

Using FastCGI with Apache HTTP Server 2.4

Jeff Trawick

[http://emptyhammock.com/
trawick@emptyhammock.com](http://emptyhammock.com/trawick@emptyhammock.com)

April 8, 2014

Revisions

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

2014-04-10

Add `Require expr ...` to `/www/tools/`
configuration in *More classic CGI configuration*
slide to resolve a potential security hole. *Thank*
you Eric Covener!

Get these slides...

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

`http://emptyhammock.com/projects/info/slides.html`

Table of Contents

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- 1 The world of FastCGI
- 2 FastCGI with Apache httpd 2.4
- 3 Choosing
- 4 mod_fcgid
- 5 mod_proxy_fcgi
- 6 mod_authnz_fcgi
- 7 Other tools
- 8 PHP Applications and FastCGI
- 9 Future

Introduction — Who am I?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- I've worked at
 - several large corporations, for over two decades
 - my own one-person *company*, Emptyhammock, for the last two years
- I've worked on
 - several products which were primarily based on or otherwise included Apache HTTP Server
 - lower-level networking products
 - web applications
- I've developed or maintained some of the FastCGI support in the Apache HTTP Server project.

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

The world of FastCGI

FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- A protocol for communicating between a web server and persistent application processes which can handle any of several different phases of requests.

FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- A protocol for communicating between a web server and persistent application processes which can handle any of several different phases of requests.
- Implemented for most web servers.

FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- A protocol for communicating between a web server and persistent application processes which can handle any of several different phases of requests.
- Implemented for most web servers.
- Implemented for most programming languages and a number of frameworks.

Inputs and outputs are similar to CGI:

- environment variables
CONTENT_LENGTH, SCRIPT_NAME, etc.

Inputs and outputs are similar to CGI:

- environment variables
CONTENT_LENGTH, SCRIPT_NAME, etc.
- input stream for request body

Inputs and outputs are similar to CGI:

- environment variables
CONTENT_LENGTH, SCRIPT_NAME, etc.
- input stream for request body
- output stream for response headers and body

Inputs and outputs are similar to CGI:

- environment variables
CONTENT_LENGTH, SCRIPT_NAME, etc.
- input stream for request body
- output stream for response headers and body
- output stream for log messages

Inputs and outputs are similar to CGI:

- environment variables
CONTENT_LENGTH, SCRIPT_NAME, etc.
- input stream for request body
- output stream for response headers and body
- output stream for log messages

But binary encoded on a stream socket or pipe (Windows).
FastCGI process waits repeatedly for new connections.

Raw CGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
int main(int argc, char **argv)
{
    extern char **environ;
    /* environ is {"CONTENT_LENGTH=105",
                  "SCRIPT_NAME=/path/to/foo.fcgi", etc. } */
    const char *cl_str;

    if ((cl_str = getenv("CONTENT_LENGTH")) {
        read(FILENO_STDIN,,); /* request body */
    }
    write(STDOUT_FILENO,,); /* response headers
                             * and body */
    write(STDERR_FILENO,,); /* to web server log */
}
```

Raw FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
int main(int argc, char **argv)
{
    socket(); bind(); listen();
    while (1) {
        int cl = accept();
        read(cl, buf);
|0000 0101000100080000 0001000000000000 ..... |
|0010 0104000100190000 090e485454505f48 ..... ..HTTP_H|
|0020 4f53543132372e30 2e302e313a383038 OST127.0 .0.1:808|
|0030 3101040001002000 000f0f485454505f 1..... . ...HTTP_|
|0040 555345525f414745 4e54417061636865 USER_AGE NTApache|
|0050 42656e63682f322e 3301040001001000 Bench/2. 3.....|
|0060 000b03485454505f 4143434550542a2f ...HTTP_ ACCEPT*/|
        write(cl, buf);
|0000 01060001041a0600 436f6e74656e742d ..... Content-|
|0010 547970653a207465 78742f706c61696e Type: te xt/plain|
|0020 0a0a787878787878 7878787878787878 ..xxxxxx xxxxxxxx|
|0030 7878787878787878 7878787878787878 xxxxxxxxx xxxxxxxxx|
        close(cl);
    }
}
```


Raw FastCGI

How the datastream is defined:

- version — 1 byte, always 1
- type — 1 byte, stuff like begin-request, abort-request, params, stdin-data, stdout-data, etc.
- request id — 2 byte request identifier
- content length — 2 byte length of bundled data
- padding length — 1 byte length of padding data
- the data (0–64k bytes)
- the padding (0–255 bytes)

```
|0101000100080000 0001000000000000 ..... |
|0104000100190000 090e485454505f48 ..... ..HTTP_H|
```

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Programming support

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

After all, nobody would want to reinvent *that* protocol.

- FastCGI protocol libraries are available for use with Perl, Python, Ruby, C, etc., often based on the C library from Open Market.

Code to the API to implement a FastCGI application.

With some APIs, a properly coded FastCGI app will also work as plain CGI.

- PHP supports it transparently.
- Some frameworks support it transparently.

