Wikiprint Book

Title: Trac を FastCGI で使用する

Subject: SilverFrost - TracFastCgi

Version: 3

Date: 06/04/26 07:05:18

SilverFrost 目次

Trac を FastCGI で使用する

バージョン 0.9 以降、 Trac は FastCGI インタフェースに対応するようになりました。 mod_python 同様、 Trac を常駐させるため、外部の各リクエストに対して新しいプロセスを生成する CGI インタフェースよりも処理速度が速いです。その上 mod_python とは異なり SuEXEC に対応することも可能です。また、より多くの種類の Web サーバにサポートされています。

Windows 向けの Note: Trac の FCGI は Windows では使用できません。 Windows では_fcgi.py で使用されている Socket.fromfd が実装されていないためです。

単純な Apache の設定

Apache で利用可能な FastCGI モジュールは 2 種類あります: mod_fastcgi と mod_fcgid です。これ以降に書かれている FastCgiIpcDir と FastCgiConfig ディレクティブ は mod_fastcgi のディレクティブです; DefaultInitEnv は mod_fcgid のディレクティブです。

mod_fastcgi では、 Apache の設定ファイルに以下の設定を追記します。 ファイル:

```
# Enable fastcgi for .fcgi files
# (If you're using a distro package for mod_fcgi, something like
# this is probably already present)
<IfModule mod_fastcgi.c>
  AddHandler fastcgi-script .fcgi
  FastCgiIpcDir /var/lib/apache2/fastcgi
</IfModule>
LoadModule fastcgi_module /usr/lib/apache2/modules/mod_fastcgi.so
```

デフォルトの設定に問題がなければ、 FastCgiIpcDir の設定は必須ではありません。 LoadModule の行は IfModule グループの後になければいけないことに注意して下さい。

ScriptAlias を設定するもしくは似たオプションが TracCgi で説明されていますが、 trac.cgi の代わりに trac.fcgi を呼びます。

TRAC_ENV を以下のように設定することができます:

```
FastCgiConfig -initial-env TRAC_ENV=/path/to/env/trac
```

もしくは複数の Trac プロジェクトを扱っているときは、このように設定します:

```
FastCgiConfig -initial-env TRAC_ENV_PARENT_DIR=/parent/dir/of/projects
```

これらの設定は、 mod_fcgid では動きません。似ていますが mod_fcgid での部分的な解決策は以下の通りになります:

```
DefaultInitEnv TRAC_ENV /path/to/env/trac/
```

しかし、これは Directory や Location コンテキストで使用することができません。 よって、複数のプロジェクトに対応するのは難しくなります。

これらのモジュールの両方（同様に lighttpd と CGI も）で動かすよりよい方法は、 trac.fcgi に以下の値を設定することです。 Web サーバに環境変数を設定する必要がなくなります。例:

```
import os
os.environ['TRAC_ENV'] = "/path/to/projectenv"
```

or

```
import os
os.environ['TRAC_ENV_PARENT_DIR'] = "/path/to/project/parent/dir"
```

プロジェクトごとの ScriptAliases と .fcgi スクリプトを設定すれば、 この方法を使用して複数のプロジェクトに対応することができます。 trac.fcgi をコピーして、ファイル名を適切に変更し、上記のコードをそれぞれのスクリプトに追記します。

この fcgid 設定例 を見たところ、 ScriptAlias ディレクティブでは末尾の / も含めて、このように設定する:

```
ScriptAlias / /srv/tracsite/cgi-bin/trac.fcgi/
```

Simple Cherokee Configuration

Configuration wanted.

単純な Lighttpd の設定

FastCGI フロントエンドは最初 lighttpd のような、 Apache 以外の Web サーバのために開発されました。

lighttpd はセキュアで高速で、規格に準拠したとても柔軟な Web サーバで、高いパフォーマンスの環境で最適化されます。 他の Web サーバに比べて CPU や、メモリの占有率がとても少ないです。

trac.fcgi を lighttpd で使用するためには、 lighttpd.conf に以下の行を追加します:

```
fastcgi.server = ("/trac" =>
                  ("trac" =>
                    ("socket" => "/tmp/trac-fastcgi.sock",
                     "bin-path" => "/path/to/cgi-bin/trac.fcgi",
                     "check-local" => "disable",
                     "bin-environment" =>
                       ("TRAC_ENV" => "/path/to/projenv")
                    )
                  )
                )
```

