Webový server Apache 2

Seminární práce 4IZ110 (cvičení 007, PhDr. Otakar Pinkas)

Karel Kohout
karel@kohout.se
FIS VŠE, 1. ročník

7.4.2008

Karel Kohout
karel.kohout@centrum.cz
FIS VŠE
7.4.2008

Obsah

Karel Kohout
karel.kohout@centrum.cz
FIS VŠE

1 Úvod

Cílem této seminární práce je popsat základní instalaci, konfiguraci a údržbu Apache[1], provozovaného v produkčním prostředí na serveru s operačním systémem GNU/Linux, distribucí Debian[2]. Zabývám se též některými vybranými důležitými moduly a konfigurací „virtuálních serverů“ (virtual host) a stručného popisu zabezpečení (či absence zabezpečení kvůli výkonu). Uvítám jakékoliv připomínky či doporučení ohledně další konfigurace. Pokud v práci používám Apache, mám na mysli Apache verze 2. Cesty a názvy souborů uvádím tak, jak jsou v Debianu a v ostatních distribucích se mohou mírně lišit. Grafickým rozhraním se nezabývám.

2 Instalace

2.1 Linux

Instalace Apache v Linuxu probíhá nejčastěji z balíčků dané distribuce, pro Debian pomocí programů *apt* či *Aptitude*.

```
root@server:/#apt-get install apache2
```

Pro doinstalování dalších modulů (k nalezení například přes *apt-cache search libapache2-mod*)

```
root@server:/#apt-get install libapache2-mod-nazev_modulu
```

Upgrade Apache probíhá stejně jako zbytek systému přes správce balíčků, tedy v prostředí *Aptitude* nebo pomocí

```
root@server:/#apt-get update
root@server:/#apt-get upgrade
```

Alternativou k instalaci z balíčků je instalace ze zdrojových kódů, získaných buď přes *apt-get source packagename* nebo z webu http://httpd.apache.org/. V drtivé většině případů nemá ruční kompilování žádný efekt na výkon a představuje bezpečnostní riziko, protože každý update znamená nové sestavení, což není za provozu (a při lenosti správce) nejjednodušší – vyplatí se pouze u jednoúčelových serverů nebo velkých serverových farem.

V jiných distribucích založených na Debianu (Ubuntu a deriváty) je možné používat stejné nástroje, u distribucí založených na Red Hat (Fedora, CentOS) existuje *yum*, v Gentoo je *emerge* (z výše uvedených důvodů není Gentoo na běžný server vhodné).

2.2 Windows

Pro operační systém Windows je Apache distribuován jako „instalační .msi“, ve variantě s a bez knihoven OpenSSL. Server neběží standardně po instalaci, předpokládám, že autoři se snaží uživatele donutit alespoň otevřít konfigurační soubor a proto je nutné nastavit port, na kterém má server naslouchat. Konfigurace je obdobná jako pro Linux, nebudu se jí ale zabývat, nemám zkušenosti s provozem serveru pod Windows.

Operační systém od Microsoftu přináší několik problémů – kromě nejnovější verze ho není možné provozovat bez grafického rozhraní, což znamená méně systémových prostředků pro Apache. Údržba je rovněž ztížena, protože server není aktualizován v rámci operačního systému a vše se musí provádět ručně (platí i pro další komponenty MySQL, PHP z klasiky „WAMP“).

[1] Server version: Apache/2.2.8 (Debian), Server built: Jan 17 2008 21:31:10
[2] Linux 2.6.16.28xen #4 SMP Tue Jan 23 15:19:04 UTC 2007 x86_64 GNU/Linux

3 Spuštění

Server je možné ovládat přes

```
/usr/sbin/apache2ctl  start|stop|restart|graceful|graceful-stop| \
configtest|status|fullstatus
```

Druhou možností (kterou nedoporučuji, často se server chybně ukončí) je využít skriptů v init.d:

```
/etc/init.d/apache2 force-reload|restart|start-htcacheclean| \
stop-htcacheclean|reload|start|stop
```

4 Konfigurace

Všechny konfigurační soubory se nachází v */etc/apache2*, v mém případě adresář vypadá takto:

```
./conf.d/
./mods-available/
./mods-enabled/
./sites-available/
./sites-enabled/
apache2.conf
envvars
httpd.conf
ports.conf
```