Perl example, moving from CGI to FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

CGI-only Perl script:

```
use CGI;
my $q = CGI->new;
$mode = $q->param('mode');
print $q->header(-type => 'text/html');
print $q->start_html('hello, world'),
      $q->h1('hello, world'),
      $q->end_html;
}
```

Perl example, moving from CGI to FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

CGI-only Perl script:

```
use CGI;
my $q = CGI->new;
$mode = $q->param('mode');
print $q->header(-type => 'text/html');
print $q->start_html('hello, world'),
      $q->h1('hello, world'),
      $q->end_html;
}
```

Dual-mode CGI/FastCGI Perl script:

```
use CGI::Fast;
while (my $q = CGI::Fast->new) {
    $mode = $q->param('mode');
    print $q->header(-type => 'text/html');
    print $q->start_html('hello, world'),
          $q->h1('hello, world'),
          $q->end_html;
}
```

Perl example, moving from CGI to FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI


Future

CGI-only Perl script:

```
use CGI;
my $q = CGI->new;
$mode = $q->param('mode');
print $q->header(-type => 'text/html');
print $q->start_html('hello, world'),
      $q->h1('hello, world'),
      $q->end_html;
}
```

Dual-mode CGI/FastCGI Perl script:

```
use CGI::Fast;
while (my $q = CGI::Fast->new) {
    $mode = $q->param('mode');
    print $q->header(-type => 'text/html');
    print $q->start_html('hello, world'),
          $q->h1('hello, world'),
          $q->end_html;
}
```

But beware of unintentionally saved state. 

Web server support for FastCGI applications

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Available for most web servers, including Apache httpd, IIS, Lighttpd, nginx, iPlanet, and others
- Typically implemented as a shared library (plug-in module) which can be loaded if the feature is desired

History

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- The FastCGI protocol was originally developed in 1995–1996 for a web server from Open Market, perhaps in response to the NSAPI programming model which allowed for C language plug-ins for the Netscape (now iPlanet once again) web server.

<http://www.fastcgi.com/devkit/doc/fcgi-spec.html>

- One web server implementation of particular interest, also from Open Market: mod_fastcgi 1.0 for Apache 1.0 in 1996

See the August 28, 1996 issue of Apache Week.

What happened after that?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- FastCGI was great for converting existing CGIs (often Perl) and drastically improving performance.

But:

- Native web server APIs were exploited more and more, either for existing scripting languages like Perl or new languages like PHP.
- Apache httpd modules took off. Web developers and deployers became accustomed to native code plug-ins.

What happened after that?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- FastCGI was great for converting existing CGIs (often Perl) and drastically improving performance.

But:

- Native web server APIs were exploited more and more, either for existing scripting languages like Perl or new languages like PHP.
- Apache httpd modules took off. Web developers and deployers became accustomed to native code plug-ins.

(Surplus of C programmers?)

Native module drawbacks

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Overall resource use often larger when app runs in the web server, especially for prefork model
 - memory
 - connections to database, LDAP, etc.

Resources are often left behind on any thread/process that occasionally runs the application — underutilized.

Native module drawbacks

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Overall resource use often larger when app runs in the web server, especially for prefork model

- memory
- connections to database, LDAP, etc.

Resources are often left behind on any thread/process that occasionally runs the application — underutilized.

- Introduces instability into server (these modules can be at least as complicated as the core server)

Native module drawbacks

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Overall resource use often larger when app runs in the web server, especially for prefork model

- memory
- connections to database, LDAP, etc.

Resources are often left behind on any thread/process that occasionally runs the application — underutilized.

- Introduces instability into server (these modules can be at least as complicated as the core server)
- Collisions between requirements of different modules

Native module drawbacks

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Overall resource use often larger when app runs in the web server, especially for prefork model
 - memory
 - connections to database, LDAP, etc.

Resources are often left behind on any thread/process that occasionally runs the application — underutilized.

- Introduces instability into server (these modules can be at least as complicated as the core server)
- Collisions between requirements of different modules
- Generally unable to support multiple script interpreter versions

Native module drawbacks

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Overall resource use often larger when app runs in the web server, especially for prefork model
 - memory
 - connections to database, LDAP, etc.
- Resources are often left behind on any thread/process that occasionally runs the application — underutilized.
- Introduces instability into server (these modules can be at least as complicated as the core server)
 - Collisions between requirements of different modules
 - Generally unable to support multiple script interpreter versions
 - Potential lack of thread safety, or expensive locking

But with FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Often the required application thread/process count is a fraction of that of the web server (so resources not left behind on threads/processes occasionally used).

But with FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Often the required application thread/process count is a fraction of that of the web server (so resources not left behind on threads/processes occasionally used).
- A particular application usually can't introduce instability into the server, so basic services and other applications are unaffected.

But with FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Often the required application thread/process count is a fraction of that of the web server (so resources not left behind on threads/processes occasionally used).
- A particular application usually can't introduce instability into the server, so basic services and other applications are unaffected.
- Different applications can use different libraries, interpreter versions, framework versions, etc.

