

APACHE httpd SERVER

INTRODUCTION:

Apache is an open source HTTP web server. It handles HTTP Requests sent to it and then it is able to them. Apache has evolved with the years of the internet. Server is used to support both static and dynamic pages online. Many programming languages are supported by the Apache Server are as follows: PHP, Perl, Python and alongside with MySQL. Apache is comprised of two main building Blocks with the Latter being comprised of many other little building blocks. The Building Blocks are the Apache Core and then the Apache Modules that in a sense extend the Apache core. It is very easy to implement and very easy to add extend its abilities by the adding of different modules.

Apache Server Architecture:

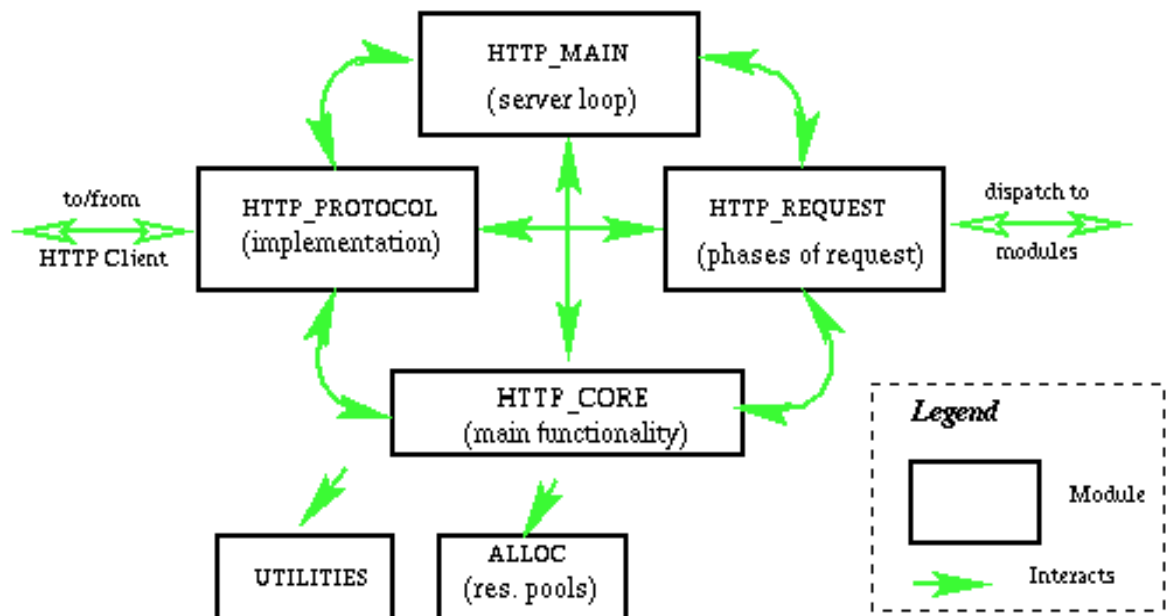
Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Popular compression methods on Apache include the external extension module, `mod_gzip`, implemented to help with reduction of the size (weight) of web pages served over HTTP. The core of a Apache Server implements the basic functionality of the server. In addition it implements a number of utility functions.

The core components of make up the Apache core are as follows:

- o `http_protocol.c`: This is the component that handles all of the routines that communicate directly with the client by using the HTTP protocol. This is the component that knows how to also handle the socket connections through which the client connects to the server. All data transfer is done through this component.
- o `http_main.c`: this component is responsible for the startup of the server and contains the main server loop that waits for and accepts connections. It is also in charge of managing timeouts.
- o `http_request.c`: This component handles the flow of request processing, passing control to the modules as needed in the right order. It is also in charge of error handling.
- o `http_core.c`: The component implementing the most basic functionality, it just bairly serves documents.

- o alloc.c: the component that takes care of allocating resource pools, and keeping track of them.
- o http_config.c : this component provides functions for other utilities, including reading configuration files and managing the information gathered from those files (), as well as support for virtual hosts. An important function of http_config is that it forms the list of modules that will be called to service during different phases of the requests that are going on within the server.

Apache Overview Diagram:



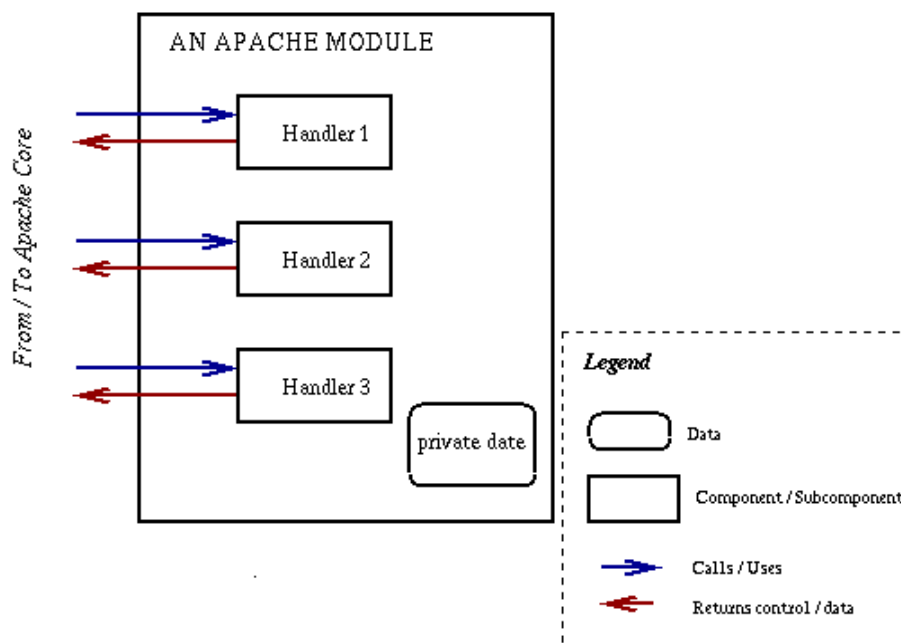
How Requests are Handled ?

