

Apache httpd 2.2



Lukáš Zapletal
lukas.zapletal@liberix.cz

Co je to Apache httpd?

- Apache httpd (nebo někdy jen httpd) je kvalitní, robustní, modulární webový server dostupný zdarma
- vyhovuje standardu HTTP/1.1
- existuje snad pro všechny operační systémy (a také pro Windows)
- tvoří jej nezisková organizace Apache Software Foundation (ASF)

Kde ho sehnat

- stránky www.apache.org
(konkrétně httpd.apache.org)
- je ve všech linuxových distribucích
- pro Windows je dostupný exe instalátor
„Next. Next. Next. Finish. Restart.“
- instalace v Linuxu: triviální (balíčky)
jmenují se `apache2` nebo `httpd2`
- můžete také stáhnout zdrojové kódy a
zkompilovat vlastní (odlehčenou) verzi

Mám ho. Jak to spustit?

- v UNIXu standardní démon;
ve Windows běží jako služba
- tedy `/etc/init.d/httpd start`
nebo `/etc/init.d/apache2 start`
(záleží na distribuci)
- nebo také můžete použít program
`apachectl`:
`apachectl start (stop | restart | reload ...)`
- skripty v `init.d` volají právě `apachectl`

Bezpečnost

- jedná se o veřejně dostupnou službu dostupnou po síti - je nutné být maximálně opatrní
- Apache se spouští pod superuživatelem (root), protože musí navázat port 80
- poté přepne efektivního uživatele na účet s nižšími právy (obvykle apache, httpd nebo www-run)
- pod tímto účtem dále pracuje - je nutno to mít na paměti

Jak Apache pracuje

- jelikož funguje pod mnoha systémy (vyžadující si různý přístup ke konkurečnímu zpracování požadavků), tato jeho část je modulární (MPM: Multi-Processing Modules)
- unixová verze nabízí několik modulů (prefork, worker, leader, perchild)
- ostatní systémy (Windows, Netware, OS/2) mají obvykle jednu implementaci (jeden proces, více vláken)

MPM: prefork

- nejpoužívanější (a historický) přístup k vyřizování požadavků
- jeden hlavní proces, ostatní podřízené
- MinSpareServers, MaxSpareServers: kolik procesů se může maximálně/minimálně „flákat“ (idle)
- StartServers: kolik jich má spustit při startu
- MaxClients: maximum procesů, které server může spustit (hranice)

MPM: prefork

- prefork nepoužívá vlákna (nemusí, procesů běží víc a každý vyřídí jeden požadavek)
- postupně se zvyšuje nebo snižuje počet procesů podle zátěže
- prefork je někdy jediná možnost, pokud používáte knihovny, které nejsou vláknově bezpečné (PHP...)
- chyba (pád) jednoho požadavku nevyřídí jiné požadavky (spadne jen jeden)

MPM: worker

- hybridní implementace, více procesů, každý proces má více vláken
- obvykle výkonnější než prefork, ale pád jednoho procesu může „zrušit“ jiné (nevinné) požadavky
- nastavení je podobné
- ThreadsPerChild: počet vláken na proces (statická hodnota)
- při zátěži se nezvyšuje počet vláken, ale počet procesů

MPM: další moduly

- perchild: hybridní modul, podobný workeru, ale pracuje s proměnlivým počtem vláken
z Apache httpd 2.2 byl odstraněn (není ještě hotov)
- event: podobný workeru, ale předává požadavky ve stavu KEEP ALIVE tomu určenému vlákně, pracovní vlákno se tím uvolňuje pro další zpracování

Ukázka: první spuštění

- implicitní nastavení je obvykle jednoduché a bezpečné
- server je nastaven obvykle na podávání stránek z adresáře `/var/www/htdocs`
- logování nastaveno do adresáře `/var/log/httpd` (nebo `/var/log/apache2`)
- na unixu je zvolen modul `prefork` (na Windows samozřejmě `mpm_winnt`)
- server je nastaven na střední výkon

Ukázka: první spuštění

- ukázka: spuštění, přístup na stránky
- ukázka: základní konfigurace, nejdůležitější konfigurační hodnoty
- ukázka: logy, rotování logů
- k analýze logů dobře poslouží například výborný webalizer - statistiky přístupů

Apache - moduly

- webový server od ASF je modulární
- je možné ho výborně škálovat, rozšiřovat
- moduly dělíme do dvou hlavních skupin:
 - filters
 - handlers
- filtry nějakým způsobem modifikují vstup nebo výstup serveru
- handlers zpracovávají požadavky

Apache - filtry

- nejvíce se používají výstupní filtry (např. pro kompresi nebo šifrování výsledků)
- `mod_include`: vkládá do HTML stránek uživatelem definované elementy, jiné soubory, výstupy...
- `mod_deflate`: nabízí možnost komprese výsledků (u nevytížených serverů zlepšíme odezvu)
- `mod_ssl`: šifrování komunikace (HTTPS)

Ukázka: filtry

- mod_deflate
- snadný na konfiguraci, výsledek se dostaví ihned

```
<Directory "/var/www/htdocs/komprimovane">  
  AddOutputFilterByType DEFLATE text/html  
</Directory>
```

Ukázka: modul SSL

- SSL (Secure Sockets Layer) poskytuje možnost šifrování a autentizace klienta i serveru
- instalace spočívá v několika krocích
- aktivovat SSL modul, vygenerovat certifikát, restartovat Apache
- pro Debian existuje šikovná utilitka `apache2-ssl-certificate`

Ukázka: modul PHP

- PHP je jednoduchý skriptovací jazyk
- velmi rozšířený, těší se velké oblibě
- instalace modulu je triviální: aktivovat, restartovat
- obliba u začátečníků-programátorů, jako administrátor se musí velmi dbát na bezpečnost

Apache - virtuální hosté

- rozlišujeme dva typy virtuálních hostů:
 - IP based
 - vhost based
- první typ se váže na IP adresu (jedno síťové rozhraní může mít více IP adres - tzv. virtuálních)
- druhý typ je možný díky podpoře hlavičky Host v HTTP dotazu

Apache - virtuální hosté

```
<VirtualHost 172.20.30.40>  
DocumentRoot /www/example1  
ServerName www.example.com  
</VirtualHost>
```

```
<VirtualHost 172.20.30.50>  
DocumentRoot /www/example2  
ServerName www.example.org  
</VirtualHost>
```

Apache - virtuální hosté

NameVirtualHost *:80

<VirtualHost *:80>

DocumentRoot /www/example1

ServerName www.example1.com

....

</VirtualHost>

<VirtualHost *:80>

DocumentRoot /www/example2

ServerName www.example2.com

....

</VirtualHost>

Apache - mod_vhost_alias

get the server name from the Host: header

UseCanonicalName Off

LogFormat "%V %h %l %u %t \"%r\" %s %b" vcommon

CustomLog logs/access_log vcommon

VirtualDocumentRoot /www/hosts/%0/docs

VirtualScriptAlias /www/hosts/%0/cgi-bin