But with FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Often the required application thread/process count is a fraction of that of the web server (so resources not left behind on threads/processes occasionally used).
- A particular application usually can't introduce instability into the server, so basic services and other applications are unaffected.
- Different applications can use different libraries, interpreter versions, framework versions, etc.
- Independent start/stop of web server and application

But with FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Often the required application thread/process count is a fraction of that of the web server (so resources not left behind on threads/processes occasionally used).
- A particular application usually can't introduce instability into the server, so basic services and other applications are unaffected.
- Different applications can use different libraries, interpreter versions, framework versions, etc.
- Independent start/stop of web server and application
- Independent identity or chroot env vs. web server and other applications

Relative cost of threads

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Threads that serve application requests are more expensive than threads that service static files or proxy requests elsewhere.

Relative cost of threads

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Threads that serve application requests are more expensive than threads that service static files or proxy requests elsewhere.
- If you run the application request on web server threads, you turn cheap threads into expensive threads.

Relative cost of threads

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Threads that serve application requests are more expensive than threads that service static files or proxy requests elsewhere.
- If you run the application request on web server threads, you turn cheap threads into expensive threads.
- If you run the application request on threads in dedicated application processes, you have lower resource consumption since you'll have fewer expensive threads.

Ummm, why are there fewer application threads?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Some requests don't run the application (static files, pings, etc.)

Ummm, why are there fewer application threads?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Some requests don't run the application (static files, pings, etc.)
- *When the application thread isn't a web server thread:* The application often isn't invoked until after all or much of the request is received by a server thread, and the application is relinquished before all of the response is sent to the client.

Ummm, why are there fewer application threads?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Some requests don't run the application (static files, pings, etc.)
- *When the application thread isn't a web server thread:* The application often isn't invoked until after all or much of the request is received by a server thread, and the application is relinquished before all of the response is sent to the client.
 - Some isolation from slower clients, for better utilization of the expensive threads

Criticisms of FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- generally troublesome implementations

*just switch from mod_fastcgi to mod_fcgid or httpd to
Lighttpd or ...*

Criticisms of FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- generally troublesome implementations

just switch from mod_fastcgi to mod_fcgid or httpd to Lighttpd or ...

- more or different configuration required for the application

Concerns for some special situations

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Protocol is complex enough that unsanitized input could expose bugs in non-mainstream protocol support *in app but also in server*

Concerns for some special situations

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Protocol is complex enough that unsanitized input could expose bugs in non-mainstream protocol support *in app but also in server*
- Care needed with TCP to protect access
 - Instant auth: Just set REMOTE_USER
 - Throw garbage at the TCP port, see what happens

AF_UNIX has filesystem permissions and is system-only.

Competitors

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- HTTP — If you use HTTP you can interoperate with almost anything. *But if you use HTTP you have a lot to implement to be able to interoperate with what people will throw at you.*
- SCGI — <http://python.ca/scgi/protocol.txt>
 - about 100 lines, so easy to implement yourself if existing library support isn't available or suitable
 - commonly used FastCGI capabilities, plus sendfileOne unfortunate omission: doesn't provide a way for routing stderr messages (integration into web server logs).

Competitors

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- CGI — why not, if load isn't an issue?
- AJP — not just for Java application servers
- custom

These have varying infrastructure to help with process management and protocol.

(And of course *mod_foo* and JVM or CLR-based interpreters.)

Non-competitors

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- providers of APIs
 - Modules such as `mod_wsgi` or `mod_php` provide an API rather than implement a well-known wire protocol such as FastCGI, HTTP, or SCGI.
 - This can use FastCGI or other protocols. *IOW, WSGI and PSGI are not competitors to FastCGI.*

So when would I run the application inside httpd

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Very lightweight interpreter (e.g., Lua)

So when would I run the application inside httpd

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Very lightweight interpreter (e.g., Lua)
- Very light load that isn't expected to increase (e.g., some non-Internet-facing special-purpose application)

So when would I run the application inside httpd

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Very lightweight interpreter (e.g., Lua)
- Very light load that isn't expected to increase (e.g., some non-Internet-facing special-purpose application)
- I know that I can continue to run the application inside httpd even if the server as a whole scales up or switches to a different MPM

FastCGI with Apache httpd 2.4

Primary FastCGI modules for Apache httpd 2.4

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Several modules are available from the Apache HTTP Server project, and others are in use as well.

These can be compared based on support for key features:

- Application process management
- Communication mechanism
- Supported FastCGI application roles

Application process management support

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Can the module manage the lifecycle of FastCGI application processes, based on load and the web server lifecycle?

- A module might only interact with processes it created.
- A module might only interact with sockets owned by processes it doesn't know about,
- A module might support both models.

Communication mechanism support

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

What types of IPC are supported for communication with the application?

- A module may or may not support TCP sockets.
- A module may or may not support Unix sockets (on Unix, of course).
- A module may or may not support local pipes (only on Windows in practice).

FastCGI role support

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

The FastCGI specification defines several different *roles* that a FastCGI application may implement:

responder role

Generating the response body in a manner similar to CGI (**the typical requirement**)

authorizer role

Deciding whether or not the user is authorized to access the requested resource. This generally is implemented with httpd-specific extensions for the three Apache httpd AAA phases.

filter role

Generating the response body based on an extra stream of input data.

