

# Web server and Apache

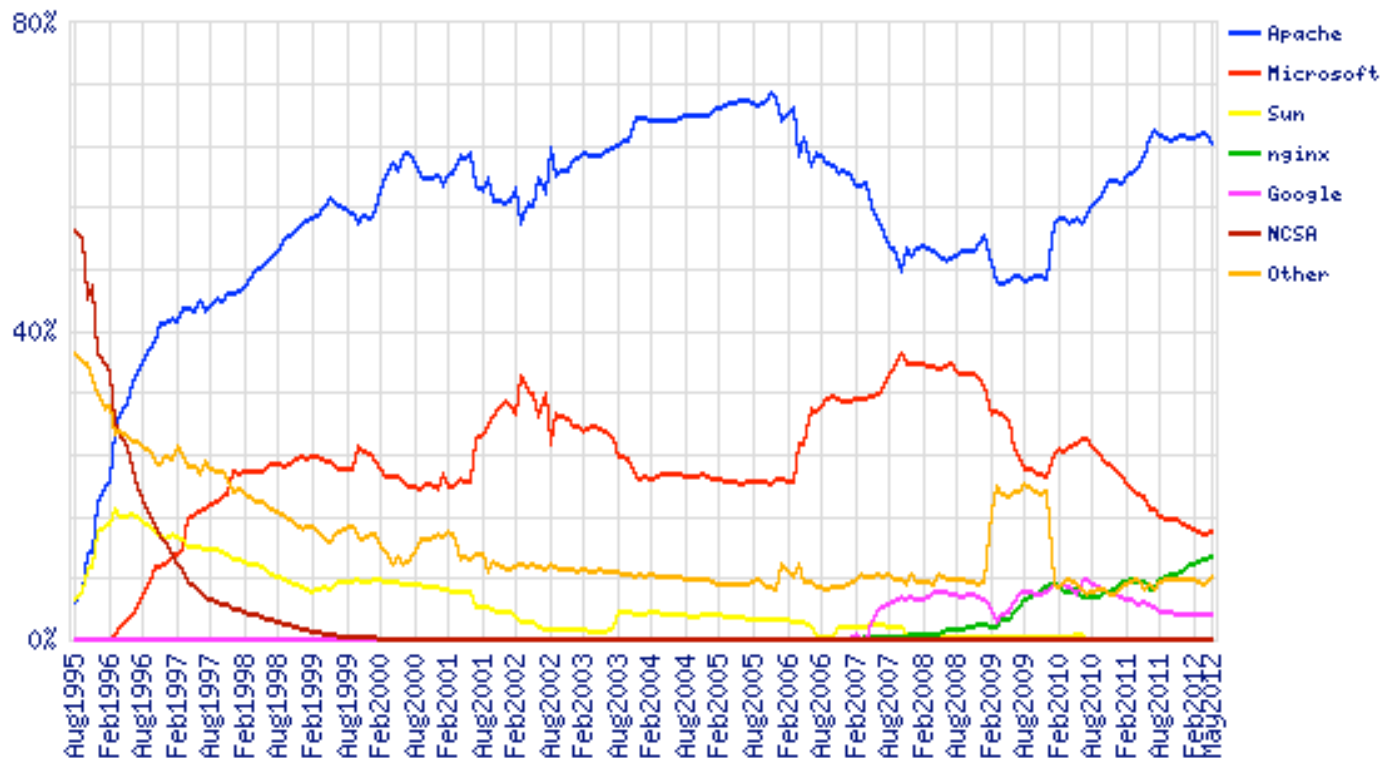
Lorenzo Bracciale

Marco Bonola

# Apache diffusion

Market Share for Top Servers Across All Domains

August 1995 - May 2012



Developer	April 2012	Percent	May 2012	Percent	Change
Apache	443,102,561	65.46%	425,631,721	64.20%	-1.26
Microsoft	92,488,751	13.66%	92,406,480	13.94%	0.28
nginx	69,869,916	10.32%	70,764,248	10.67%	0.35
Google	22,039,901	3.26%	21,264,616	3.21%	-0.05

because of  
user-generated content →

# Where to study

- Apache Server 2 - Mohammed J. Kabir
  - Hungry Minds
  
- Apache Server 2 Official Documentation
  - <http://httpd.apache.org/docs/2.0/>

# Apache

- developed by the *Apache Software Foundation* (released April 1995)
- bundled in the popular “LAMP” (**L**inux **A**pache **M**ysql **P**hp)
  
- apt-get install apache2
  
- Start / stop:
  - /etc/init.d/apache2 start
  - /etc/init.d/apache2 stop

BeOS	<u>beos</u>
Netware	<u>mpm_netware</u>
OS/2	<u>mpmt_os2</u>
Unix	<u>prefork</u>
Windows	<u>mpm_winnt</u>

