The Apache Web Server

Khalid Baheyeldin

Sept 14, 2009

KW Linux Users Group

Agenda

- Introduction
- Web workflow and the HTTP protocol
- Overview
- Installation
- Configuration
- Dynamic content (CGI, FastCGI, embedded)
- Security
- Discussion

About Khalid

2bits

- 25 years in software development and consulting
- Sinclair ZX Spectrum, mainframe, then UNIX since 1987
- Linux discovered 1990, using it regularly since 1995, “LAMP” since 1999
- Open source developer, contributor since 2003
- Full time open source consulting

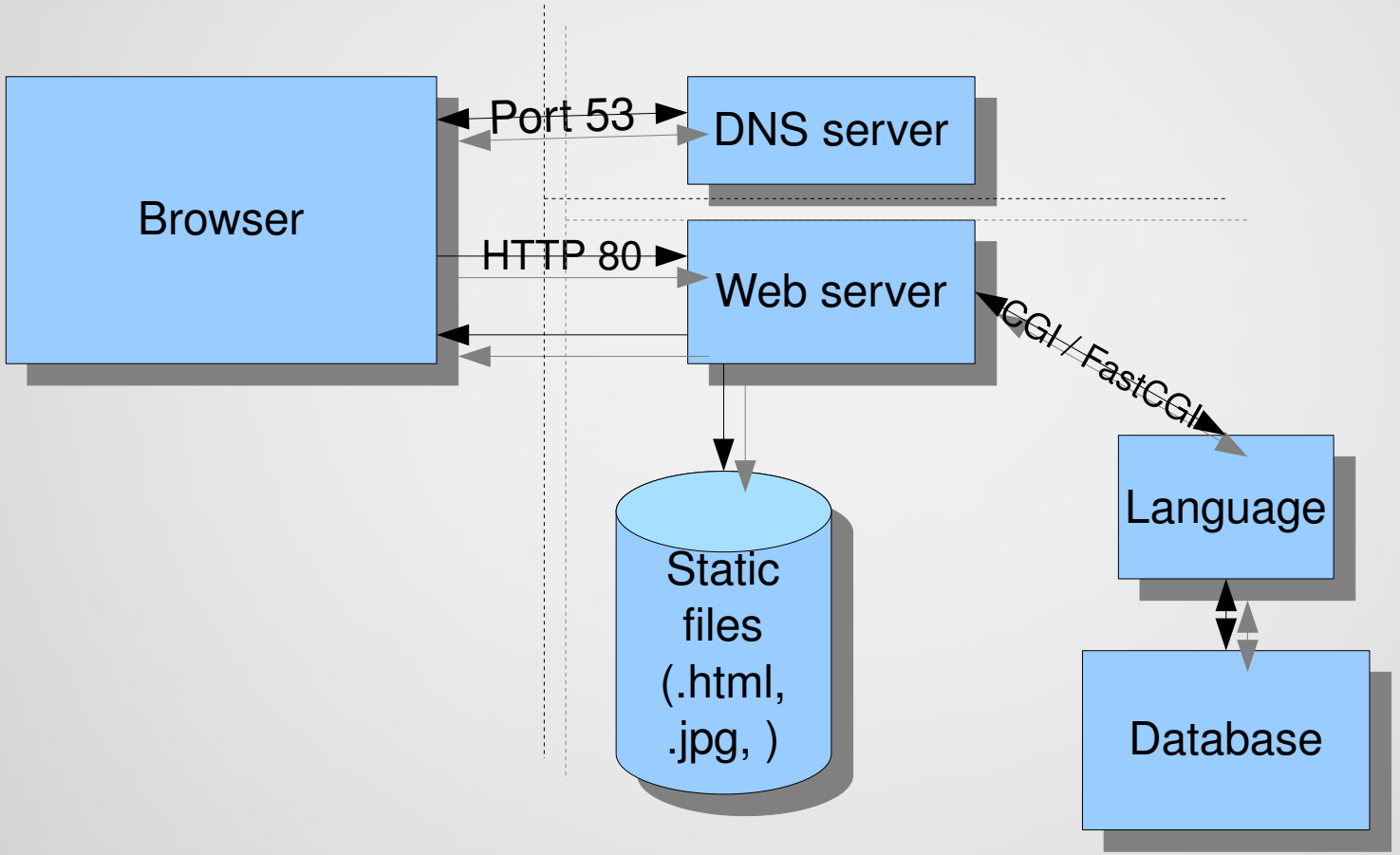
About 2bits.com

- Founded in 1999
- Drupal CMS/CMF since 2003
- Full time consulting
- Services
  - Drupal development
  - LAMP performance optimization and tuning
  - Server provisioning for performance and uptime
  - Manage huge sites for various clients
- http://2bits.com

Web workflow

- A human enters a URL in a browser
- Broken into two parts, host and path
- Host part is looked up in DNS for its IP address
- Then an HTTP request is made to the host with the path passed in
- The HTTP request is received by the web server, and processed
  - Either a static content file is retrieved (/about.html)
  - Or a dynamic request via interpreter, CGI, FastCGI, (/products.php)
- The response is sent back, either good results or an error

Web workflow



HTTP

2bits

- HyperText Transfer Protocol
- Powers the entire web
- Browsers use it to talk to servers, and back
- Connectionless protocol
  - Open, Request, Response, Close, end of story
- Text based

HTTP example

2bits

- Request

```
GET / HTTP/1.1

User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US;
  rv:1.9.0.8) Gecko/2009033100 Ubuntu/9.04 (jaunty)
  Firefox/3.0.8
```

- Headers are colon separated
- An empty line, then any data follows the headers

HTTP example