Primary FastCGI modules for Apache httpd 2.4

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

	fcgid	fastcgi	proxy_fcgi	authnz_fcgi
Manages app processes?	Yes	Yes	No	No
Works with other processes?	No	Yes	Yes	Yes
TCP?	No	Yes	Yes	Yes
Local sockets or pipes?	Yes	Yes	Yes (Unix)	No
Responders?	Yes	Yes	Yes	No
AAA?	Yes	Yes	No	Yes (plus providers)
Filters?	No	No	No	No

Other FastCGI modules for Apache httpd 2.4

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Kean Johnston has mentioned his work-in-progress, `mod_extfcgi`, a couple of times on the httpd developer mailing list, but it is not yet available. The unique feature planned is support for the FastCGI *FILTER* role.
- *your module here*

About mod_authnz_fcgi...

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- This is not currently part of 2.4 releases; it must be obtained from httpd *trunk* and built separately from httpd.
- `http://svn.apache.org/viewvc/httpd/httpd/trunk/modules/aaa/mod_authnz_fcgi.c?view=co`
- `apxs -ci mod_authnz_fcgi.c`

Is it production ready? It has been tested by multiple parties, but it has not been formally reviewed for merging into the 2.4.x branch. I requested that last week.

About mod_fastcgi...

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- <http://www.fastcgi.com/>
- The latest *semi-official* code is a snapshot created in 2009; the last release was 2.4.6 in 2007.
- A reasonably-complete patch to add support for httpd 2.4 is available from a third party at <https://github.com/ByteInternet/libapache-mod-fastcgi>.
- Why is mod_fastcgi still interesting? Feature set / migration headaches...

We won't mention mod_fastcgi further.

I hate that, but there hasn't been a consolidation of patches different people are using for one issue or another, resulting in frustration and wasted effort.

Choosing among the contenders

Choosing

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

After kicking mod_fastcgi to the curb, choosing among FastCGI implementations is very simple:

- If you want something to manage the application process lifecycle in conjunction with httpd, **use mod_fcgid**.
- If you want different or more flexible application process lifecycle management, **use mod_proxy_fcgi** for *responders* and **use mod_authnz_fcgi** for *authorizers*.

mod_fcgid

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

mod_fcgid has

- *mostly* nice facilities for managing processes
- monitoring of processes via mod_status reports

mod_fcgid is missing

- load balancing to FastCGI processes *in the usual sense*
- support for FastCGI apps which can handle multiple concurrent requests or include their own process management

Key considerations:

- Pair it with `mod_authnz_fcgi` if you need to support *AUTHORIZER* role
- Does not currently support multiple requests per connection (i.e., you'll have lots of connection establishment, lots of sockets in certain states)
- As a scheme handler for `mod_proxy`, it shares many features with the rest of the `mod_proxy` family, including
 - Load balancing
 - I/O parameters
 - Balancer manager
- Unix socket support is a recent addition

Notable features beyond those typically implemented:

- Supports combined *authn/authz* like in the FastCGI spec
- Supports the modern *provider* configuration mechanism
- Supports application delivery of the response body with a failed *authn* check

mod_authnz_fcgi limitations

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Notable limitations beyond those mentioned earlier:

- Does not share connections with mod_proxy_fcgi
- Does not interoperate with mod_authn_socache
- Does not support multiple requests per connection
- Does not support the httpd *access control* phase

Configuration of ASF FastCGI modules

Configuration of ASF FastCGI modules — mod_fcgid

Simple CGI and FastCGI configuration

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

`mod_fcgid`

`mod_proxy_fcgi`

`mod_authnz_fcgi`

Other tools

PHP
Applications
and FastCGI

Future

Start with CGI:

```
Alias /fastcgi/ \  
/home/trawick/ApacheCon/inst/fastcgi/
```

```
<Location /fastcgi>  
Options +ExecCGI  
SetHandler cgi-script  
</Location>
```

IOW, enable the ExecCGI option and set the handler appropriately. (ScriptAlias kind-of does that)
Change the handler name to fcgid-script (mod_fcgid).

More classic CGI configuration

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
<Location /myapp/>
  AddHandler prettify txt
  Action prettify /tools/prettify.pl
</Location>
```