# How apache works: MPM

- MultiProcessing Modules (**MPMs**) since Apache2:
  - In apache 1.3 uses a preforking architecture
    - the parent creates/destroys children if required
    - does not work well on some platform (such Windows)
  - MPM offers several alternatives (implemented in MPM modules) :
    - prefork MPM (like Apache 1.3)
    - worker MPM (each child, several threads)
    - perchild MPM (multiprocess/mutithread as well, fixed amount of thread, different hosts for different uid)
    - winnt MPM: single process, multithread (specific for windows)

```
<IfModule mpm_prefork_module>
StartServers      5
MinSpareServers  5
MaxSpareServers  10
MaxClients       150
MaxRequestsPerChild  0
</IfModule>
```



We can tune parameters in /etc/apache2/apache2.conf



```
<IfModule mpm_worker_module>
StartServers      2
MaxClients       150
MinSpareThreads  25
MaxSpareThreads  75
ThreadsPerChild  25
MaxRequestsPerChild  0
</IfModule>
```

# Processes vs Threads

- Both threads and processes are methods of parallelizing an application
- **Processes** are independent execution units that contain their own state information, use their own address spaces, and only interact with each other via interprocess communication mechanisms
- **Threads** share the same state and same memory space, and can communicate with each other directly, because they share the same variables

Are your cgi library thread safe?

# Apache Web server files

```
# Include generic snippets of statements
Include /etc/apache2/conf.d/

# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/
```

<code>/usr/sbin/apache2</code>	Apache 2 server <u>binary</u>
<code>/usr/sbin/apache2ctl</code>	Apache2 <u>control interface</u> (configtest could help!)
<code>/etc/apache2/apache2.conf</code>	default <u>configuration</u> file (could be overwritten during apache upgrade)
<code>/etc/apache2/httpd.conf</code>	user <u>configuration</u> files (and files inside /etc/apache2/conf.d)
<code>/etc/apache2/conf.d</code>	other <u>configuration</u> files (included as well in apache2.conf)
<code>/etc/apache2/sites-available</code>	<u>configuration</u> files for virtual hosting
<code>/etc/apache2/sites-enabled</code>	<u>symbolic links</u> to sites-available files (created with a2ensite, a2dissite)
<code>/etc/apache2/mods-available</code>	<u>configuration</u> files for modules
<code>/etc/apache2/mods-enabled</code>	<u>symbolic links</u> to mods-available files (created with a2enmod, a2dismod)
<code>/var/log/apache2</code>	<u>log files</u>

# Configuring Apache

- ~ 360 directives (!!!). Few selected:
  - ServerRoot: path to configuration, error and log files
  - PidFile
  - ServerName: name and port of the server
  - DocumentRoot: where find files to serve
  - ErrorDocument: override standard error messages



# Apache Modules

- Apache has modular architecture:
  - To enable/disable modules : `a2enmod / a2dismod MODNAME`
  - configurable via directives
  - `apache2ctl -M #list`
  - `mod_so` load module at runtime (Dynamic Shared Object (DSO) mechanism) `LoadModule`

- ◆ **Environment-related:** These directives allow you to set and reset environment variables.
- ◆ **Authentication and access control:** These directives allow you to authenticate and authorize user access to restricted parts of your Web site.
- ◆ **Dynamic contents generation:** These directives allow you to run external programs such as CGI scripts or Server Side Includes to create dynamic contents.
- ◆ **Content-type configuration:** These directives allow you to control MIME types of files.
- ◆ **Directory listing:** These directives allow you to control how directory listings are formatted.
- ◆ **Response header:** These directives allow you to control HTTP response headers.
- ◆ **Server information and logging:** These directives allow you to control server logs and status information.
- ◆ **URL mapping:** These directives allow you to map, rewrite, and create aliases for a URL.
- ◆ **Miscellaneous modules:** These directives allow you to control miscellaneous aspects of Apache such as proxy service, WEBDEV module, etc.

# Logging

```
ErrorLog /var/log/apache2/cgrrlweb.log  
  
# Possible values include: debug, info, notice, warn, error, crit,  
# alert, emerg.  
LogLevel warn
```

- location and content: CustomLog directive
- Format: LogFormat
  - specified with common logfile format\*