Adresář či soubor	Funkce
./conf.d/	Ostatní nastavení serveru, které není v *apache2.conf*
./mods-available/	Nastavení dostupných modulů, instalovaných přes správce balíčků
./mods-enabled/	Symlinky na jednotlivé soubory *mods-available*
./sites-available/	Dostupné „virtuální servery“ (vhosts)
./sites-enabled/	Povolené „virtuální servery“ (vhosts)
apache2.conf	Hlavní konfigurační soubor Apache
envvars	Lokální proměnné
httpd.conf	Další konfigurace, v Debianu prázdný soubor (v některých jiných distribucích a ve standardní instalaci Apache je zde hlavní nastavení místo v *apache2.conf*
ports.conf	Porty, na kterých Apache očekává požadavky

V dalším textu se pokusím systematicky probrat jednotlivá nastavení v jednotlivých souborech. Pro načtení nové konfigurace je třeba pustit *apache2ctl*:

```
root@server:/# apache2ctl reload
```

4.1 ports.conf

Obsahuje nastavení jednotlivých portů, na kterých Apache očekává příchozí spojení. Syntaxe je *Listen cislo_portu*, obvykle obsahuje jen 80 (http) a 443 (https), ale je možné zadat libovolný jiný volný port (zde například 444 pro Subversion).

```
Listen 80
Listen 443
Listen 444
```

4.2 envvars

V nových verzích Apache je možné v hlavním souboru *apache2.conf* používat proměnné prostředí místo přímého psaní hodnot. Syntaxe je *export PROMENNA=hodnota*, odpovídá příkazům shellu. Proměnné je možné používat v *apache2ctl*, */etc/init.d/apache2*, */etc/logrotate.d/apache2* a podobně.

```
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
export APACHE_PID_FILE=/var/run/apache2.pid
```

4.3 apache2.conf

Hlavní konfigurační soubor. Velká část nastavení je specifická pro jednotlivé virtuální servery a proto jejich popis vynechám a budu se zabývat jen rozebíráním položek důležitých pro celý server. Syntaxe je *Promenna Hodnota*, umožňuje i nastavení podmíněné zavedením modulu v blocích <*IfModule jmeno_modulu*> , uzavírá se <*/IfModule*>. U některých proměnných se mi nepovedlo vypátrat jejich význam.

4.3.1 Interní nastavení

```
ServerRoot "/etc/apache2"
LockFile /var/lock/apache2/accept.lock
PidFile ${APACHE_PID_FILE}
```

ServerRoot udává hlavní adresář Apache, *LockFile* je relevantní pouze v případě síťových disků (musí být na lokálním úložišti), do *PidFile* server ukládá číslo svého procesu.

```
Timeout 150
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
```

Timeout je hodnota, po jejímž uplynutí server (překvapivě) pošle timeout, zajímavá položka *KeepAlive* povoluje posílání více dotazů v rámci jednoho spojení (výhodné, pokud jich klient posílá najednou několik za sebou v krátkém časovém úseku), *MaxKeepAliveRequests* udává maximální počet takových spojení a *KeepAliveTimeout* po jaké době budou ukončeny, pokud není odezva klienta (všechny údaje v sekundách).

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

Standardní balíček Apache v Debianu je MPMprefork, tedy verze, která nepoužívá vlákna, ale spouští samostatné procesy pro každé příchozí spojení. Teoreticky je mírně pomalejší než MPM-worker, zato nemá problémy s moduly a aplikacemi, které nejsou „thread-safe“.

StartServers je počet procesů, které mají být spuštěny na začátku, *MinSpareServers* je minimální počet volných procesů, které mají čekat „v záloze“ na příchozí spojení, *MaxSpareServers*

jejich maximální počet. *MaxClients* udává maximální počet spojení na celý server a *MaxRequestsPerChild* je maximální počet požadavků, který jednotlivý proces vyřídí (0 znamená neomezeno).

```
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
```

Uživatel (*User*) a skupina (*Group*), pod kterými Apache běží, nastaveno v *envvars*, v Debianu standardně *www-data* a nikdy *root*.