```
<Directory /www/tools/>
  Require expr %{reqenv:REDIRECT_HANDLER} == 'prettify'
  Options +ExecCGI
  SetHandler cgi-script
</Directory>
```

Again, change the handler name to fcgid-script (mod_fcgid).

Note the check for the expected handler; this ensures that scripts in that directory are not accessible except via the action configured for resources in other locations.

mod_fcgid — Is that all?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

It might be, unless...

- default timeouts or other I/O settings aren't okay
 - connect timeout, read/write timeout, hang detection

mod_fcgid — Is that all?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

It might be, unless...

- default timeouts or other I/O settings aren't okay
connect timeout, read/write timeout, hang detection
- default process management isn't okay

mod_fcgid — Is that all?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

It might be, unless...

- default timeouts or other I/O settings aren't okay
connect timeout, read/write timeout, hang detection
- default process management isn't okay
limits on numbers of processes, rules for shrinking the pool

mod_fcgid — Is that all?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

It might be, unless...

- default timeouts or other I/O settings aren't okay
connect timeout, read/write timeout, hang detection
- default process management isn't okay
limits on numbers of processes, rules for shrinking the pool
- minor protocol adjustments, environment variables, etc.

mod_fcgid — Is that all?

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

It might be, unless...

- default timeouts or other I/O settings aren't okay
connect timeout, read/write timeout, hang detection
- default process management isn't okay
limits on numbers of processes, rules for shrinking the pool
- minor protocol adjustments, environment variables, etc.

Additional configuration is likely except for sites with only relatively simple FastCGI applications.

Hung process detection

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

By default, if a request does not *complete* within five minutes the application will be terminated.

No way to disable. *Fixme.*

```
# my report generates output over a long period of  
# time; don't kill it
```

```
FcgidBusyTimeout 3600
```

```
# kill anything that doesn't respond within 30  
# seconds
```

```
FcgidBusyTimeout 30
```

I/O timeouts (hung process?)

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

By default, if no data can be read or written within 40 seconds, the application will be terminated.

```
# my report generates output over a long period of  
# time; don't kill it  
FcgidBusyTimeout 3600  
# oh, and there are long pauses between generation  
# of any output  
FcgidIOTimeout 300
```

Process management

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Simple stuff:

- `FcgidMaxProcesses` — global limit on number of processes
- `FcgidMaxProcessesPerClass` — limit on number of processes per application
- `FcgidIdleTimeout` — termination after idle for this long
- `FcgidMaxRequestsPerProcess` — termination after handling this many requests
- `FcgidProcessLifetime` — termination after alive for this long

Tuning of process management algorithms

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Spawn score: internal calculation which represents process activity for a FastCGI application; used to determine if a new instance (process) can be created.

```
# Set this high.  If actual score is higher for an app,  
# more instances can't be created.  
FcgidSpawnScoreUpLimit      7000
```

Tuning of process management algorithms

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Spawn score: internal calculation which represents process activity for a FastCGI application; used to determine if a new instance (process) can be created.

```
FcgidSpawnScoreUpLimit      7000
```

```
# Default value.  Each process creation adds this to the  
# score.
```

```
FcgidSpawnScore              1
```


Tuning of process management algorithms

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Spawn score: internal calculation which represents process activity for a FastCGI application; used to determine if a new instance (process) can be created.

```
FcgidSpawnScoreUpLimit      7000
FcgidSpawnScore              1
```

```
# By default, termination increases the score.  But why?
# If a process goes away, create additional headroom for
# creating a replacement.
```

```
FcgidTerminationScore       -1
```

Tuning of process management algorithms

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Spawn score: internal calculation which represents process activity for a FastCGI application; used to determine if a new instance (process) can be created.

```
FcgidSpawnScoreUpLimit      7000
FcgidSpawnScore              1
FcgidTerminationScore       -1

# Subtracted from the score each second.
FcgidTimeScore               3
```

Ugly tuning of process management algorithms

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

mod_fcgid scans for certain conditions at configurable intervals. The default values for the intervals are quite high for some — 120 seconds.

```
# Scan for processes which have exceeded idle timeout every  
# second.  
FcgidIdleScanInterval          0
```

Ugly tuning of process management algorithms

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

mod_fcgid scans for certain conditions at configurable intervals. The default values for the intervals are quite high for some — 120 seconds.

```
FcgidIdleScanInterval          0
```

```
# Scan for processes which need to be terminated every second.
```

```
FcgidErrorScanInterval        0
```

Ugly tuning of process management algorithms

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

mod_fcgid scans for certain conditions at configurable intervals. The default values for the intervals are quite high for some — 120 seconds.

```
FcgidIdleScanInterval          0
FcgidErrorScanInterval         0
```

```
# Scan for zombie processes every second.
#   (Why don't we just call waitpid() to see if any children
#   exited?)
FcgidZombieScanInterval        0
```

Request body limits

- `mod_fcgid` is very aggressive in isolating the application from slow clients, to the extent that it will first read the request body before sending to the application.

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Request body limits

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- `mod_fcgid` is very aggressive in isolating the application from slow clients, to the extent that it will first read the request body before sending to the application.
- `FcgidMaxRequestLen` limits the amount of body that will be spooled; the request fails (500) if the body exceeds this amount.

Request body limits

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

`mod_fcgid`

`mod_proxy_fcgi`

`mod_authnz_fcgi`

Other tools

PHP
Applications
and FastCGI

Future

- `mod_fcgid` is very aggressive in isolating the application from slow clients, to the extent that it will first read the request body before sending to the application.
- `FcgidMaxRequestLen` limits the amount of body that will be spooled; the request fails (500) if the body exceeds this amount.
- The default limit is 128K.

Set this lower if you can.

Don't trust the common WordPress configuration instructions that tell you to increase this to the max (denial of service), but be aware that you may need to increase it to handle larger uploads with WordPress or any other application.

Unfortunately this is a strike against cheaper web server threads.

Wrappers

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- a command which will run for certain requests — by container or extension
- typically is outside of the web space
- typically is a script which encapsulates command-line parameters and environment settings such as envvars and ulimits

```
<Location /phpapp/>  
    AddHandler fcgid-script .php  
    Options +ExecCGI  
    FcgidWrapper /usr/local/bin/php-wrapper .php  
</Location>
```

Command-specific options

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

`mod_fcgid`

`mod_proxy_fcgi`

`mod_authnz_fcgi`

Other tools

PHP
Applications
and FastCGI

Future

Most directives apply to every app in a certain container (vhost or per-dir).

`FcgidCmdOptions` allows many directives to be applied for a single specific command.