- Response

```
HTTP/1.0 200 OK
Cache-Control: private, max-age=0
Date: Wed, 08 Jul 2009 02:05:18 GMT
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 22396
Connection: close

<html>
...
</html>
```

Demo

Telnet and netcat!

```
#!/bin/sh

while `nc -lp 1500 -c 'echo "HTTP/1.0 200 OK\nContent-
Type: text/html\n"; cat index.html'`
do
  echo Request sent
done
```

Tip: HTTP Headers

2bits

- Curious about a site?
- Here is how to glean some info:
  - `time wget -S -O /dev/null http://digg.com`
- Shows HTTP Headers
  - Language used
  - Web server

Apache

- Initially based on NCSA HTTPd
- “A patchy server” -> “Apache” (1995)
- Free/Open Source Software
- Most widely used web server (54.32% of active sites, 66.9% of top sites in August 2009 as per Netcraft http://bit.ly/10Ly2o, survey of 226 million hostnames)
- Next most used server is Microsoft IIS (20.05% of active sites, 18.14% of top sites)
- Used to be higher, Microsoft “bought” market share in 2006-2007
- Current stable version 2.2

Apache License

2bits

- “Apache Software License”
- BSD-style license
  - Less strict than the GPL
  - An entity can take the source an close it off
  - Components can be proprietary
- Therefore some Apache modules can be commercial only, or it can be embedded in commercial products with extensions

Popularity

- Among the Top 20 sites, the following use Apache
  - #2 Yahoo.com, #10, Yahoo Japan (unadvertised in server headers)
  - #3 Youtube.com
  - #4 Facebook
  - #7 Wikipedia
  - #9 Baidu (Chinese search engine)

Popularity

2bits

- The rest are
  - Google (own server called GWS, rumoured to be Apache based)
  - Microsoft Live, MSN, Bing (Microsoft IIS)
  - MySpace (Microsoft IIS)
  - QQ (Chinese company, own server)
- Branding is possible (Its Open source, it may still be Apache ...)