4.3.2 Kontrolní soubory

```
AccessFileName .htaccess

<Files ~ "^\.ht">
Order allow,deny
Deny from all
</Files>
```

Apache při vyřizování požadavku prohledává každý adresář na vyšší úrovni až k požadovanému obsahu, zda v něm není „speciální“ soubor .htacces. Pokud ho nalezne, provede v něm obsažené příkazy. Jméno souboru je možné nastavit v *AccessFileName*, sekvence *<Files...* brání jeho zobrazení přes web.

4.3.3 Zaznamenávání chyb

```
HostnameLookups Off
```

Kontroluje dohledávání doménových jmen k IP adresám, s ohledem na rychlost obvykle vypnuté.

```
ErrorLog /var/log/apache2/error.log
LogLevel warn
```

ErrorLog udává umístění hlavního souboru s chybami (do něj se zapisují problémy s moduly a podobně), *LogLevel* úroveň chyb, jaké se mají zaznamenávat (hodnoty: debug >> info > notice >> warn > error > crit > alert >> emerg), v produkčním prostředí se nedoporučuje hodnota vyšší než warn (případně notice), protože info a zvláště debug znamenají enormní množství údajů na spojení (řádově kilobajty až desítky kilobajtů pro https).

Nastavení logů pro neexistující soubory a přístupy popisuji u jednotlivých „virtuálních serverů“ (vhost), úrovně chyb ale zůstávají stejné.

4.3.4 Include xyz

```
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf
Include /etc/apache2/httpd.conf
Include /etc/apache2/ports.conf
```

V principu je možné veškerý obsah souborů v */etc/apache2* nacpat do */etc/apache2/apache2.conf*, což je ovšem nepraktické, proto se vkládají pomocí direktivy *Include /uplna/cesta/k/souboru*.

4.3.5 Ukrytí serveru

```
ServerTokens Prod
```

Ovlivňuje identifikaci serveru, která se odesílá v hlavičkách (hodnoty Full > OS > Minor > Minimal > Major > Prod). Z bezpečnostních důvodů je vhodné nastavit *Prod*, kdy se server „představuje“ jen jako „Apache“, oproti ostatním, kdy odchází i verze. Zabezpečení není efektivní, protože verzi serveru je možné najít například v php pomocí *phpinfo ()*, což ovšem předpokládá přístup k serveru. Eliminuje útoky script-kiddies či masové vyhledávání Apache v konkrétní verzi s konkrétní chybou.

```
ServerSignature Off
```

Kontroluje, jaké údaje se zobrazí na chybových stránkách - s *On* opět relativně nebezpečné, pokud na serveru běží WebDAV, mod_svn, mod_python a podobně – o těchto službách na nestandardních portech (s využitím PostSentry) nemusí útočník vědět a je šance, že se během hledání prozradí.

Příklad:

```
Apache/2.0.54 (Unix) mod_ssl/2.0.54 OpenSSL/0.9.7b PHP/5.1.0RC1 \
Server at x Port 80
```

4.3.6 Ostatní

```
NameVirtualHost *:80
NameVirtualHost *:443
NameVirtualHost *:444
```

Nastavení virtuálních serverů – na kterých portech a IP adresách má Apache očekávat spojení. Syntaxe: *NameVirtualHost ip_adresa:port*. Varianta *NameVirtualHost 80* způsobuje chyby v případě více portů nebo nestandardní konfigurace. Lze kombinovat IP adresy (místo hvězdičky).

```
Auth_MySQL_Info localhost uzivatelske_jmeno heslo
```

Apache v rámci jednoho z modulů umožňuje získávat uživatelská jména nejen z .htaccess (.htpasswd) souborů, ale i z databáze (MySQL, PgSQL,...) Podporuje i přístup dle skupin – výhodou je, že takovouto databázi je možné využít i pro přístup přes ftp (umí například PureFTPD) nebo další služby, ať už v rámci samostatného přihlašování nebo přes Apache a „Basic authentication“ (nejlépe přes SSL zabezpečené připojení).

```
AddType application/x-x509-ca-cert .crt
```

Tato direktiva nastavuje, jak bude Apache v odesílané hlavičce daný soubor identifikovat. Zde donutí prohlížeč nabídnout instalaci kořenového certifikátu.

```
DefaultType text/plain
```

Standardní chování pro soubory, u kterých Apache nerozpozná MIME typ (a tedy je odešle v http(s) hlavičce jako prostý text).