```
FcgidCmdOptions /path/to/info.pl \  
    IdleTimeout 30 \  
    InitialEnv VHOST=www2.example.com \  
    IOTimeout 5 \  
    MaxRequestsPerProcess 10000
```

mod_fcgid's server status support

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Jeff, this is where you show the sample server-status page.

Configuration of ASF FastCGI modules — mod_proxy_fcgi

Simplest mod_proxy_fcgi

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
LoadModule proxy_module modules/mod_proxy.so  
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
```

```
ProxyPass /app/ fcgi://127.0.0.1:10080/
```

Not so interesting; just replace “http” in the usual pattern with “fcgi”. There’s no need for ProxyPassReverse.

Add load balancing

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
```

```
ProxyPass /app/ balancer://app-pool/
<Proxy balancer://app-pool/>
    BalancerMember fcgi://127.0.0.1:10080
    BalancerMember fcgi://127.0.0.1:10081
</Proxy>
```

Again, just replace “http” with “fcgi”.

Use Unix sockets instead

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so

ProxyPass /app/ unix:/var/run/FCGI/hello|fcgi://127.0.0.1/
```

Add load balancing (again)

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
```

```
ProxyPass /app/ balancer://app-pool/
<Proxy balancer://app-pool/>
    BalancerMember unix:/var/run/FCGI/hello1|fcgi://127.0.0.1/
    BalancerMember unix:/var/run/FCGI/hello2|fcgi://127.0.0.1/
</Proxy>
```


Proxy examples showing important protocol data

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

We'll use this configuration with a Perl FastCGI script that spits out important variables:

```
ProxyPass /app1/ fcgi://127.0.0.1:10101/  
ProxyPass /app2/ unix:/tmp/app2|fcgi://127.0.0.1:10102/  
ProxyPass /app1a/ fcgi://127.0.0.1:10101/  
<Location /app1a/>  
    SetEnvIf Request_URI "." proxy-fcgi-pathinfo  
</Location>
```

The script

This is essentially printenv for FastCGI, that only prints certain variables.

```
#!/usr/bin/perl
use CGI::Fast;
my %dumpme = ("SCRIPT_NAME" => 1, ...);
while (my $q = CGI::Fast->new) {
    print "Content-Type: text/plain\r\n\r\n";
    foreach my $env (keys %ENV) {
        print "env $env = $ENV{$env}\n"
            if $dumpme{$env};
    }
}
```

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Running the script

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
$ spawn-fcgi -n -a 127.0.0.1 -p 10101 ./showenv.pl
```

```
...
```

```
$ spawn-fcgi -n -s /tmp/app2 ./showenv.pl
```

```
...
```

http://127.0.0.1:8080/app1/a/b/c?a=true

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
env SCRIPT_NAME = /app1/a/b/c
env FCGI_ROLE = RESPONDER
env GATEWAY_INTERFACE = CGI/1.1
env SERVER_PORT = 8080
env REQUEST_URI = /app1/a/b/c?a=true
env SERVER_ADDR = 127.0.0.1
env HTTP_HOST = 127.0.0.1:8080
env REQUEST_SCHEME = http
env QUERY_STRING = a=true
```

http://127.0.0.1:8080/app2/a/b/c?a=true

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
env SCRIPT_NAME = /app2/a/b/c
env FCGI_ROLE = RESPONDER
env GATEWAY_INTERFACE = CGI/1.1
env SERVER_PORT = 8080
env REQUEST_URI = /app2/a/b/c?a=true
env SERVER_ADDR = 127.0.0.1
env HTTP_HOST = 127.0.0.1:8080
env REQUEST_SCHEME = http
env QUERY_STRING = a=true
```

http://127.0.0.1:8080/app1a/a/b/c?a=true

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

```
env SCRIPT_NAME = /app1a
env FCGI_ROLE = RESPONDER
env GATEWAY_INTERFACE = CGI/1.1
env SERVER_PORT = 8080
env REQUEST_URI = /app1a/a/b/c?a=true
env SERVER_ADDR = 127.0.0.1
env HTTP_HOST = 127.0.0.1:8080
env PATH_INFO = /a/b/c
env REQUEST_SCHEME = http
env QUERY_STRING = a=true
```

Configuration of ASF FastCGI modules — mod_authnz_fcgi

FastCGI AUTHORIZER

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

This example uses the provider interface for collaboration with other providers in more complex configurations.

