumper::Apache writing to /usr/local/apache/logs/dbi.prof.2604.2619
```

Now you can use dbiprof to examine the data:

```
dbiprof /usr/local/apache/logs/dbi.prof.2604.*
```

By passing dbiprof a list of all generated files, dbiprof will automatically merge them into one result set. You can also pass dbiprof sorting and querying options, see dbiprof for details.

CLEANING UP

Once you've made some code changes, you're ready to start again. First, delete the old profile data files:

```
rm /usr/local/apache/logs/dbi.prof.*
```

Then restart your server and get back to work.

OTHER ISSUES

Memory usage

DBI::Profile can use a lot of memory for very active applications because it collects profiling data in memory for each distinct query run. Calling `flush_to_disk()` will write the current data to disk and free the memory it's using. For example:

```
$dbh->{Profile}->flush_to_disk() if $dbh->{Profile};
```

or, rather than flush every time, you could flush less often:

```
$dbh->{Profile}->flush_to_disk()
if $dbh->{Profile} and ++$i % 100;
```

AUTHOR

Sam Tregar <sam@tregar.com>

COPYRIGHT AND LICENSE