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\" vhost_combined
```

(\*) <http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format>

not only for  
apache!

# Logrotate

Log Size: Typically 1 MB for 10000 requests so...

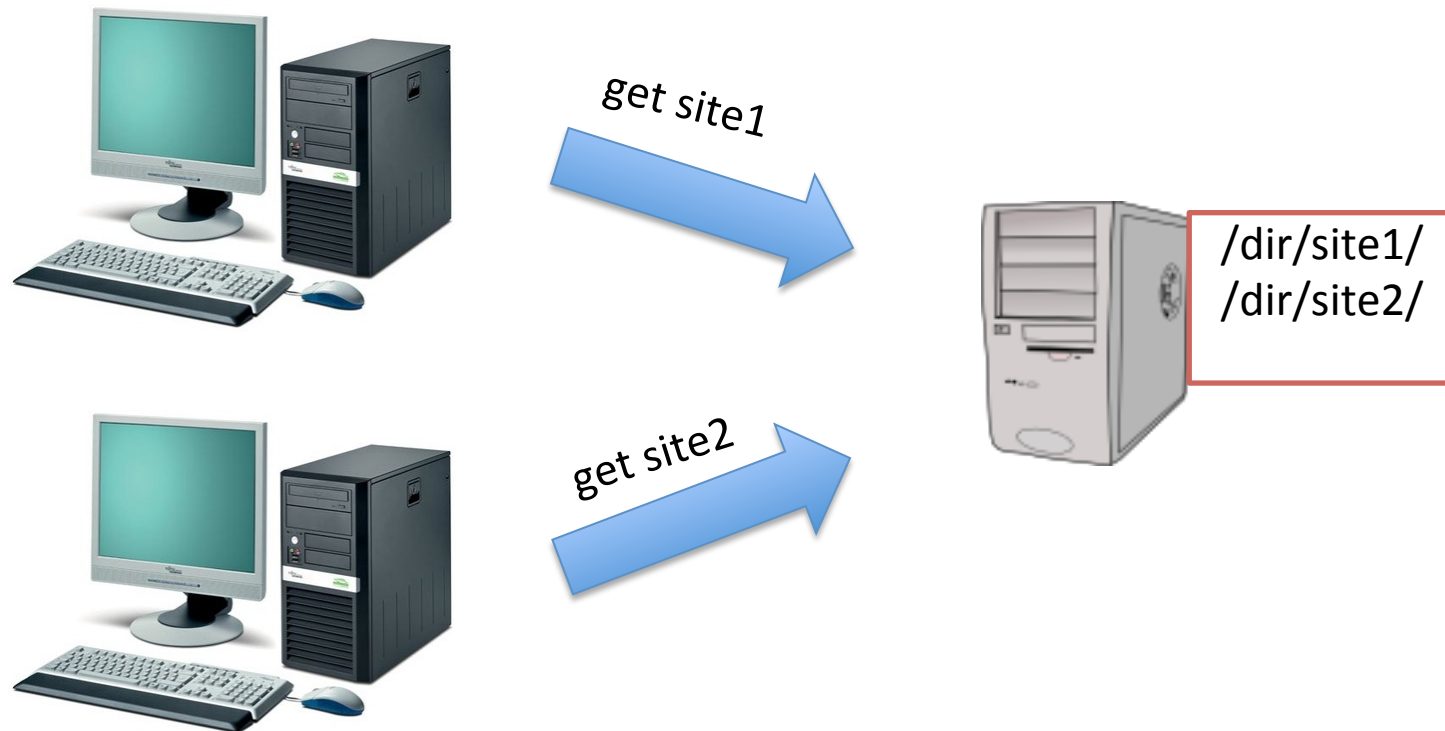
[/etc/logrotate.d/apache2](#)

```
/var/log/apache2/*.log {
    weekly
    missingok
    rotate 52
    compress
    delaycompress
    notifempty
    create 640 root adm
    sharedscripts
    postrotate
        if [ -f "`cat /etc/apache2/envvars`" ]; then
            /etc/init.d/apache2 reload > /dev/null
        fi
    endscript
}
```

- rotate at most 52 times, weekly
- compress (you can see that using zcat, zless or pipelining gzip and cat/tail)

# Virtual Hosting

- Problem: Several websites, one webserver
  - Typically: *name-based* virtual host (but IP based is possible as well)



# HTTP Request

Transmission Control Protocol, Src Port: 49845 (49845), Dst Port: http (80), Seq: 1, Ack:

Hypertext Transfer Protocol

GET /cgrl/ HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET /cgrl/ HTTP/1.1\r\n]

Request Method: GET

Request URI: /cgrl/

Request Version: HTTP/1.1

Host: stud.netgroup.uniroma2.it\r\n

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_6\_8) AppleWebKit/534.55.3 (KHTML,

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8\r\n

Accept-Language: en-us\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: keep-alive\r\n

\r\n

# HTTP Response

