8.6. Rewriting Examples

Next, we will provide several rewriting examples.

To turn on the rewrite engine, use:

```
RewriteEngine on
```

To force a redirect to a new location for a document, use:

```
RewriteRule ^/old.html$ new.html [R]
```

To rewrite a **URI** based on the web browser (without a redirect), use:

```
RewriteCond %HTTP_USER_AGENT .*Chrome.*
```

```
RewriteRule ^/foo.html$ /chrome.html [L]
```

To create a human-readable **URI** from a **CGI** script (Note the **PT** or **Passthrough** flag: without it, the rewrite is treated as a file, not a **URI**), use:

```
RewriteRule ^/script/(.*) /cgi-bin/script.cgi?$1 [L,PT]
```

The **mod_alias** command maps **URIs** to filesystem paths outside of normal **DocumentRoot.** It also provides **Alias** and **Redirect** directives, **Apache** access control considerations, file permission considerations, and **SELinux** considerations.

Mod_alias is much simpler than **mod_rewrite**. A redirect directive sends a redirect that is similar to the **R** flag for **mod_rewrite**.

To serve the files located at **/newdocroot/** under the **URI /new/** (Note the **<Directory>** stanza for access control modifications), use:

```
Alias /new/ /newdocroot/
<Directory /newdocroot/>
     Options Indexes FollowSymLinks
     AllowOverride None
     Require all granted
</Directory>
```

Care must be taken to properly set the **SELinux**, **Apache** access controls, and filesystem permissions to allow **Apache** to serve content from the directory if it is not inside the default document root.

8.8. AliasMatch

AliasMatch is like the **Alias** directive with regular expressions. It is more powerful than **Alias**, but not as powerful as a rewrite.

To serve the files located at `/newdocroot/<SUBDIR>` under the **URI** `/new/<SUBDIR>` with different permissions for each use:

```
AliasMatch ^{}/new/(.*) /newdocroot/$1/
<Directory /newdocroot/foo/>
     Options Indexes FollowSymLinks
     AllowOverride None
     Require all granted
</Directory>
<Directory /newdocroot/bar/>
     Options -Indexes FollowSymLinks
     Require all granted
</Directory>
...
```

8.9. AliasMatch Considerations

AliasMatch is not a direct replacement for **Alias**. Because **AliasMatch** uses a regular expression, `Alias /new/ /newdocroot/` is not the same as `AliasMatch /new/ /newdocroot/`.

The latter will send all **URIs** which contain the string `/new/` anywhere to `/newdocroot/`.

The proper way to replace **Alias** with **AliasMatch** is:

```
AliasMatch ^/new/(.*)$ /newdocroot/$1
```

CGI scripts should not be in your document root. There are two ways to run scripts outside your document root:

- Create an alias and set the handler to **`cgi-script`**.
- Use **`ScriptAlias`**.

Use the same considerations as **`Alias`** and **`AliasMatch`**:

- **Apache** access control considerations.
- File permission considerations.
- **SELinux** considerations.

Because of the nature of **CGI** scripts, they should not be directly served as plain files.

`ScriptAlias` will set up an alias and enable the **cgi-script** handler. An example of this is:

```
ScriptAlias /newscripts/ /newscripts-docroot/
<Directory /newscripts-docroot/>
     Options Indexes FollowSymLinks
     AllowOverride None
```

Mod_Status has the following characteristics:

- **Apache** server status.
- Human-readable page.
- Machine-readable page.
- Security considerations with access control and extended status.

To enable **mod_status**, use:

```
<Location /server-status>
     SetHandler server-status
     Require ip 10.100.0.0/24
</Location>
```

To get the machine-readable version of the status page, append **auto** to the **URI** you have defined above.

Because of the nature of this information, it should be restricted to trusted hosts or networks.

Apache version 2.3.6 changed the default for the **ExtendedStatus** directive to be on.