Installation

2bits

- Distro binaries
  - Far easier to install and maintain
  - Easier to apply security updates
  - Good for most cases
- Compiling from Source
  - Can customize further
    - e.g. Remove certain modules to save memory
    - Optimize for a given architecture
  - You take ownership of security patches and bug fixes

Install from Source

2bits

- If you insist ...

```
wget http://mirror.csclub.uwaterloo.ca/apache/httpd-
  2.2.X.tar.gz

tar xzvf httpd-2.2.XX.tar.gz

cd httpd-2.2.XX

./configure --prefix=/usr/local/apache2

make

sudo bash

make install
```

Distro install

2bits

- Debian/Ubuntu
  - Minimal install:

    aptitude install apache2
  - Details on 2bits.com at http://bit.ly/j27Kc
- CentOS
  - Yum

Other options

- Ubuntu Server CD has an “install LAMP server” option.
- Ready made “software appliances”
  - Can run in a virtual machine (VirtualBox, VMWare, ...etc.)
  - http://www.turnkeylinux.org/
  - LAMP, MediaWiki, Drupal, Java, and much more

Demo

Let us install Apache ...

Configuration

2bits

- Normally in /etc/httpd.conf
- Debian/Ubuntu: /etc/apache2/* (entire tree structure)
- Can be in other places, e.g. If you compile from source /usr/local/httpd

Core Configuration

2bits

- BindAddress, useful when there are multiple network cards (e.g. 1 for MySQL, 1 for net)
- Listen *port* OR Listen *ip_address:port*
  - Useful if you have more than one network card
- ServerRoot
  - Where the config, log, ...etc are located
- ServerName something.example.com
- DirectoryIndex index.php index.html

Core Configuration

2bits

- User username
- Group groupname
  - These are the Linux user and group to run the server as.
- AddType: map a mime type to an extension
  - **`AddType application/x-httpd-php .php`**
- AddHandler: map a mime type to a handler
  - **`AddHandler fcgid-script .php`**

Tip: HTML as PHP

2bits

- You can have PHP inside .html files, using the MIME type

```
AddHandler php5-script php
AddType text/html php
```

- Beware performance implications: static content is no longer static!

Core Configuration

2bits

- ServerSignature
  - For the footer of error pages
- ServerTokens, level of info to display

```
ServerTokens Prod: Apache
ServerTokens Major: Apache/2
ServerTokens Minor: Apache/2.2
ServerTokens Min: Apache/2.2.8
ServerTokens OS: Apache/2.2.8 (Linux)
ServerTokens Full (default): Apache/2.2.8 (Linux)
  PHP/5.2.6 MyMod/1.2
```

Virtual Hosts

2bits

- Historically
  - One physical server for one host
  - Multiple Apache instances possible, different ports
  - Would people want their site to be on site1.com:81, site2.com:83 ...?
- Vhosts
  - A way to allow the same Apache instance to server multiple hosts
  - Less hardware and software required
  - Pooling of resources (memory, CPU, ...etc.)

Virtual host types

2bits

- Port based virtual hosts
  - Suitable for reverse proxies, e.g. Caching using Squid
- IP based virtual hosts
  - Unique IP address assigned to each host
- Name virtual hosts (most common)
  - IP shared for several hosts (less use of them)
  - Requests directed to vhost depending on host name

VirtualHost Config

2bits

```
<VirtualHost 1.2.3.4>
  ServerName site1.example.com
  DocumentRoot /home/www/site1
</VirtualHost>

<VirtualHost 1.2.3.5>
  ServerName site2.example.com
  DocumentRoot /home/www/site2
</VirtualHost>
```

VirtualHost Config

2bits

```
NameVirtualHost *
<VirtualHost *>
 ServerAdmin webmaster@example.com
 ServerName server1.example.com
 DocumentRoot /var/www/server1
 <Directory /var/www/server1>
  AllowOverride All
  Options MultiViews Indexes Includes FollowSymLinks ExecCgi
  Order allow,deny
  Allow from all
 </Directory>
</VirtualHost>
```

Advanced

- Virtual hosting 1000s of domains? No problem!

```
UseCanonicalName Off # Don't look it up in DNS
VirtualDocumentRoot /www/vhosts/%0/html
```

  - Requests to http://site1.example.com will be mapped to /www/vhosts/site1.example.com/html
- More elaborate example:

```
VirtualDocumentRoot /www/vhosts/%2.1/%3/%1/html
```

  - Requests to http://www.example.com will be mapped to /www/vhosts/e/example.com/site1/html

Demo

2bits

Let us see a vhost!