動かしたい Trac のインスタンス毎に fastcgi.server のエントリを追加する必要があります。別の方法として、、上記の TRAC_ENV の代わりに TRAC_ENV_PARENT_DIR を使用でき、 lighttpd.conf に設定する代わりに trac.fcgi ファイルに bin-environment (上記の Apache の設定に書かれています) の2つのうちのどちらかを設定します。

lighttpd で2つのプロジェクトを動かすには、 lighttpd.conf に以下の設定を追加します:

```
fastcgi.server = ("/first" =>
                  ("first" =>
                   ("socket" => "/tmp/trac-fastcgi-first.sock",
                    "bin-path" => "/path/to/cgi-bin/trac.fcgi",
                    "check-local" => "disable",
                    "bin-environment" =>
                      ("TRAC_ENV" => "/path/to/projenv-first")
                    )
                  ),
                  "/second" =>
                    ("second" =>
                    ("socket" => "/tmp/trac-fastcgi-second.sock",
                     "bin-path" => "/path/to/cgi-bin/trac.fcgi",
                     "check-local" => "disable",
                     "bin-environment" =>
                       ("TRAC_ENV" => "/path/to/projenv-second")
                    )
                  )
                )
```

各フィールドの値が異なることに注意して下さい。もし .fcgi スクリプトに 環境変数を設定するほうが好ましい場合は、 trac.fcgi をコピー/名前変更をして下さい。例として、 first.fcgi と second.fcgi が上記の設定では参照されるようにします。
上記の設定で、両方のプロジェクトが 同じ trac.fcgi スクリプトで起動していても、 異なるプロセスになることに注意して下さい。

Note from c0Oi9Own: server.modules をロードする順番はとても重要です。もし、 mod_auth が mod_fastcgi より 先に ロードされる設定になっていない場合、サーバはユーザの認証に失敗します。

認証のために lighttpd.conf の 'server.modules' 中で mod_auth を有効にして、 auth.backend と認証方法を選択して下さい:

```
server.modules              = (
...
 "mod_auth",
...
)

auth.backend               = "htpasswd"

# Separated password files for each project
# See "Conditional Configuration" in
# http://trac.lighttpd.net/trac/file/branches/lighttpd-merge-1.4.x/doc/configuration.txt

$HTTP["url"] =~ "^/first/" {
 auth.backend.htpasswd.userfile = "/path/to/projenv-first/htpasswd.htaccess"
}
$HTTP["url"] =~ "^/second/" {
 auth.backend.htpasswd.userfile = "/path/to/projenv-second/htpasswd.htaccess"
}

# Enable auth on trac URLs, see
# http://trac.lighttpd.net/trac/file/branches/lighttpd-merge-1.4.x/doc/authentication.txt

auth.require = ("/first/login" =>
               ("method"  => "basic",
                "realm"   => "First project",
                "require" => "valid-user"
               ),
               "/second/login" =>
               ("method"  => "basic",
                "realm"   => "Second project",
                "require" => "valid-user"
               )
              )
```

パスワードファイルがない場合、 lighttpd (確認したバージョンは 1.4.3) が停止するので注意して下さい。

バージョン 1.3.16 以前では lighttpd は 'valid-user' をサポートしていないので注意してください。

条件付の設定は静的リソースをマッピングするときに便利です。例として FastCGI を経由せずに直接イメージファイルや CSS を参照するときなどです。:

```
# Aliasing functionality is needed
server.modules += ("mod_alias")

# Setup an alias for the static resources
alias.url = ("/trac/chrome/common" => "/usr/share/trac/htdocs")

# Use negative lookahead, matching all requests that ask for any resource under /trac, EXCEPT in
# /trac/chrome/common, and use FastCGI for those
$HTTP["url"] =~ "^/trac(?!/chrome/common)" {
# Even if you have other fastcgi.server declarations for applications other than Trac, do NOT use += here
fastcgi.server = ("/trac" =>
                  ("trac" =>
                    ("socket" => "/tmp/trac-fastcgi.sock",
                     "bin-path" => "/path/to/cgi-bin/trac.fcgi",
                     "check-local" => "disable",
                     "bin-environment" =>
                       ("TRAC_ENV" => "/path/to/projenv")
                     )
```

```
                  )
                )
}
```

複数のプロジェクトのそれぞれにエイリアスを作れば、複数のプロジェクトを動かすのは技術的には簡単です。 fastcgi.server を条件ブロックの中で宣言しラッピングします。 複数のプロジェクトをハンドルするもう一つの方法があります。 TRAC_ENV_PARENT_DIR を TRAC_ENV の代わりに使用し、グローバルの認証機構を使用します。サンプルを見てみましょう:

```
#  This is for handling multiple projects
 alias.url       = ( "/trac/" => "/path/to/trac/htdocs/" )

 fastcgi.server += ("/projects"  =>
                     ("trac" =>
                       (
                         "socket" => "/tmp/trac.sock",
                         "bin-path" => "/path/to/cgi-bin/trac.fcgi",
                         "check-local" => "disable",
                         "bin-environment" =>
                           ("TRAC_ENV_PARENT_DIR" => "/path/to/parent/dir/of/projects/" )
                       )
                     )
                   )
#And here starts the global auth configuration
 auth.backend = "htpasswd"
 auth.backend.htpasswd.userfile = "/path/to/unique/htpassword/file/trac.htpasswd"
 $HTTP["url"] =~ "^/projects/.*/login$" {
   auth.require = ("/" =>
                    (
                      "method"  => "basic",
                      "realm"   => "trac",
                      "require" => "valid-user"
                    )
                  )
 }
```