5 Moduly

Moduly pro Apache lze rozdělit do dvou skupin – na moduly přímo v jádře, „zabudované“ při kompilaci (core, prefork,...) a na dynamicky zaváděné moduly, jejichž výběr lze provést při každém spuštění serveru. (drtivá většina ostatních). Dostupné moduly (nainstalované přes správce balíčků) mají zaváděcí soubory v */etc/apache2/mods-available*, a spouštěny jsou všechny v */etc/apache2/mods-enabled* (obvykle symlink do *mods-available*. Většina modulů má naprosto minimální standardní konfiguraci a vše se nastavuje pro jednotlivé vhosts.

5.1 Přidání nebo odebrání modulů

Nejrpve je nutné moduly nainstalovat přes *apt* / *Aptitude*

```
root@server:/# apt-get install libapache2-mod-php5
```

Aby Apache začal modul používat, je třeba modul zavést a poté Apache restartovat.

```
root@server:/# a2enmod  php5
root@server:/# apache2ctl reload
```

Odebírání modulů probíhá přes *a2dismod*. Pochopitelně je možné i ručně vytvořit symlink z *mods-available* do *mods-enabled*.

5.2 Zajímavé moduly

Modul	Využití
dav, dav_fs	Umožňuje použít WebDAV, a spolu s *dav_svn* provozovat na serveru Subversion přístupné přes Apache.
mime a negotiation	Identifikuje typ souboru (text/plain, image/gif), pomocí určitých direktiv podporuje i ruční nastavení (například jazyku, kódování, atd.)
fcgid	Nahrazuje moduly cgi, fastcgi (teoreticky je rychlejší), Apache díky němu spouští cgi skripty, ve spolupráci s dalšími moduly i skripty v jiných jazycích.
php5	Jeden z nejčastěji provozovaných modulů, propojuje Apache a skriptovací jazyk PHP.
mod_python	Podpora pro skripty v Pythonu, respektive jejich spouštění přes Apache a ne přes zabudovaný server – lze tak provozovat například Trac.
ssl	Zabezpečení SSL – zdroj častých problémů. Při použití SSL certifikátů u vhosts (tedy více domén druhého řádu na jedné IP adrese) bude server hlásit chybu (viz **??**) a v certifikátech je nutné uvést * místo IP adresy/domény (a pochopitelně pak nejde použít kvalifikovaný certifikát). Situace se zlepšuje postupným zaváděním TLS, které ale nezvládá Internet Explorer.
vhost_alias	Podpora pro virtual hosts.
auth(z)_x	Moduly starající se o přihlašování skrze Apache, lze omezit přístup k adresářům, případně navázat další programy (svn) nebo skripty (php, python – Trac). Méně známý je *auth_mysql*, získávající údaje z databáze.
rewrite	Po PHP další klasika, na základě zadaného řetězce zvládá přepisovat za běhu adresy v požadavcích přicházejících na server, například z *xyz.cz/objednat/knihy/apache_bible* vytvoří *xyz.cz/index.php?action=order&cat=knihy&prod=apache_bible* – dnes základ pro „pěkná URL“. Konkrétní nastavení se provádí buď v souborech .htaccess, nebo u jednotlivých vhosts.

5.3 Méně zajímavé moduly

Moduly, které se tváří zajímavě, ale jejichž provoz je v praxi problematický nebo se mi neosvědčil.

5.3.1 mod_security / mod_security2

Na první pohled úžasná věc – filtr, kterým musí projít jakýkoliv požadavek od klienta než se vůbec k Apache dostane. Teoreticky tak lze zamezit útokům typu SQL injection, cross-site scripting a podobně. Bohužel, mod_security se standardními pravidly odchytává zcela běžné (byť nebezpečné) aplikace typu Wordpress, phpBB a tudíž způsobuje víc škody než užitku. Mod_security2 má navíc velmi komplikovaná pravidla (výrazy na několik řádků).

5.3.2 suphp

Suphp je modul, který php skripty spouští vždy pod vlastníkem souboru a nikoliv jako www-data. Paradoxně tak ohrožuje jednotlivé stránky, protože většině útoků obvykle zabrání odlišný vlastník souboru (například ftp-user), tudíž je www-data nemůže přepsat – avšak s suphp by klienti museli nastavit soubory jen pro čtení (444), což je mimo realitu. Navíc suphp je mimořádně náročný na výkon serveru (spouští se pod daným uživatelem až s příchozím požadavkem).