Directories

- Permissions
  - Which IP addresses
  - HTTP auth (htpasswd)
- Features
  - Show list of files?
  - Will the directory have and index?
  - Will symbolic links be followed?
  - Much more ...

Directory Example

2bits

- An example

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

Redirection

- Common case: want example.com to go to www.example.com, or vice versa

```
<VirtualHost *>
 ServerAdmin webmaster@localhost
 ServerName example.com
 RedirectPermanent / http://www.example.com/
</VirtualHost>
```

ErrorDocument

- You can have custom “page not found” and “access denied” pages

```
ErrorDocument 404 /notfound.html

ErrorDocument 403 /forbidden.html
```

HTTP Auth

- Originally for password protecting directories
  - Utility called htpasswd to create files
  - Hooks for other schemes, e.g. database (MySQL, PostgreSQL), OpenID, PAM, ...
- Still useful with dynamic sites
  - Protects certain paths, admin info, ...etc, not necessarily a physical directory
  - Hide test sites from public view and from crawlers

```
AuthType Basic
AuthName "Restricted Area"
AuthUserFile /etc/htpasswd
Require valid-user
```

.htaccess

2bits

- Per directory overrides, even for same vhost
- Specific things to a site or to an application
  - Rewrite rules: e.g. Drupal clean URLs
  - Custom error response (404, 403, ...)
  - Authentication (password protection)
- Most things that go in Apache's configuration can go in .htaccess, if the admin allows overrides
- Can have a performance impact

Logging

- Error logging
- Access logging

Error Logging

2bits

- For errors, administrator alerts, ...etc.
- Also for debugging
- Tip: For eAccelerator instability, had a script to check the error log every 45 seconds and restart Apache when a segmentation fault was encountered

```
ErrorLog /var/log/apache2/error.log
# debug, info, notice, warn, error, crit, alert, emerg.
LogLevel warn
```

Access Logging

2bits

- For logging accesses from site visitors
- Vital information (bots, crawlers, leechers, ...etc.)
- Also used for statistics and reporting(e.g. Awstats, ...)
- Make sure you rotate them, since they can grow
  - Put them on a separate disk for large sites
  - Or just filter what gets logged (e.g. Ignore images)

Access Logging

2bits

- What is logged? Configurable, most commonly:
  - Visitor IP address, host, HTTP Auth name, date/time/timezone, Operation, URL, HTTP code, bytes, referer, User Agent
- Can be one file for all sites (less common)

```
CustomLog /var/log/apache2/access.log combined
```

- Or per site (in each vhost)

```
CustomLog /var/log/apache2/access-example.com.log combined
```

Modules

- Apache has a modular design, meaning you can extend it in any way you imagine
- Can be static (complied in) or shared (pluggable, implemented as a dynamic library)
- Dynamic Shared Objects (DSO)
- Modules do various specific things

Modules

2bits

- Free or commercial, generic or specific
- Can be disabled/enabled, reduce size
- Normally configured like this:
  - Static modules

```
AddModule rewrite_module /usr/lib/.../mod_rewrite.so
```

  - Dynamic modules (DSO)

```
LoadModule rewrite_module /usr/lib/.../mod_rewrite.so
```

Modules categories

2bits

- Authentication (MySQL, Postgres, Kerberos, ...)
- Logging (to SQL databases)
- Languages (Python, PHP, Perl, Java, Ruby, Mono, Neko, ...)
- Miscellany
  - GeoIP
  - Subversion version control
  - Evasive: anti-DoS/DDoS measures

mod_rewrite

2bits

- manipulate URLs in various ways
- e.g. Prettier URLs
- Before `example.com/index.php?q=something`
- After `example.com/something`
- Can be used for mapping old content to new