```
▶ Transmission Control Protocol, Src Port: http (80), Dst Port: 49845 (49845), Seq: 1, Ack: 341, Len: 1082
▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 200 OK\r\n
    ▶ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Request Version: HTTP/1.1
      Response Code: 200
      Date: Tue, 08 May 2012 15:35:23 GMT\r\n
      Server: Apache\r\n
    ▶ Content-Length: 888\r\n
      Keep-Alive: timeout=15, max=100\r\n
      Connection: Keep-Alive\r\n
      Content-Type: text/html; charset=ISO-8859-1\r\n
      \r\n
  ▼ Line-based text data: text/html
    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">\n
    <html>\n
      <head>\n
        <title>Index of /cgrl</title>\n
```

# Virtual Hosting: example

put virtual hosts confs in sites-available dir!

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerName www.domain.tld
    ServerAlias domain.tld *.domain.tld
    DocumentRoot /www/domain
</VirtualHost>

<VirtualHost *:80>
    ServerName www.otherdomain.tld
    DocumentRoot /www/otherdomain
</VirtualHost>
```

IP/Port for listening requests

Name of the virtual host (match **host** http field)

Where the html files are (simple case)

use /etc/hosts to test virtual host without DNS modifications

# DocumentRoot

- Where apache finds your documents (html files etc)
  - Typically search for: index.html index.cgi index.pl index.php index.xhtml index.htm
  - Defined from DirectoryIndex (order matters)

```
<html>  
  <body><h1>  
    HELLO CGRL  
  </h1></body>  
</html>
```

index.html sample file



# Exercise 1: virtual hosting

1. Create two dirs “cgrl-web” and “yourname-web”. Put in the directory two index.html files
2. Configure 2 virtual web hosts
  - [www.cgrl.edu](http://www.cgrl.edu)
  - [www.yourname.com](http://www.yourname.com)
3. Start apache
4. modify /etc/hosts
5. View your websites with a browser

# Standard Container Directives

- Many Container contexts:
  - <VirtualHost ...>: already seen...
  - <Directory>: applies one or more directives to a directory
  - <Files>: applies one or more directives to a file
  - <Location>: applies one or more directives to a URL
- AllowOverride: enable/disable directories directives overriding.
- **.htaccess**: default filename for the per-directory configuration
  
- (also RE applicable: e.g. <DirectoryMatch regex>)

# Options directive

None	No options.
All	All options except for MultiViews.
ExecCGI	Execution of CGI scripts is permitted.
FollowSymLinks	The server follows symbolic links in the directory. However, the server does not change the pathname used to match against <Directory> sections.
Includes	SSI commands are permitted.
IncludesNOEXEC	A restricted set of SSI commands can be embedded in the SSI pages. The SSI commands that are not allowed are #exec and #include.
Indexes	If a URL that maps to a directory is requested and there is no DirectoryIndex (for example, index.html) in that directory, then the server returns a formatted listing of the directory.
SymLinksIfOwnerMatch	The server only follows symbolic links for which the target file or directory is owned by the same user as the link.
MultiViews	Enables content negotiation based on a document's language.

Options +Setting1 – Setting2

```
<Directory /var/www/>  
Options Indexes FollowSymLinks MultiViews  
AllowOverride None  
Order allow,deny  
allow from all  
</Directory>
```

# Allow-Deny

Order allow,deny  
Allow from all

- First, all Allow directives are evaluated; at least one must match, or the request is rejected. Next, all Deny directives are evaluated. If any matches, the request is rejected. Last, any requests which do not match an Allow or a Deny directive are denied by default.

A domain name, IP, network/netmask (CIDR)

```
# Deny from all [REDACTED]  
# Allow from 127.0.0.0/255.0.0.0 ::1/128
```

# Exercise 2: directory listening

1. Take the previous example
2. Create a directory in your DocumentRoot “myfiles” and put some stuff (try a symbolic link) inside that

```
<Directory /your/dir/myfiles>
```

```
Options +Indexes
```

```
</Directory>
```

3. Create a directory inside “myfiles”:  
“mysecretfiles”:

```
<Directory /your/dir/myfiles/mysecretfiles>
```

```
Options -Indexes
```

```
</Directory>
```

# .htaccess

- Same syntax as the main configuration files
  - so use <Directory> block instead (it's faster!)
  - Common misconception: not specifically for passwords or rewrite!
- “AllowOverride” : Types of directives that are allowed in .htaccess files (None, All, one or more directive inside these groups: {**AuthConfig, FileInfo, Indexes, Limit, Options**})