lighttpd では環境変数の LC_TIME を上書きして、日付/時間のフォーマットを変更することも出来ます。

```
fastcgi.server = ("/trac" =>
                   ("trac" =>
                     ("socket" => "/tmp/trac-fastcgi.sock",
                      "bin-path" => "/path/to/cgi-bin/trac.fcgi",
                      "check-local" => "disable",
                      "bin-environment" =>
                        ("TRAC_ENV" => "/path/to/projenv",
                        "LC_TIME" => "ru_RU")
                     )
                   )
                 )
```

使用言語指定の詳細については TracFaq の 2.13 の質問を参照して下さい。

その他重要な情報、例えば、 lighttpd の TracInstall や、 TracCgi などは fast-cgi 固有ではありませんが、インストールの詳細をつかむのに有用でしょう。

trac-0.9 を使用している場合、 些細なバグ について読んでください。

lighttpd を再起動し、ブラウザに `http://yourhost.example.org/trac` を入力して、 Trac にアクセスして下さい。

制限された権限で lighttpd を起動するにあたって気をつけること:

もし、 trac.fcgi が lighttpd の設定で server.username = "www-data" や server.groupname = "www-data" を設定しても起動せずどうしようもないときは、 bin-environment セクションの PYTHON_EGG_CACHE を www-data のホームディレクトリまたは www-data アカウントで書き込みが可能なディレクトリに設定して下さい。 (訳注: debian 系 Linux に限定した話だと思われます。 www-data は lighttpd を起動するユーザに適宜読み替えてください。)

簡単な LiteSpeeed の設定

FastCGI フロントエンドは最初 LiteSpeed のような、 Apache 以外の Web サーバのために開発されました。

LiteSpeed Web サーバはイベント駆動、非同期型であり、Apache に代わるものとしてセキュアで拡張可能になるようにゼロからデザインされています。そして、最低限のリソースで操作できます。 LiteSpeed は Apache の設定ファイルから直接操作でき、ビジネスに不可欠な環境をターゲットにしています。

セットアップ

1) 最初に Trac プロジェクトをインストールして動作することを確認して下さい。最初のインストールでは、 "tracd" を使用します。

2) このセットアップでは仮想ホストを作成します。以下、この仮想ホストのことを TracVhost と呼びます。このチュートリアルで、先ほど作ったプロジェクトが以下の URL 経由でアクセスできると仮定します:

```
http://yourdomain.com/trac/
```

3) "TracVhost → External AApps" タブへ移動し、新しい "External Application" を作成します。

```
Name: MyTracFCGI
Address: uds://tmp/lshttpd/mytracfcgi.sock
Max Connections: 10
Environment: TRAC_ENV=/fullpathto/mytracproject/ <--- path to root folder of trac project
Initial Request Timeout (secs): 30
Retry Timeout (secs): 0
Persistent Connection        Yes
Connection Keepalive Timeout: 30
Response Bufferring: No
Auto Start: Yes
Command: /usr/share/trac/cgi-bin/trac.fcgi  <--- path to trac.fcgi
Back Log: 50
Instances: 10
```

4) (非必須) "TracVhost → Security" タブへ移動し、新しいセキュリティ "Realm" を作成することができます。

```
DB Type: Password File
Realm Name: MyTracUserDB              <--- any name you wish and referenced later
User DB Location: /fullpathto/htpasswd <--- path to your htpasswd file
```

もし、 htpasswd ファイルを持っていない、もしくは作り方を知らない場合は、 http://sherylcanter.com/encrypt.php にアクセスし、ユーザ名:パスワード の一対を生成して下さい。

5) "PythonVhost → Contexts" へ移動し、新しい "FCGI Context" を作成します。

```
URI: /trac/                             <--- URI path to bind to python fcgi app we created
Fast CGI App: [VHost Level] MyTractFCGI  <--- select the trac fcgi extapp we just created
Realm: TracUserDB                        <--- only if (4) is set. select realm created in (4)
```

6) /fullpathto/mytracproject/conf/trac.ini を修正します。

```
#find/set base_rul, url, and link variables
base_url = http://yourdomain.com/trac/ <--- base url to generate correct links to
url = http://yourdomain.com/trac/      <--- link of project
link = http://yourdomain.com/trac/     <--- link of graphic logo
```

7) LiteSpeed を "lswsctrl restart" で再起動し、新しい Trac プロジェクトに以下の URL でアクセスします:

```
http://yourdomain.com/trac/
```

See also TracCgi, TracModPython, TracInstall, TracGuide