Drupal “clean URLs”

2bits

- More friendly, for humans and search engines
- Query String Append, Last rule

```
<IfModule mod_rewrite.c>
  RewriteEngine on
  RewriteBase /
  # Rewrite URLs of the form '/x' to the form '/index.php?q=x'.
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule ^(.*)$ index.php?q=$1 [L,QSA]
</IfModule>
```

Drupal boost cache

-

```
<FilesMatch "\.(html)$">
 <IfModule mod_headers.c>
  Header set Expires "Sun, 19 Nov 1978 05:00:00 GMT"
  Header set Cache-Control "no-store, no-cache, must-revalidate,
   post-check=0, pre-check=0"
 </IfModule>
</FilesMatch>
```

Drupal boost cache

2bits

```
# Only handle GET requests

RewriteCond %{REQUEST_METHOD} !^GET$ [OR]

# Exclude certain paths that will always be dynamic

RewriteCond %{REQUEST_URI}
   ^(/admin|/cache|/misc|/modules|/sites|/system|/themes|/user/lo
   gin) [OR]

# If the user is logged in

RewriteCond %{HTTP_COOKIE} DRUPAL_UID

# Skip next two rules (pseudo if/else)

RewriteRule .* - [S=2]

RewriteCond %{DOCUMENT_ROOT}/cache/%{SERVER_NAME}%{REQUEST_URI}_%
   {QUERY_STRING}.html -f

RewriteRule .* cache/%{SERVER_NAME}%{REQUEST_URI}_%
   {QUERY_STRING}.html [L]
```

mod_fcgid

2bits

- For FastCGI (e.g. PHP, Python)
- Default configuration is:

```
LoadModule fcgid_module /usr/lib/apache2/modules/mod_fcgid.so
<IfModule mod_fcgid.c>
  AddHandler fcgid-script .fcgi
  IPCConnectTimeout 20
</IfModule>
```

mod_status

- ExtendedStatus On or Off
- Provides statistics and information
- Can be parsed by other applications, e.g. Munin to give things like:
  - Number of Apache processes
  - Number of requests per second
  - Number of bytes per second
- http://www.apache.org/server-status
- http://www.apache.org/server-status?auto

Other modules

2bits

- mod_deflate
  - compresses web pages, css, js to save bandwidth
- mod_include
  - Server Side Includes (SSI) for .shtml, allows some primitive dynamism (by today's standards)

The Debian Way

2bits

- On Debian/Ubuntu
- Provides commands to enable/disable modules and sites
- No need to much with configuration files (most of the time)
  - a2enmod, a2dismod
  - a2ensite, a2dissite

Apache modes

2bits

- MPM Pre Fork
- MPM Worker

Apache pre-fork

2bits

- Each request is handled by a separate process
- Apache pre-forks a configurable number of processes, leaving spares around
- Incoming requests always have a handler for them, as long as the maximum is not reached
- Most common for PHP on VPS or dedicated, shared hosting more often CGI
- Configuration is important
  - Too low and you can get users complaining!
  - Too high and you get the server swapping!

Pre-fork configuration

2bits

- Parameters for many things

```
StartServers              5
MinSpareServers           5
MaxSpareServers          10
MaxClients              150
MaxRequestsPerChild       0
```

Apache MPM Worker

2bits

- Threaded mode
- Each request is handled by a thread
- Much less memory usage
- Does not work with certain setups (e.g. PHP with mod_php)

MaxClients

2bits

- *“For [...] prefork, MaxClients translates into the maximum number of child processes that will be launched to serve requests. [...]”*
- *“For threaded [...] servers [...] MaxClients restricts the total number of threads that will be available to serve clients. [...] the default value is 16 (ServerLimit) multiplied by the value of 25 (ThreadsPerChild). [...]”*