# Exercise 3: .htaccess password protection

- Let we create a new file with passwords:
  - `htpasswd -c /root/apache_pass cgrl`
  - Then put these directives in .htaccess (or <Directory> )

```
AuthType Basic
AuthName "Restricted Files"
# (Following line optional)
AuthBasicProvider file
AuthUserFile /usr/local/apache/passwd/passwords
Require user rbowen
```

password file

users

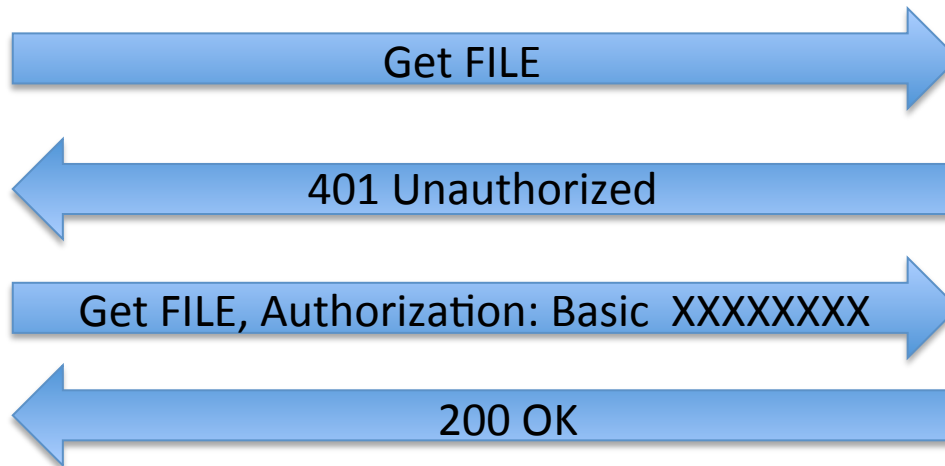
- Now protect our secret file...

# What we did?

- Authentication
    - process by which you verify that someone is who they claim he is
  - Authorization
    - someone is allowed to be where they want to go, or to have information that he wants to have
- Authentication type (see the [AuthType](#) directive)
    - [mod\\_auth\\_basic](#)
    - [mod\\_auth\\_digest](#)
  - Authentication provider (see the [AuthBasicProv](#))
    - [mod\\_authn\\_anon](#)
    - [mod\\_authn\\_dbd](#)
    - [mod\\_authn\\_dbm](#)
    - [mod\\_authn\\_file](#)
    - [mod\\_authnz\\_ldap](#)
    - [mod\\_authn\\_socache](#)
  - Authorization (see the [Require](#) directive)
    - [mod\\_authnz\\_ldap](#)
    - [mod\\_authz\\_dbd](#)
    - [mod\\_authz\\_dbm](#)
    - [mod\\_authz\\_groupfile](#)
    - [mod\\_authz\\_host](#)
    - [mod\\_authz\\_owner](#)
    - [mod\\_authz\\_user](#)



# HTTP Basic Authentication



```
▶ GET /myfiles/mysecretfiles/xxx.txt HTTP/1.1\r\n
```

```
Host: www.mysite.com\r\n
```

```
User-Agent: Links (2.3pre1; Linux 3.0.0-16-generic i686; 126x36)\r\n
```

```
Accept: */*\r\n
```

```
Accept-Encoding: gzip,deflate\r\n
```

```
[truncated] Accept-Charset: us-ascii, ISO-8859-1, ISO-8859-2, ISO-8859-3\r\n
```

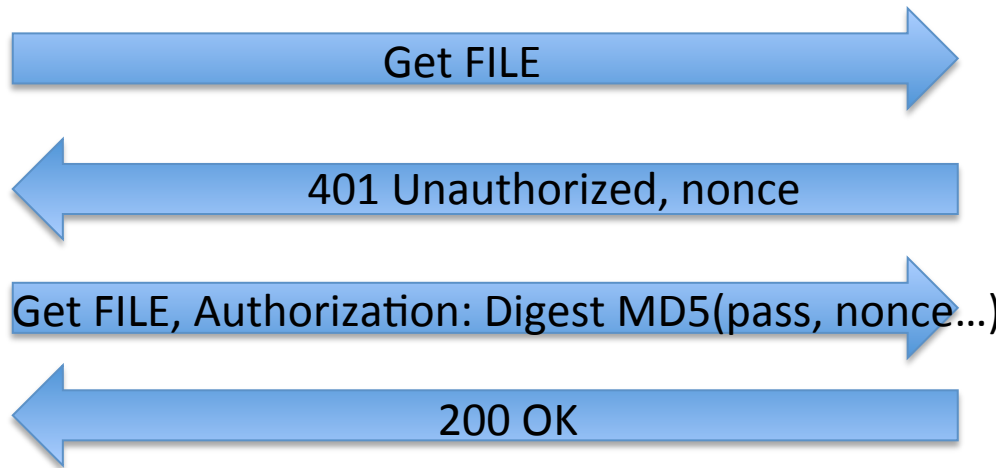
```
Accept-Language: en,*;q=0.1\r\n
```

```
Connection: keep-alive\r\n
```

```
▼ Authorization: Basic Y2dybDpjZ3JscGFzcw==\r\n
```

```
  Credentials: cgrl:cgrlpass
```

