TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: X2612 - Elektrotechnika a informatika
Studijní obor: 39067001 - Mechatronika

Přístupové statistiky www stránky

Access statistics of www page

Diplomová práce

Autor: Lukáš Bartůněk
Vedoucí práce: Mgr. Jiří Vraný

V Liberci dne 16. 5. 2007

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

Datum:

Podpis:

Poděkování

Rád bych poděkoval vedoucímu diplomové práce Mgr. Jiřímu Vranému za spolupráci a připomínky při realizaci diplomové práce. Dále firmě VIA.CZ s.r.o., která mi poskytla logovací soubory serveru Apache a webový prostor pro umístění aplikace při její realizaci a zkoušení. A především rodině za podporu při studiu.

Anotace

Cílem diplomové práce je seznámit čtenáře s problematikou záznamu dat o chování uživatelů na webovém serveru Apache a představit způsob i možnosti zpracování logovacích souborů. Další částí je rešerže dostupných řešení pro analýzu logovacích souborů a sledování uživatelů webových serverů.

Zbývající částí je návrh aplikace pro analýzu logovacích souborů serveru Apache. Aplikace disponuje importem stávajícího logovacího souboru do databáze a dále data zpracovává pro statistické údaje (návštevy, odeslaná data, chybová hlášení). Hlavní činností je možnost sledování jednotlivých uživatelů identifikovaných dle IP adres.

Klíčová slova: Apache, statistiky, logování, logy

Abstract

The aim of this Diploma is to make the readers acquainted with the problems of recording the data of the users acting on the web server Apache as well as the ways and possibilities of how to treat the login files. The next part concentrates on a research of the available solutions for the analysis of login files and observing the web users.

The remaining part of this study is an application concept for the analysis of the login files on the server Apache. The application disposes of an import of the actual login file into the database and processes those data for the statistics (visits, sent data, error reports etc.) onward. The main function is the observing of each user, who is beeing identified according to their IP addresses.

Keywords: Apache, statistics, logging, logs

Obsah

Prohlášení ... 3
Poděkování ... 4
Anotace ... 5
Abstract ... 5
Seznam použitých zkratek ... 8
Úvod ... 9
1 Teoretická část ... 10
1.1 Webový server Apache ... 10
1.1.1 Popis ... 10
1.1.2 Logování ... 11
1.2 Statistická data ... 15
1.3 Dostupná řešení ... 18
1.3.1 Online systémy ... 18
1.3.2 Offline systémy ... 23
2 Praktická část ... 27
2.1 Návrh aplikace ... 27
2.2 Funkce aplikace ... 29
2.2.1 Statistiky ... 29
2.2.2 Sledování uživatelů ... 30
2.2.3 Import, nastavení ... 31
2.2.4 Export dat ... 31
2.3 Realizace ... 32
2.3.1 Použitá řešení ... 32
2.3.2 Zabezpečení ... 37
2.3.3 Struktura ... 38
2.3.4 Spouštění ... 38
2.3.5 Import, nastavení ... 40
2.3.6 Statistiky ... 45
2.3.7 Sledování ... 47
2.3.8 Export ... 48
Závěr ... 49
Literatura ... 50
Příloha A – Přehled HTTP kódů ... 51
Příloha B – Náhled aplikace - statistiky ... 52
Příloha C – Náhled aplikace - sledování ... 53
Příloha D – Náhled aplikace – sledování, detail ... 54

Seznam obrázků a tabulek

Obrázek 1: Souborová struktura aplikace.....38
Obrázek 2: Náhled domovské stránky aplikace.....39
Obrázek 3: Import nových dat.....42
Obrázek 4: Náhled formuláře pro nastavení zobrazovacích proměnných.....44
Obrázek 5: Náhled přehledu dostupných tabulek v databázi.....45
Obrázek 6: Náhled a přehled statistik.....46
Tabulka 1: Výpis dostupných argumentů pro konfiguraci logovacího souboru.....12
Tabulka 2: Rozpis údajů Common Log Format.....13
Tabulka 3: Rozpis údajů Combined Log Format.....14
Tabulka 4: Rozvržení databázové tabulky pro logovací data.....34
Tabulka 5: Rozvržení databázové tabulky pro statistická data.....35
Tabulka 6: Rozvržení databázové tabulky pro rozšířená statistická data.....36
Tabulka 7: Rozvržení databázové tabulky pro IP adresy a jejich doménová jména.....36
Tabulka 8: Rozvržení databázové tabulky pro nastavení.....36
Tabulka 9: Příklad záznamu obecných statistických dat.....43
Tabulka 10: Příklad záznamu pro TOP ukazatel (prohlížeče).....44