Worker configuration

2bits

- Parameters for:

```
<IfModule mpm_worker_module>
  ServerLimit             600
  StartServers             10
  ThreadsPerChild          10
  MaxClients              600
  MinSpareThreads          30
  MaxSpareThreads          50
  MaxRequestsPerChild    3000
</IfModule>
```

Tip: Worker + FastCGI

2bits

- Using Worker (threaded server) and FastCGI has the potential for memory savings

Interpreter Modes

2bits

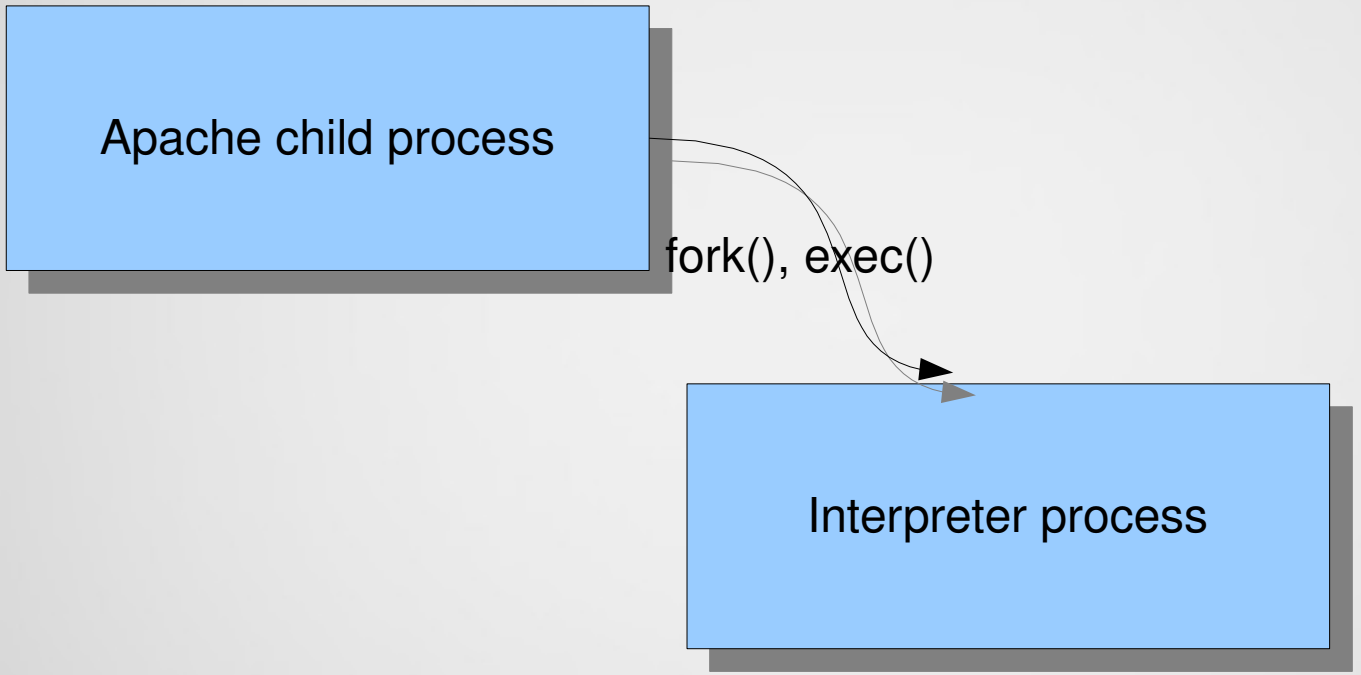
- Dynamic sites use Apache as the web server, in conjunction with a scripting language (e.g. PHP, Python, Perl)
- Several modes
  - CGI
  - Embedded
  - FastCGI

Common Gateway Interface

2bits

- Oldest method of running dynamic stuff inside a a web site (early to mid 90s?)
- Workflow:
  - Request comes in e.g `/cgi-bin/something.cgi?param=value`
  - Web server forks a new process (fork() then exec())
  - New process executes an interpreter (e.g. Perl)
  - A script is executed by the interpreter, with parameters passed
  - Passing back the result
  - Terminate the process