# HTTP Digest Authentication



```
GET /dir/index.html HTTP/1.0
Host: localhost
Authorization: Digest username="Mufasa",
                        realm="testrealm@host.com",
                        nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
                        uri="/dir/index.html",
                        qop=auth,
                        nc=00000001,
                        cnonce="0a4f113b",
                        response="6629fae49393a05397450978507c4ef1",
                        opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

# Static web pages



returns the content of a file

# Dynamic web pages



passes the request to a program and responses with the output

client-side scripting

server-side scripting

## Client-side scripting languages

- javascript
- actionscript

DO NOT  
WHATEVER YOU  
WANT

## Server-side scripting languages

- C/C++
- bash (!)
- Perl
- ASP
- PHP
- Java
- Python
- Lua
- Ruby
- ColdFusion
- ...

CGI

DO WHATEVER  
YOU WANT

# Common Gateway Interface

- Standard way to delegate the generation of web pages to executable files
- processes isolated from the core Web server
- check the lab for an example!
  - [www.cgrl.edu/cgi-bin/env.cgi](http://www.cgrl.edu/cgi-bin/env.cgi)
  - [www.cgrl.edu/cgi-bin/interact.cgi](http://www.cgrl.edu/cgi-bin/interact.cgi)

## Apache Tutorial: Dynamic Content with CGI



### Introduction

#### Related Modules

[mod\\_alias](#)

[mod\\_cgi](#)

#### Related Directives

[AddHandler](#)

[Options](#)

[ScriptAlias](#)

<http://httpd.apache.org/docs/2.0/howto/cgi.html>

```
ScriptAlias /cgi-bin/ /root/cgrl-web/cgi-bin/
<Directory "/root/cgrl-web/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    AddHandler cgi-script .cgi
    Order allow,deny
    Allow from all
</Directory>
```

# FastCGI

- *Any program assigned to the handler fcgid-script is processed using the FastCGI protocol*
- *mod\_fcgid starts a sufficient number instances of the program to handle concurrent requests, and these programs remain running to handle further incoming requests.*
- *This is significantly faster than using the default mod\_cgi or mod\_cgid modules to launch the program upon each request.*

# Server Side Include

- are “directives that are placed in HTML pages, and evaluated on the server while the pages are being served.”

Options +Includes

and

```
AddType text/html .shtml
```

```
AddOutputFilter INCLUDES .shtml
```

common directives	examples
include	<code>&lt;!--#include virtual="header.html" --&gt;</code>
exec	<code>&lt;!--#exec cgi="/cgi-bin/foo.cgi" --&gt;</code> <code>&lt;!--#exec cmd="ls -l" --&gt;</code>
echo	<code>&lt;!--#echo var="REMOTE_ADDR" --&gt;</code>
if, elif, else, endif	(control directives)

Installed on more than 20 million Web sites and 1 million web server!

*used by:  
wordpress, joomla  
facebook, flickr  
and many more!*

# PHP



## Warning

We do not recommend using a threaded MPM in production with Apache 2. Use the prefork MPM, which is the default MPM with Apache 2.0 and 2.2. For information on why, read the related FAQ entry on using [Apache2 with a threaded MPM](#)

```
LoadModule php5_module modules/libphp5.so
```

```
<FilesMatch \.php$>
```

```
    SetHandler application/x-httpd-php
```

```
</FilesMatch>
```

*A "handler" is an internal Apache representation of the action to be performed when a file is called*





```

class Person {
    public $firstName;
    public $lastName;

    public function __construct($firstName, $lastName = '') { //Optional parameter
        $this->firstName = $firstName;
        $this->lastName = $lastName;
    }

    public function greet() {
        return "Hello, my name is " . $this->firstName . " " . $this->lastName . ".";
    }

    public static function staticGreet($firstName, $lastName) {
        return "Hello, my name is " . $firstName . " " . $lastName . ".";
    }
}

$he = new Person('John', 'Smith');
$she = new Person('Sally', 'Davis');
$other = new Person('Joe');

echo $he->greet(); // prints "Hello, my name is John Smith."
echo '<br />';
echo $she->greet(); // prints "Hello, my name is Sally Davis."
echo '<br />';
echo $other->greet(); // prints "Hello, my name is Joe ."
echo '<br />';
echo Person::staticGreet('Jane', 'Doe'); // prints "Hello, my name is Jane Doe."

```

# MVC



`mod_passenger` (aka `mod_rails`)  
LoadModule passenger\_module ...  
PassengerRoot ...  
PassengerRuby ...



`mod_wsgi`

WSGI: python standard to communicate with  
web server