Do stejné kategorie spadá (byť se nejedná o modul) i provozování Apache „v chroot“, které sice chrání server, ale ne to nejpodstatnější – data, ke kterým má Apache přístup.

6 „Virtuální servery“ - virtual hosts

V této části se pokusím rozebrat některé zapeklitosti správné konfigurace virtuálních serverů (přesněji „name-based virtual hosts“). Konfigurace, kterou popisuji, je podřízena některým specifickým nastavením serveru a nemusí být nezbytně jediná správná.

Virtuální servery je možné nastavit buď přímo v apache2.conf (httpd.conf) nebo v adresářích, z nichž budou soubory načteny pomocí příkazu *Include*. V Debianu sídlí v *sitesavailable*, povolují se buď přidáním symlinku do *sitesenabled* nebo spuštěním

```
root@server:/#a2ensite jmeno_souboru_s_vhost
```

Analogicky funguje *a2dissite*.

6.1 Ukázková konfigurace

Uvádím zde jeden rozsáhlejší funkční soubor pro doménu s návštěvností kolem 1500 UIP denně.

6.1.1 Hlavička

```
<VirtualHost *:80>
ServerName        domena.tld
ServerAdmin       nobody@domena.tld
ServerAlias       *.domena.tld
DocumentRoot       /www/domena.tld/web/
```

<**VirtualHost *:80**> Udává, na jakém portu Apace naslouchá. Hvězdičku je možné nahradit IP adresou, port musí odpovídat hlavnímu nastavení NameVirtualHost *:80. Chybné údaje způsobují velké problémy (web dostupný pod jinou doménou). Zároveň slouží jako odlišení začátku souboru.

ServerName domena.tld Doménové jméno, pod kterým Apache bude nabízet obsah specifikovaný dále. Technicky vzato, v každém http (HTTP 1.1) požadavku (v novějších prohlížečích) je specifikována doména, kterou následně Apache hledá mezi virtuálními servery. Pokud nenajde odpovídající ani u jednoho v ServerName (například přístup přes IP adresu), zobrazí obsah vhost označeného jako default. Je možné specifikovat několik domén: *ServerName domena.com ServerName domena.net*

ServerAlias *.domena.tld Pod jakými jinými doménami server naslouchá – obvykle subdomény.

DocumentRoot /www/domena.tld/web/ Udává, který adresář bude brán jako kořenový – bez dalšího nastavení požadavek na http://domena.tld/index.html směřuje na /www/domena.tld/web/index.htm

```
    php_admin_value open_basedir /www/domena.tld/web/
    php_admin_value upload_tmp_dir /www/domena.tld/php_upload/
    php_admin_value session.save_path /www/domena.tld/tmp/
```

php_admin_value Mění hodnoty nastavené pro PHP, které nelze následně přepsat v jednotlivých .htaccess souborech (Apache při pokusu poněkud neprakticky hlásí chybu 500 – Internal server error, díky čemuž může neohlášená změna nastavení způsobit nedostupnost některých stránek).

```
<Directory />
        Options             FollowSymLinks -Indexes
        AllowOverride       AuthConfig FileInfo Indexes Limit
</Directory>
```

<**Directory** /> Nastavení platící jen pro daný adresář (absolutní cesta, oproti Location s relativní z URL).

Options *FollowSymLinks* – Apache bude sledovat symlinky v adresářích, *-Indexes* – nebude zobrazovat všechny soubory v adresáři, pokud nenajde (obvykle, viz *DirectoryIndex*) index.htm(l)/php.

AllowOverride Udává, která nastavení je možné změnit v souborech .htaccess v adresářích jednotlivých stránek.

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^www.domena.tld [OR]
RewriteCond %{HTTP_HOST} domena.tld [OR]
RewriteRule ^(.*) /www/domena.tld/web/www/$1 [L]
```

Zapíná *mod_rewrite*, neboli úpravy adresy požadavku před pracováním serverem. Lze využít ke vzbuzení „dojmu“ automaticky vytvářených subdomén (přináší určité drobné problémy s chybovými stránkami – server je nemusí najít). Uváděný příklad není kompletní, ale bohužel rozsah práce mi neumožňuje vše rozebrat dopodrobna.