- URI to filename translation.
- Check access based on host address, and other available information.
- Get an user id from the HTTP request and validate it
- Authorize the user.
- Determine the MIME type of the requested object (the content type, the encoding and the language).
- Fix-ups (for example replace aliases by the actual path).
- Send the actual data back to the client.
- Log the request.

The concept of a Handler:

- A handler is for Apache the action that must be performed in some phase of servicing the request
- They are defined by modules. A module might specify handlers for one, many or none of the phases of the request.
- If you are using a static configuration of Apache, choose the modules you wish to incorporate with care. Using static mode comes at a price — the more modules, the more memory you use. Thus, a forked multi-processing module can have a significant effect on the machine's memory requirements.
- Note that some items are automatically included, so you'll need to explicitly enable and disable needed modules. Also remember to include any third-party modules (e.g., authentication, PHP, or mod_perl), the Web service requires. Use configure --help to get a list of the available options.

Architecture of an Apache Module:



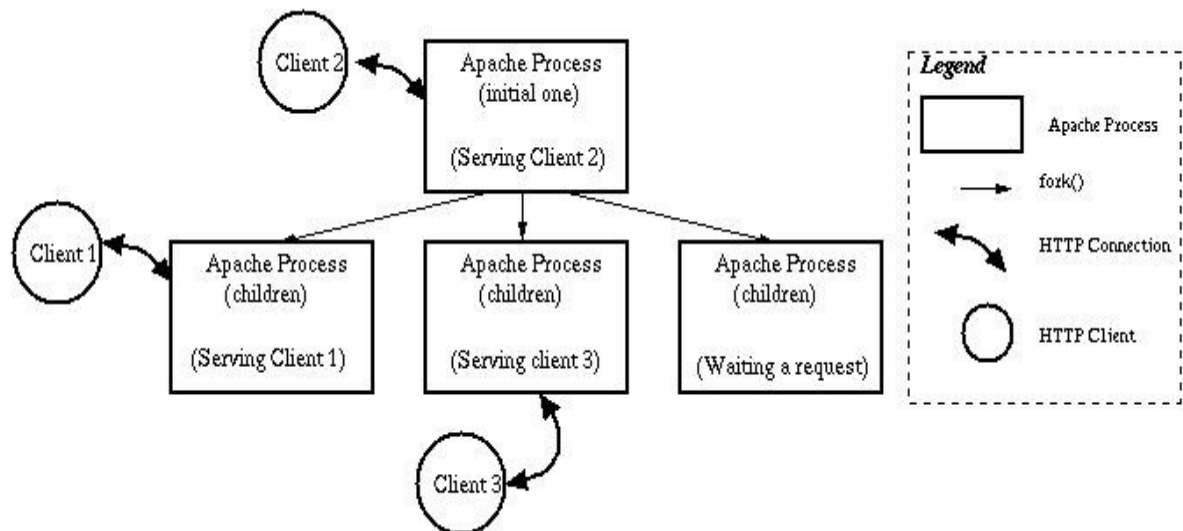
- Overview of the Handler system within an Apache Module. As you can see the Handler does what it needs to do to fulfill a request then the sends that process back to the HTTP_REQUEST component of the Apache core in order to be sent to another module for processing or back to the client.

Persistent Server Processes:

- 'Persistent Server Processes' is a concept that explains the implementation of Concurrency in Apache.

- o To handle large incoming requests to website TCP/IP servers fork a new child to handle new incoming request from clients.
- o However in the situation of a busy web site the overhead of forking a huge number of children will simply suffocate the machine.
- o To solve this problem Apache uses Persistent Server Process.
- o It forks a fixed number of children, right from the beginning. The children service incoming requests independently (different address spaces).
- o It is interesting that Apache server can dynamically control the number of children it forks (i.e. increasing or decreasing it), based on current load.

Concurrency in Apache:



Configure Apache Web Server :

- We have to first configure the DNS. Then configure the following file `/etc/httpd/conf/httpd.conf`.
- The root directory of Web server is `/etc/httpd`, which is divided into three parts.
- The three parts are:
 - `/etc/httpd/conf` (where configuration files stays)
 - `/etc/httpd/logs` (where the logs of Web server and site accessing stay)
 - `/etc/httpd/modules` (where the module stays, which enables the server side programmer to do programming in the languages supported by Web server)

Specific functional requirements:

- Availability and Reliability
- Reusability
- Security
- Maintainability
- Portability
- Performance

Apache Usage:

- Apache is primarily used to serve both static content and dynamic Web pages on the World Wide Web. Many web applications are designed expecting the environment and features that Apache provides.
- Apache is the web server component of the popular LAMP web server application stack, alongside MySQL, and the PHP/Perl/Python programming languages.

BENEFITS:

- o Function wealthy
- o Easy to customize
- o Easy Management
- o Extensible
- o OPERATING SYSTEM Impartial

Conclusion:

- o Apache Web Server has a modular architecture. The core provides the basic functionality and separate set of supporting modules for handling HTTP requests.
- o Concurrency exists only between persistent identical processes that service incoming HTTP requests on the same port.
- o Functionality is easily changed by writing new or editing existing modules.