```
WSGIScriptAlias / /path/to/mysite.com/mysite/  
wsgi.py
```

# Mod Rewrite

```
DocumentRoot /var/www/example.com
Alias /myapp /opt/myapp-1.2.3
<Directory /opt/myapp-1.2.3>
  RewriteEngine On
  RewriteBase /myapp/
  RewriteRule ^index\.html$ welcome.html
</Directory>
```

- **Goal:** rewrite an URL to another
- **Why?** typical: user friendly URL
- **How?**
  - LoadModule rewrite\_module modules/mod\_rewrite.so
  - AddModule mod\_rewrite.c
  - RewriteEngine on

<http://netgroup.uniroma2.it/index.php?post=258&cat=43422342>



<http://netgroup.uniroma2.it/people/postdoc/marco-bonola/>

# Mod Rewrite

<code>RewriteRule PATTERN SUBSTITUTION [FLAGS]</code>	Define a rule: if find a pattern in the URL, then substitute. Flags: send headers to browsers (e.g. 401)
<code>RewriteCond %{HTTP_USER_AGENT} ^Mozilla.*</code>	Apply the next rule only in this condition is true. Rules are applied only if ALL the previous conditions are true

## Examples of rules:

```
RewriteRule ^/shortcut$ /complicated/and/way/too/long/url/here
```

```
RewriteRule /products/([0-9]+) /siteengine/products.php?id=$1
```

```
RewriteRule
```

```
^/products/([0-9]+),([ad]*),([0-9]{0,3}),([0-9]*),([0-9]*$)
```

```
/test/index.php?id=$1&sort=$2&order=$3&start=$4
```



**KEEP  
CALM  
AND  
CARRY  
ON**

# Mod Rewrite Example: wordpress