```
    DirectoryIndex index.html index.htm index.php index.php5 index.php4 index.ph
    AccessFileName     .htaccess
```

DirectoryIndex Nastavuje, které soubory má Apache nabízet, pokud v požadavku není žádný konkrétní.

```
    HostnameLookups    Off
    ErrorLog           /www/domena.tld/logs/error_log
    LogLevel           warn

#    LogFormat         "%t %h %{User-Agent}i" muj
#    LogFormat         "%h %l %u %t \"%r\" %%>$s %b \"%{Referer}i\" \"%{User-Agent
#    LogFormat         "%h %l %u %t \"%r\" %%>$s %b" common
#    LogFormat         "%{Referer}i -$>$ %U" referer
#    LogFormat         "%{User-agent}i" agent

#    CustomLog          /www/domena.tld/logs/access_log muj
```

HostnameLookups Umožňuje vypnout dohledávání DNS záznamů k IP adresám – u zatížených stránek vhodné vypnout.

ErrorLog Umístění souboru s chybami.

LogLevel Úroveň chyb, které má server zaznamenávat.

LogFormat Definuje vlastní formát záznamů, obvykle pro sledování návštěvnosti. Na zatíženém serveru může soubor růst o desítky MB denně (proto zakomentováno – vhodné spíše na testování například výrazů z mod_rewrite).

```
    ServerSignature    Off
```

ServerSignature Jaké údaje se zobrazí na chybové stránce – nejlépe žádné (standardně email administrátora, podrobnosti o serveru).

```
        Alias /error/   "/www/domena.tld/error/"
        ErrorDocument  400  /error/400.htm"
#    "401 Authorization Required",
        ErrorDocument  401  /error/401.html
#    "403 Forbidden",
        ErrorDocument  403  /error/403.html
#    "404 Not Found",
        ErrorDocument  404  /error/404.html
#    "500 Internal Server Error",
        ErrorDocument  500  /error/500.html
```

Alias Vytváří „symlink“ ve stylu Apache.

ErrorDocument Umožňuje definovat vlastní chybové stránky pro jednotlivé domény druhého řádu (vhosts) a subdomény. Při současném nastavení musím do každé subdomény umístit symlink na adresář error (způsobeno nejspíš řešením subdomén přes mod_rewrite).

6.2 Další konfigurace

Mimo výše uvedené je možné využít některé další zajímavé příkazy.

```
        SSLEngine on
        SSLCipherSuite HIGH:MEDIUM:!aNULL:+SHA1:+MD5:+HIGH:+MEDIUM
        SSLCertificateFile /www/domena.tld/ssl/domena.tld.cert.pem
        SSLCertificateKeyFile /www/domena.tld/ssl/domena.tld.key.pem
```

Zapíná SSL šifrování. Doporučuji nastavit použité sady šifer na co nejúčinnější (ač se jedná o určitou zátěž navíc pro server). V certifikátech pro Apache není dobré vyplňovat heslo, protože potom při automatickém restaru zůstane proces viset a čeká na vstup od správce, aby mohl certifikát dešifrovat.

```
<Directory /www/domena.tld/web/trac/trac/>
        AuthMYSQL on
        AuthMySQL_Authoritative on
        AuthMySQL_DB databaze
        AuthMySQL_Password_Table users
        Auth_MySQL_Username_Field username
```

```
        Auth_MySQL_Password_Field password
        AuthMySQL_Group_Table users
        Auth_MySQL_Group_Field gname
        AuthMySQL_Empty_Passwords off

        AuthName "Trac"
        AuthType Basic
        require group trac admin

        SetEnv TRAC_ENV "/www/domena.tld/web/trac/trac/"
        SetEnv PYTHON_EGG_CACHE "/www/domena.tld/web/trac/trac/tmp_python/"

        SetHandler mod_python
                #PythonDebug on
        PythonInterpreter main_interpreter
        PythonHandler trac.web.modpython_frontend
        PythonOption TracEnv /www/domena.tld/web/trac/trac/
        PythonOption TracUriRoot /trac/
        PythonPath "sys.path + ['/www/domena.tld/web/trac/trac/']"
        SetEnv HTTPS 1

</Directory>
```

Jednoduché nastavení pro Trac, oblíbený vývojářský software (integruje wiki, přehledy z svn, seznam úkolů,...) K běhu je možné použít buď (f)cgi nebo mod_python (zde Python, který je výrazně rychlejší). Uživatelské účty přebírá Trac z MySQL databáze skrze Apache – proto je nutné vynutit https[3], aby nebylo možné číst hesla po cestě v „plain-textu“.