CGI



CGI (cont'd)

2bits

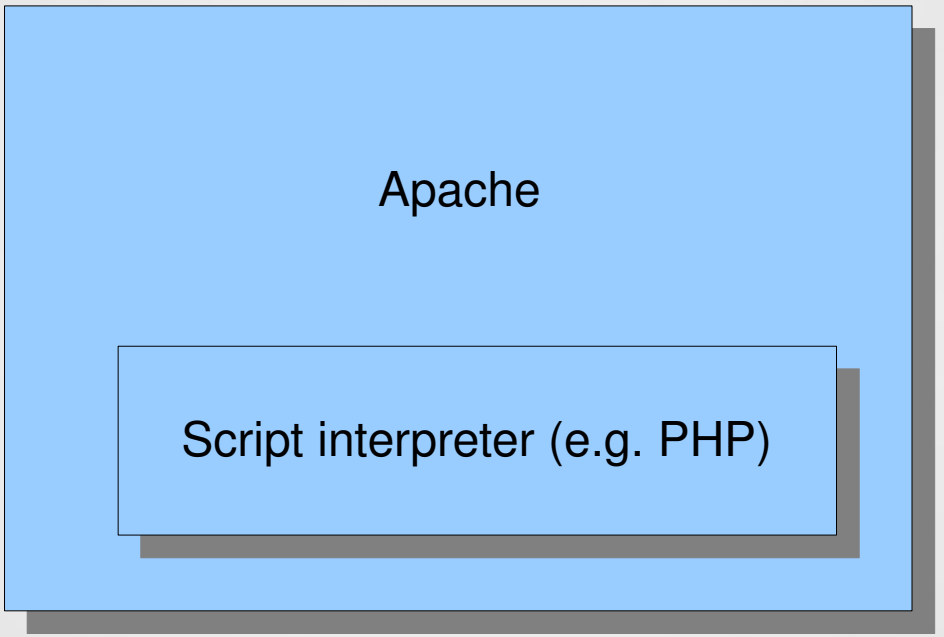
- Advantages
  - Simple, returns results as standard output
  - Provided a way to create dynamic web sites
  - Security (if setup correctly, interpreter process can run other than the web server user, with limited privileges)
- Drawbacks
  - Slow (because of the fork system call)
  - Wasteful on resources (fork is expensive)
  - Not suitable for large web sites (does not scale)

Embedded (modules)

2bits

- Interpreter inside the Apache process
- PHP (mod_php)
- Perl (mod_perl)
- Python (mod_python)