```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
# END WordPress
```

(last rule)

(is not regular  
file)

(is not  
directory)

# mod\_rewrite

REGULAR EXPRESSION SYNTAX	RewriteRule FLAGS	Server Variables																																																										
<table border="1"><tr><td>^</td><td>Start of string</td></tr><tr><td>\$</td><td>End of string</td></tr><tr><td>.</td><td>Any single character</td></tr><tr><td>(a b)</td><td>a or b</td></tr><tr><td>(...)</td><td>Group section</td></tr><tr><td>[abc]</td><td>Item in range (a or b or c)</td></tr><tr><td>[^abc]</td><td>Not in range (not a or b or c)</td></tr><tr><td>a?</td><td>Zero or one of a</td></tr><tr><td>a*</td><td>Zero or more of a</td></tr><tr><td>a+</td><td>One or more of a</td></tr><tr><td>a{3}</td><td>Exactly 3 of a</td></tr><tr><td>a{3,}</td><td>3 or more of a</td></tr><tr><td>a{3,6}</td><td>Between 3 and 6 of a</td></tr><tr><td>!(pattern)</td><td>"Not" prefix. Apply rule when URL does not match pattern.</td></tr></table>	^	Start of string	\$	End of string	.	Any single character	(a b)	a or b	(...)	Group section	[abc]	Item in range (a or b or c)	[^abc]	Not in range (not a or b or c)	a?	Zero or one of a	a*	Zero or more of a	a+	One or more of a	a{3}	Exactly 3 of a	a{3,}	3 or more of a	a{3,6}	Between 3 and 6 of a	!(pattern)	"Not" prefix. Apply rule when URL does not match pattern.	<table border="1"><tr><td>R[=code]</td><td>Redirect to new URL, with optional code (see below).</td></tr><tr><td>F</td><td>Forbidden (sends 403 header)</td></tr><tr><td>G</td><td>Gone (no longer exists)</td></tr><tr><td>P</td><td>Proxy</td></tr><tr><td>L</td><td>Last Rule</td></tr><tr><td>N</td><td>Next (i.e. restart rules)</td></tr><tr><td>C</td><td>Chain</td></tr><tr><td>T=mime-type</td><td>Set Mime Type</td></tr><tr><td>NS</td><td>Skip if internal sub-request</td></tr><tr><td>NC</td><td>Case insensitive</td></tr><tr><td>QSA</td><td>Append query string</td></tr><tr><td>NE</td><td>Do not escape output</td></tr><tr><td>PT</td><td>Pass through</td></tr><tr><td>S=x</td><td>Skip next x rules</td></tr><tr><td>E=var:value</td><td>Set environmental variable "var" to "value".</td></tr></table>	R[=code]	Redirect to new URL, with optional code (see below).	F	Forbidden (sends 403 header)	G	Gone (no longer exists)	P	Proxy	L	Last Rule	N	Next (i.e. restart rules)	C	Chain	T=mime-type	Set Mime Type	NS	Skip if internal sub-request	NC	Case insensitive	QSA	Append query string	NE	Do not escape output	PT	Pass through	S=x	Skip next x rules	E=var:value	Set environmental variable "var" to "value".	<p><b>Format</b> %{NAME_OF_VAR}</p> <p><b>HTTP Headers</b> HTTP_USER_AGENT HTTP_REFERER HTTP_COOKIE HTTP_FORWARDED HTTP_HOST HTTP_PROXY_CONNECTION HTTP_ACCEPT</p> <p><b>Request</b> REMOTE_ADDR REMOTE_HOST REMOTE_USER REMOTE_IDENT REQUEST_METHOD SCRIPT_FILENAME PATH_INFO QUERY_STRING AUTH_TYPE</p> <p><b>Server</b> DOCUMENT_ROOT SERVER_ADMIN SERVER_NAME SERVER_ADDR SERVER_PORT SERVER_PROTOCOL SERVER_SOFTWARE</p> <p><b>Time</b> TIME_YEAR TIME_MON TIME_DAY TIME_HOUR TIME_MIN TIME_SEC TIME_WDAY TIME</p> <p><b>Special</b> API_VERSION THE_REQUEST REQUEST_URI REQUEST_FILENAME IS_SUBREQ</p> <p><b>Directives</b> RewriteEngine RewriteOptions RewriteLog RewriteLogLevel RewriteLock RewriteMap RewriteBase RewriteCond RewriteRule</p>
^	Start of string																																																											
\$	End of string																																																											
.	Any single character																																																											
(a b)	a or b																																																											
(...)	Group section																																																											
[abc]	Item in range (a or b or c)																																																											
[^abc]	Not in range (not a or b or c)																																																											
a?	Zero or one of a																																																											
a*	Zero or more of a																																																											
a+	One or more of a																																																											
a{3}	Exactly 3 of a																																																											
a{3,}	3 or more of a																																																											
a{3,6}	Between 3 and 6 of a																																																											
!(pattern)	"Not" prefix. Apply rule when URL does not match pattern.																																																											
R[=code]	Redirect to new URL, with optional code (see below).																																																											
F	Forbidden (sends 403 header)																																																											
G	Gone (no longer exists)																																																											
P	Proxy																																																											
L	Last Rule																																																											
N	Next (i.e. restart rules)																																																											
C	Chain																																																											
T=mime-type	Set Mime Type																																																											
NS	Skip if internal sub-request																																																											
NC	Case insensitive																																																											
QSA	Append query string																																																											
NE	Do not escape output																																																											
PT	Pass through																																																											
S=x	Skip next x rules																																																											
E=var:value	Set environmental variable "var" to "value".																																																											
Redirection Header Codes	RewriteCond FLAGS																																																											
<table border="1"><tr><td>301</td><td>Moved permanently</td></tr><tr><td>302</td><td>Moved temporarily</td></tr><tr><td>403</td><td>Forbidden</td></tr><tr><td>404</td><td>Not Found</td></tr><tr><td>410</td><td>Gone</td></tr></table>	301	Moved permanently	302	Moved temporarily	403	Forbidden	404	Not Found	410	Gone	<table border="1"><tr><td>NC</td><td>Case insensitive</td></tr><tr><td>OR</td><td>Allows a rule to apply if one of a series of conditions are true.</td></tr></table>	NC	Case insensitive	OR	Allows a rule to apply if one of a series of conditions are true.																																													
301	Moved permanently																																																											
302	Moved temporarily																																																											
403	Forbidden																																																											
404	Not Found																																																											
410	Gone																																																											
NC	Case insensitive																																																											
OR	Allows a rule to apply if one of a series of conditions are true.																																																											
Example Rules																																																												
<pre># Site has permanently moved to new domain # domain.com to domain2.com RewriteCond %{HTTP_HOST} ^www.domain.com\$ [NC] RewriteRule ^{.}* \$ http://www.domain2.com/\$1 [R=301,L]  # Page has moved temporarily # domain.com/page.htm to domain.com/new_page.htm RewriteRule ^page.htm\$ new_page.htm [R,NC,L]  # Nice looking URLs (no querystring) # domain.com/category-name-1/ to domain.com/categories.php?name=category-name-1 RewriteRule ^{[A-Za-z0-9-]+}/?\$ categories.php?name=\$1 [L]  # Nice looking URLs (no querystring) with pagination # domain.com/articles/title/5/ to domain.com/article.php?name=title&amp;page=5 RewriteRule ^articles/{[A-Za-z0-9-]+}/([0-9]+)/?\$ article.php?name=\$1&amp;page=\$2 [L]  # Block referrer spam RewriteCond %{HTTP_REFERER} (weight) [NC,OR] RewriteCond %{HTTP_REFERER} (drugs) [NC] RewriteRule .* - [F]</pre>																																																												