This is for a standard *AUTHORIZER* that will perform the equivalent of authentication and authorization. (It will not see *FCGI_APACHE_ROLE*.)

```
AuthnzFcgiDefineProvider authnz FooAuthnz fcgi://localhost:10103/  
<Location /protected/>  
    AuthType Basic  
    AuthName "Restricted"  
    AuthBasicProvider FooAuthnz  
    Require FooAuthnz  
</Location>
```

The mod_authnz_fcgi documentation has extensive examples, including FastCGI script source code which shows what information will be available.

Managing external processes

(if you're not using mod_fcgid)

Types of tools

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

A couple of tools shown here only get the application processes started. They aren't any help by themselves if the processes terminate unexpectedly.

A couple of other tools shown here can be used to control lifecycle (potentially with the system lifecycle) and ensure availability by reacting when the processes terminate unexpectedly.

- It only supports TCP/IP sockets, and only on Unix.
- In general this implements a rather basic subset of features in spawn-fcgi (from the Lighttpd folks).
- On the bright side, it is bundled with httpd 2.4, and is thus always available for testing.

```
$ fcgistarter -c $PWD/TestAuthn.pl -p 10000
```

The path to the application is resolved *after* changing directory to /, so a fully-qualified path is required.

Notable features beyond fcgistarter:

- Unix socket support
- pid file support
- no-fork mode for use with higher-level programs like daemontools
- chroot, set user, set group, and other identity-related features

and better diagnostics messages

```
$ spawn-fcgi -p 10000 TestAuthn.pl
```

- Relatively popular for managing FastCGI applications
- Create a service script for the application that daemontools can use to up application and framework-specific envvars and start the application.
- Use `svc` command to handle lifecycle (`-u` for *up*, `-d` for *down*).

Example `/etc/service/SERVICE-NAME/run`:

```
#!/bin/sh
export PERL5LIB=/home/trawick/perl5/lib/perl5
cd /tmp
exec 2>&1
exec spawn-fcgi -p 10000 -u trawick -n \  
    /home/trawick/bin/TestAuthn.pl
```

supervisor a.k.a. supervisord

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP

Applications
and FastCGI

Future

- Broadly similar to daemontools
- Can create sockets
- Knows about FastCGI in particular and groups of processes listening on the same socket

Example `VIRTUALENV-ROOT/etc/supervisord.conf`:

```
[supervisord]

[fcgi-program:testauthn]
command=/home/trawick/bin/TestAuthn.pl
socket=tcp://localhost:10001
process_name=%(program_name)s_%(process_num)02d
numprocs=4
user=trawick
environment=PERL5LIB=/home/trawick/perl5/lib/perl5
directory=/tmp
```

PHP-FPM

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Part of modern PHP distributions
- Configurable management of PHP FastCGI processes
- php-fpm is alternative to php-cgi

```
pid = /var/log/httpd/php-fpm.pid
error_log = /var/log/httpd/php_error_log
log_level = debug
listen = 127.0.0.1:8081
pm = dynamic
pm.max_children = 20
pm.min_spare_servers = 5
pm.max_spare_servers = 20
pm.max_requests = 0
user = george
group = ringo
```

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

API/protocol adapters

API/protocol adapters

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Web applications written to some standard API may need an adapter to create a FastCGI protocol endpoint.

A Django example, using flup for WSGI under the covers:

```
$ ./manage.py runfcgi host=127.0.0.1 port=8000
```

A Perl example, using plackup for PSGI:

```
$ plackup -s FCGI --listen /var/run/FCGI/hello hello.psgi
```

You'll probably need to use this in conjunction with a more general process management tool for reliability and lifecycle management.

PHP Applications and FastCGI

PHP and FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Not at all uncommon...

FastCGI has long been required or recommended for PHP with

- nginx
- Lighttpd
- Zeus
- IIS

PHP and FastCGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Not at all uncommon...

FastCGI has long been required or recommended for PHP with

- nginx
- Lighttpd
- Zeus
- IIS

Can work fine with Apache httpd too

PHP with mod_fcgid

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- PHP FastCGI processes **did** exit after 500 requests
- Now that is disabled by default with PHP-FPM
- If you need to use the feature:

Synchronize mod_fcgid and PHP limits to avoid 500 error.

PHP with mod_fcgid

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- PHP FastCGI processes **did** exit after 500 requests
- Now that is disabled by default with PHP-FPM
- If you need to use the feature:

Synchronize mod_fcgid and PHP limits to avoid 500 error.

In PHP wrapper:

```
PHP_FCGI_MAX_REQUESTS=10000
```

PHP with mod_fcgid

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- PHP FastCGI processes **did** exit after 500 requests
- Now that is disabled by default with PHP-FPM
- If you need to use the feature:

Synchronize mod_fcgid and PHP limits to avoid 500 error.

In PHP wrapper:

```
PHP_FCGI_MAX_REQUESTS=10000
```

In PHP-FPM configuration:

```
pm.max_requests = 10000
```

PHP with mod_fcgid

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- PHP FastCGI processes **did** exit after 500 requests
- Now that is disabled by default with PHP-FPM
- If you need to use the feature:

Synchronize mod_fcgid and PHP limits to avoid 500 error.