```
<Location /xyz>
        DAV svn
        SVNPath /var/subversion/domena.tld/

        AuthMYSQL on
...

        AuthName "Svn"
        AuthType Basic
        require group svn admin
</Location>
```

Základní konfigurace svn, opět s přihlašováním přes Apache a WebDAV (svn tak neběží jako samostatný proces a do repository se dostane například výše uvedený Trac). Je vhodné pro dané umístění vypnout jakékoliv mody, které chrání Apache přes DDoS útoky.

[3]Například tím, že určité speciální aplikace jsou přístupné přes virtuální server, který naslouchá jen na portu 443 a vyžaduje šifrované připojení, zatímco běžné stránky jdou přes jiný virtuální server na portu 80.

7 Poznámky, závěr

7.1 Výkon

Nikde jsem nenašel spolehlivou statistiku počtu dotazů, které Apache zvládne obsloužit – pouze poznámku o schopnosti Pentia II 450Mhz zahltit statickými stránkami 40 Mbps linku. Z vlastní zkušenosti mohu tvrdit, že server je nad očekávání výkonný a s výjimkou skriptů v Pythonu se *load* pohybuje mezi 0 a 0.1 (ani PHP s použitím vhodné cache není schopné server s 500Mhz CPU zahltit[4]).

7.2 Stabilita

Apache 2 je mimořádně stabilní a v zásadě bezproblémový (v kombinaci s jednoduchým skriptem typu *monit*, kontrolujícím vytížení a počet proceslů, není třeba jakákoliv obsluha). Kolaps kvůli přetížení / chybnému postavení hvězd lze velice jednoduše předvídat dle času od posledního, prakticky se jedná zhruba o jeden problém vyžadující lidský zásah za čtvrt roku (obvykle extrémní vytížení serveru, kdy se bez restartu Apache sám není schopný ukončit).

7.3 Rizika

Nevhodný update gcc a vlastní moduly mohou způsobit častý SegFault, především ve chvíli, kdy je Apache zkompilovaný pod jednou verzí a modul pod druhou. Podobně i cgi skripty mohou způsobit pád serveru nebo jeho extrémní vytížení (je tedy vhodnější používat například *mod_python* všude, kde to jde).

7.4 Srovnání s konkurencí

Z jiných webových serverů jsem krátce otestovat lighthttpd, který nabízí mírné zvýšení výkonu na úkor konfigurovatelnosti. Na specializovaných serverech najde využití, ale v dnešní době je úzké hrdlo spíš CPU (kompilace PHP) a RAM (MySQL), samotný webový server zabírá minimum systémových prostředků.

IIS jsem neměl to potěšení (?) zkoušet v reálném zatížení (samotné spuštění Windows by zabralo většinu serveru). Při zběžném nainstalování podle mého názoru obsahuje všechny výhody i nevýhody „klasických“ programů pro Windows – „přístupnost pro širokou veřejnost“, absenci textové konfigurace a v podstatě nutnost používat GUI (celkem obtížné například při připojení přes GPRS).

V praxi používám ještě webový server napsaný v Pythonu pro zobrazení chybové hlášky při údržbě (nabíhá velice rychle a není výkonově náročný).

7.5 Jazykové varianty

Pokoušel jsem se nastavit lokalizované chybové stránky (404 atd.), ale bohužel Apache ne vždy zcela srozumitelně popisuje problém, proto mi nikdy nefungovaly. Různé jazykové mutace normálních stránek jsem též zkoušel, ale nepovažuji je za ideální řešení s ohledem na uživatele, který nemá jednoduchou možnost přepnout jazyk a z hlediska tvůrce webových stránek je obtížné detekci napojit na redakční systém.

[4] Překvapivě největší „žrouti“ jsou jakékoliv emaily (Courier/Postfix), i při minimu účtů je množství spamu enormní a i bez spamového filtru server „ve špičce“ položí, což se ale netýká Apache.

7.6 Místní přístup k manuálům, uživatelské adresáře

Přístup k manuálovým stránkám jsem nezkoumal, předpokládám, že se dá řešit symlinkem do potřebného adresáře. Uživatelské adresáře ve smyslu prostoru pro systémové uživatele, který je přístupný z webu, jsem taktéž nezkoumal – na serveru z důvodu bezpečnosti nemá nikdo z běžných klientů účet (FTP server přiděluje UID > 2000, nelze se tedy na údaje z nešifrovaného FTP přihlásit, protože účty neexistují).

7.7 Indexy adresářů

Standardně je server nastavený, aby zobrazil chybovou hlášku pokud v daném adresáři při pokusu o přístup není index.html/php – jedná se o zcela logické opatření. Zapnout indexy lze přes *Options +Indexes* pro jednotlivé virtuální servery, nejlépe jen v přesném místě pomocí *<Directory>* nebo *<Location>*. Lze čarovat se způsobem zobrazení, nejsem si ale jistý, kolik lidí to ocení.

7.8 Testy

Neprováděl jsem při psaní práce žádné speciální testy, správnost konfigurace ale myslím potvrzuje relativní nevytíženost serveru a uptime v řádech měsíců. Pro ilustraci uvádím objem přenesených dat (z části se na datech podílí i FTP–upload, který činí do 1GB měsíčně, stahování přes FTP se podílí opět maximálně 1 GB).