Embedded

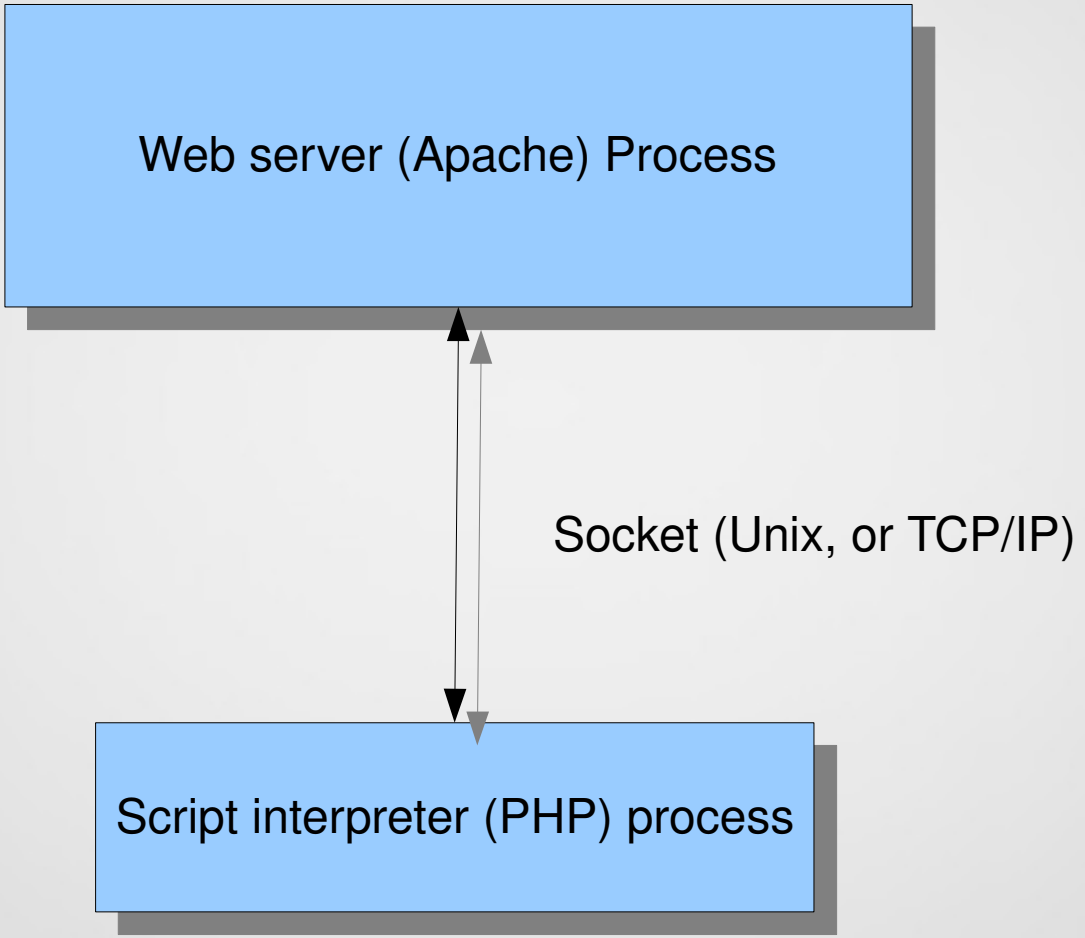


FastCGI

2bits

- Similar to CGI
- Does not fork per request
- Instead, uses a socket to communicate with an existing process that holds the interpreter
- Instead of Apache being bloated with 100s of processes, we have fewer dynamic processes, a small pool to handle dynamic requests
- Apache has mod-fastcgi (questionable stability) and mod-fcgid (much more stable)
- Best of both worlds!

FastCGI


Web server (Apache) Process
Socket (Unix, or TCP/IP)
Script interpreter (PHP) process

FastCGI (fcgid)

2bits

- Advantages
  - Significant memory savings!
  - Speed is very close to mod_php
  - Can be used with threaded server (MPM Worker)
  - Less connections to the database
- Drawbacks
  - Watch the timeouts (e.g. Large site with Drupal and cron)

Performance

2bits

- MaxClients
- .htaccess on a deep directory structure
- Disable hostname lookups (HostnameLookups)
- Don't log bytes separately
- Use MPM Worker threaded server vs MPM prefork if your site is dynamic (memory saving)
- Rotate your logs
- KeepAliveTimeout only a few seconds
- More reading
  - http://httpd.apache.org/docs/2.2/misc/perf-tuning.html

Security

2bits

- Check directory permissions (enable writing only for uploaded images, ...etc., NOT scripts)
- Don't enable mod_proxy for a publicly visible server
- Consider Apache's mod_security (web application firewall)

Resources

- Configuration reference
  - http://httpd.apache.org/docs/2.2/
- Developer reference
  - http://httpd.apache.org/docs/2.2/developer/

Conclusion

2bits

- Apache is:
  - Capable
  - Proven
  - Feature rich
- Use MPM Worker + FastCGI if possible
- Install only what you need (be a minimalist)

Future topics?

2bits

- Any interest in the following topics?
  - Introduction to PHP: a web development language
  - Introduction to MySQL database

Discussion

2bits

Questions?

Comments?