In PHP wrapper:

```
PHP_FCGI_MAX_REQUESTS=10000
```

In PHP-FPM configuration:

```
pm.max_requests = 10000
```

In fcgid configuration:

```
FcgidMaxRequestsPerProcess 10000
```


mod_fcgid PHP limitation

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- PHP FastCGI process management ineffective (wasted) with mod_fcgid, which routes only single concurrent requests to the socket of a process which it has spawned.

PHP-FPM: If you need to use this for other reasons, leave PHP child process management disabled:

```
pm static  
pm.max_children 0
```

PHP-CGI:

```
PHP_FCGI_CHILDREN=0
```

mod_fcgid PHP limitation

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

But:

- With PHP process management, single cache can be used concurrently by many processes.

mod_fcgid PHP limitation

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

But:

- With PHP process management, single cache can be used concurrently by many processes.
- Without PHP child process management, PHP opcode caches are not very effective. Cache is serially reused within single process when the same fcgid-spawned process handles another request.

mod_fcgid PHP difference

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- PHP flags in .htaccess files — no longer respected when you move from mod_php to FastCGI
- on Windows, mod_php strips the drive letter from SCRIPT_NAME; mod_fcgid doesn't

mod_fcgid PHP example with PHP-CGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

.conf:

```
LoadModule fcgid_module modules/mod_fcgid.so
```

```
FcgidMaxRequestsPerProcess 5000
```

```
# Uncomment the following line if cgi.fix_pathinfo is set to 1 in  
# php.ini:
```

```
# FcgidFixPathinfo 1
```

```
Alias /php/ /home/trawick/myhg/apache/documents/AC2012EU/php/  
<Directory /home/trawick/myhg/apache/documents/AC2012EU/php/>
```

```
Options +ExecCGI
```

```
AddHandler fcgid-script .php
```

```
FcgidWrapper /home/trawick/myhg/apache/documents/AC2012EU/php-wrapper.sh
```

```
Require all granted
```

```
</Directory>
```

mod_fcgid PHP example with PHP-CGI

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing
mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Wrapper script:

```
#!/bin/sh
export PHPRC=/home/trawick/myhg/\
apache/documents/AC2012EU/
export PHP_FCGI_MAX_REQUESTS=5000
export PHP_FCGI_CHILDREN=0
exec /usr/bin/php-cgi
```

and be sure to make this script executable

PHP and mod_proxy_fcgi

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing
mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

General considerations:

- Use PHP-FPM — no limitations like you have with mod_fcgid.
- With most recent httpd 2.4.x on Unix, use Unix sockets for better speed.

Details of configuration:

<http://wiki.apache.org/httpd/PHP-FPM>.

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

The future

Concern about mod_fcgid

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

My 2¢...

- mod_fcgid's inability to route more than one concurrent request per spawned process affects its viability for PHP deployments, a traditional area of strength.
- mod_fcgid is not currently getting sufficient love to keep up with bug reports, or resolve long-standing limitations. A couple of more interested developers is required to keep mod_fcgid moving forward.

A potentially interesting path...

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- Hypothetical mod_XXX that can manage application processes
 - Process lifetime fits within that of httpd
 - Configured in .conf
 - Sets up listening sockets or pipes if needed
 - Same implementation, same configuration mechanism for command-line, ports, envvars, etc. regardless of protocol
- Other modules, such as mod_proxy, should be able to symbolically reference the addresses of applications managed by mod_XXX to avoid duplicating that aspect of the configuration.

A potentially interesting path...

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Or...

- mod_XXX can issue commands to a more general purpose process manager like daemontools or supervisord to bring up the application processes with server startup and *vice versa* on server termination.
- Since in this case mod_XXX would not own the configuration for the application processes, other mod_proxy or other httpd modules probably wouldn't be able to symbolically reference aspects of the configuration.

More information about the ASF modules

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

Documentation:

- http://httpd.apache.org/mod_fcgid/mod/mod_fcgid.html
- http://httpd.apache.org/docs/2.4/mod/mod_proxy_fcgi.html
- http://httpd.apache.org/docs/trunk/mod/mod_authnz_fcgi.html
- <http://wiki.apache.org/httpd/PHP-FPM>

Mailing lists:

- User support:
<http://httpd.apache.org/userslist.html>
- Development:
<http://httpd.apache.org/lists.html#http-dev>

More general information

Using FastCGI
with Apache
HTTP Server
2.4

Jeff Trawick

The world of
FastCGI

FastCGI with
Apache httpd
2.4

Choosing

mod_fcgid

mod_proxy_fcgi

mod_authnz_fcgi

Other tools

PHP
Applications
and FastCGI

Future

- ApacheCon NA 2010 preso with more information on mod_fcgid and (*versus*) mod_fastcgi: <http://people.apache.org/~trawick/AC2010-FastCGI.pdf>
- ApacheCon EU 2012 preso with information on interfacing with more types of external application processes: <http://people.apache.org/~trawick/AC2012EU-Apps.pdf>