```
# vnstat -m

 eth0  /  monthly

   month         rx      |      tx      |   total
------------------------+--------------+---------------------------------------
... (data za posledních 6 měsíců) ...
  Nov '07       8.11 GB  |    87.08 GB  |    95.19 GB   %::::::::::
  Dec '07       7.74 GB  |   125.31 GB  |   133.05 GB   %::::::::::::::::
  Jan '08       7.47 GB  |   123.23 GB  |   130.70 GB   %::::::::::::::::
  Feb '08       6.96 GB  |    115.34 GB  |   122.30 GB   %:::::::::::::::
  Mar '08       8.42 GB  |   144.68 GB  |   153.10 GB   %::::::::::::::::::::
  Apr '08       1.76 GB  |    18.45 GB  |    20.21 GB   ::
------------------------+--------------+---------------------------------------
 estimated      6.97 GB  |    72.97 GB  |    79.94 GB
```

Bohužel nemám statistiky zatížení CPU, paměti atd. (munin-node), tvorba efektně vypadajících grafů se projevila jako zbytečná zátěž (a speciální modul, mod_watch, který umožňoval generovat statistiky dle jednotlivých viturálních serverů, jsem musel ručně kompilovat, což je vždy zdlouhavé).

A Použité zdroje

http://httpd.apache.org/docs/2.2/
Manuálové stránky stránky Debianu http://www.howtoforge.org
Linux – Kapesní průvodce administrátora, Martin Kysela, Grada Publishing, a.s., Praha, 2004
http://www.debianadmin.com/install-and-configure-apache2-with-php5-and-ssl-support-in-debian-etch.html
http://www.debianhelp.co.uk/selfcert.htm

B Výpisy z logů

B.1 /var/log/apache2/error_log

```
[Sun Apr 06 06:26:15 2008] [notice] mod_python: Creating 8 \
session mutexes based on 150 max processes and 0 max threads.
[Sun Apr 06 06:26:15 2008] [notice] mod_python: using mutex_directory /tmp
[Sun Apr 06 06:26:16 2008] [warn] Init: SSL server IP/port conflict: domena1.tld:443
(/etc/apache2/sites-enabled/domena1.tld.https:1) vs. domena2.tld:443 \
(/etc/apache2/sites-enabled/domena2.tld:1)
...
[Sun Apr 06 06:26:16 2008] [warn] Init: \
You should not use name-based virtual hosts in conjunction with SSL!!
[Sun Apr 06 06:26:16 2008] [notice] Apache/2.2.8 \
(Debian) DAV/2 SVN/1.4.6 mod_python/3.3.1 Python/2.4.5
mod_ssl/2.2.8 OpenSSL/0.9.8g configured -- resuming normal operations
```