Mod_Include provides filtering before a response is sent to the client. It is enabled with the **+Includes** directive. By default, this is enabled only for ***.shtml** files, but it can be enabled for all ***.html** files with the **XBitHack** directive.

To enable **includes** use:

```
<Location /dynamic/>
     Options +Includes
     XBitHack on
</Location>
```

An **HTML** file which had includes would contain:

```
<!--#set var="dude" value="very yes" -->
...
<!--#echo var="dude" -->
...
<!--#include virtual="http://localhost/includes/foo.html" -->
```

When you enable the **XBitHack** directive, **Apache** only looks for **includes** where the **execute** permission is set on the file.

Mod_Perl is a **Perl** interpreter embedded in **Apache.** It can provide full access to the entire **Apache API,** sharing information between **Apache** processes. **CGI** scripts can run unmodified.

Some of the downsides to using **Mod_Perl** are:

- Some **CPAN** modules are not thread-safe, therefore they do not work with the worker **MPM** (**M**ulti-**P**rocessor **M**odule).
- Some **Perl** methods like **die()** and **exit()** can cause performance issues.

To enable **mod_perl** use:

```
Alias /perl /var/www/perl
<Directory /var/www/perl>
     SetHandler perl-script
     PerlResponseHandler ModPerl::Registry
     PerlOptions +ParseHeaders
     Options +ExecCGI
</Directory>
```

For a sample **Perl** script using **mod_perl** use:

```
Alias /perl /var/www/perl
<Directory /var/www/perl>
     SetHandler perl-script
     PerlResponseHandler ModPerl::Registry
     PerlOptions +ParseHeaders
     Options +ExecCGI
</Directory>
```

For a sample **Perl** script using **mod_perl** use:

```
print "Content-type: text/plain \n";
print "mod_perl 2.0 rocks! \n";
```

8.14. MultiProcessing Modules (MPMs)

To allow for different performance characteristics, different multiprocessing modules can be used.

The **Prefork MPM** is the default **MPM** on most **Linux** systems. **Prefork** is non-threaded, and is thus safe for libraries and scripts which are not thread-safe. Because each request is handled by a separate process, the **Prefork MPM** uses more memory and resources to handle the same number of connections as the **Worker MPM**.

The **Worker MPM** is a hybrid multi-process, multi-threaded model. Each sub-process spawns threads which handle the incoming connections. Because threads are lighter on system resources than processes, this module has better performance with less resource utilization than the **Prefork MPM**.

The **Event MPM** is based on the **Worker MPM** but it off-loads some of the request handling to shared processes, thus increasing performance yet again.

The **Prefork MPM** is controlled by the following directives:

- **`StartServers`**: Number of child server processes to create on startup.
- **`MinSpareServers`**: Minimum number of child server processes to keep on hand to serve incoming requests.
- **`MaxSpareServers`**: Maximum number of child server processes to keep on hand to serve incoming requests. Any spare processes above this number are killed.
- **`MaxRequestsPerChild`**: Maximum number of requests a child server process serves.
- **`MaxClients`**: Maximum number of simultaneous connections to serve. To increase this number, you may also have to increase the **`ServerLimit`** directive.

8.2. Learning Objectives

By the end of this session, you should be able to:

- **Explore some of the more advanced configurations of Apache, using:**
  - **Rewrite module.**
  - **Status module.**
  - **Alias module.**
  - **Include module.**

8.3. Why Rewrite URLs?

URL rewriting is a powerful way to modify the request that an **Apache** process receives. Modifications can include moved documents, forcing **SSL**, forcing login, human-readable **URIs**, and redirecting based on the browser.

Because of the power behind **mod_rewrite**, rewriting URLs can be complex. The **Apache** documentation is a big help: http://httpd.apache.org/docs/current/rewrite/.

Rewrite rules can exist in the **.htaccess** files, the main configuration file, or **<Directory >** stanzas. The **.htaccess** files seem the most common, but the main configuration file may be more secure.

The **mod_rewrite** uses **PCRE** (**P**erl **C**ompatible **R**egular **E**xpressions). This provides nearly unlimited conditions or filters to include, like:

- Time of day.
- Environment variables.
- **URLs** or query strings.
- **HTTP** headers.
- External scripts.

8.5. Rewrite Map

Use a rewrite map for advanced or dynamic rewrite rules. The **Apache** documentation covers each of them in detail: http://httpd.apache.org/docs/current/rewrite/rewritemap.html.

Pre-compiled rewrite maps exist for specific purposes. To learn more, see the **WURFL** (wireless universal resource file) map: http://wurfl.sourceforge.net/.

Rewrite maps are used when rewrite rules are too complex or numerous. There are many different types of maps:

- **txt**: Flat text.
- **dbm**: **DBM** hash (indexed).
- **rnd**: Randomized plain text.
- **prg**: External program.

8.16. Configuring Worker

The **Worker MPM** is controlled by the following directives:

- **`StartServers`**: Number of child server processes to create on startup.
- **`MaxClients`**: Maximum number of simultaneous connections to serve.
- **`MinSpareThreads`**: Minimum number of worker threads to keep on hand to serve incoming requests.
- **`MaxSpareThreads`**: Maximum number of worker threads to keep on hand to serve incoming requests.
- **`ThreadsPerChild`**: Constant number of worker threads in each child server process.
- **`MaxRequestsPerChild`**: Maximum number of requests a server process serves.

Apachebench (**ab**) is the tool provided as a part of the **Apache httpd** server package.

The **httperf** tool was written at **HP Labs**. To learn more about it, see http://www.hpl.hp.com/research/linux/httperf/. **Httperf** can use log files as a source of **URIs** to test.

Information about **siege**, a third-party benchmark tool, can be found at http://www.joedog.org/siege-home/.

A tool named **sproxy** works as an **HTTP** proxy server, collecting all the information and **URIs** to use as the **Siege** testing corpus.

The best testing results come from using **URIs** which match the real-life workflow. Some of these tools can use log files directly, others collect information in other ways. Having a proper testing corpus is almost as important as the settings you use.

Apache can act as a forward or reverse proxy that is managed by **mod_proxy** and **mod_proxy_http**.

Debugging headers include:

X-Forwarded-For

X-Forwarded-Host

X-Forwarded-Server

To enable a reverse proxy to an internal server, use:

```
<Location /foo>
    ProxyPass http://appserver1.example.com/foo
</Location>
```

You can also use the following alternative syntax:

```
ProxyPass /foo http://appserver1.example.com/foo
```

If the first argument to **ProxyPass** has a trailing **/**, then the second should have as well.

Apache can also cache requests it proxies, using

Apache can also cache requests it proxies, using **`mod_cache`**.

There are special-purpose tools which do caching much more efficiently:

- **Varnish**: a **RAM** and disk-based cache which uses the **Linux** kernel's swapping mechanism to speed up caching.
- **Squid**: A general-purpose caching proxy.
- Specialty appliances.

Specialty **HTTP** servers have been developed to deal with different issues.

The **C10k Problem** describes the issues in making a web server which can scale to ten thousand concurrent connections/requests.

Many new, lightweight web server systems have been written with the **C10k** problem in mind:

- **cherokee**: http://www.cherokee-project.com/
  cherokee has an innovative web-based configuration panel. Benchmarks have shown **cherokee** to be much faster than **Apache** for dynamic and static content.
- **nginx**: http://wiki.nginx.org/Main
  The **nginx** server has a small footprint, so it scales well from small servers to high performance web servers. **nginx** powers some of the largest sites on the **Internet**, and now accounts for 14% of the web server market share. To learn more, see: http://news.netcraft.com/archives/2016/02/22/february-2016-web-server-survey.html
- **Lighttpd**
  Lighttpd is used to power some high-profile sites, such as **YouTube** and **Wikipedia**. **Lighttpd** has a light-weight and scalable architecture as well.