Seznam použitých zkratek

CGI	Common Gateway Interface, skript, který umožňuje vytváření dynamických stránek
CLF	Common Log File, formátovací standard pro logovací soubory serveru Apache
CSS	Cascading Style Sheets neboli kaskádové styly, soubor metod pro grafický vzhled webových stránek
DNS	Domain Name Server, systém pro převod číselných IP adres na doménová jména a opačně
FTP	File Transfer Protocol, internetový protokol pro přenos souborů mezi počítači bez závislosti na platformě
GPL	General Public License, licence pro svobodný software, lze v něm svobodně provádět úpravy a dále jej rozšiřovat pod GPL
HTML	HyperText Markup Language, značkovací jazyk pro tvorbu internetových stránek
HTTP	Hyper Text Transfer Protocol, internetový protokol určený pro výměnu hypertextových dokumentů
ID	Identification, jedinečné označení objektu
IFABC	Mezinárodní organizace definující standardy v oblasti auditu tisku, internetu
IIS	Internet Information Services, webový server společnosti Microsoft
IP adresa	Jednoznačná identifikace uživatele internetu
MySQL	Databázový server
NAT	Network Address Translation, překlad internetových adres, technologie umožňující skrytí více uživatelů/počítačů za jednu věřejnou IP adresu
Perl	Interpretační programovací jazyk, používaný především u webových serverů, vyniká rychlou prací se soubory
PDF	Portable Document Format, souborový formát pro uchování dokumentů
PHP	Skriptovací jazyk pro tvorbu interaktivních webových aplikací
SEO	Search Engine Optimalization, metodika optimalizace webových stránek
URL	Uniform Resource Locator, textový řetězec přesně specifikující umístění zdrojů dokumentů nebo služeb na internetu
XHTML	eXtensible Hypertext Markup Language, neboli rozšiřitelný hypertextový značkovací jazyk sloužící k tvorbě hypertextových dokumentů
XML	eXtensible Markup Language, rozšiřitelný značkovací jazyk

Úvod

S pokračujícím rozšiřováním informací publikovaných formou webových stránek vzniká čím dál větší potřeba sledovat, shromažďovat a analyzovat získaná data o uživatelích těchto informačních zdrojů. Takto získaná data se využívají pro optimalizaci obsahu, omezení chyb vyskytujících se na webových stránkách nebo jako další data pro zpracování v ekonomické a marketingové sféře.

Cílem diplomové práce je seznámení s problematikou analýzy dat, shrnutí dostupných systémů umožňujících procházet data a vytvoření vlastní aplikace, zaměřené na sledování konkrétních uživatelů.

Teoretická část diplomové práce se zabývá jedním z nejpopulárnějších webových serverů ve světě internetu, umožní tak čtenáři získat přehled o dostupných systémech, jejich kladech i záporech.

Praktická část představuje realizaci webové aplikace, která umožňuje zpětnou analýzu dat a možností prezentovat získané informace v přehledné formě. K realizaci aplikace jsem zvolil programovací jazyky XHTML, PHP s podporu databáze MySQL. Díky silné vývojářské a uživatelské podpoře těchto technologií lze provést všechny potřebné kroky pro vývoj aplikace, aniž by ztratila na flexibilitě.

1 Teoretická část

1.1 Webový server Apache

1.1.1 Popis

Apache je projektem organizace Apache Software Foundation sídlící v USA. Jedná se o jeden z nejpopulárnějších webových serverů současného internetu, dle přehledu společnosti Netcraft[1] zaujímá 56 % podílu trhu. Mezi hlavní faktory tak významného úspěchu patří:

- ➔ open-source[2] licence umožňující nasazení v komerční i nekomerční sféře internetu
- ➔ silná základna vývojářů, pokračující otevřený vývoj
- ➔ kvalita, stabilita a bezpečnost
- ➔ podpora více platforem (Unix, Linux, Windows, sálové počítače atd.)
- ➔ modulární architektura pro přidávání, odebírání a upravování jednotlivých funkcí dle aktuálních potřeb webového serveru

Kvalitu serveru Apache, či produktů z něj odvozených, potvrzují i známé servery např. www.amazon.com, www.yahoo.com, www.mp3.com nebo www.download.com.

Apache je distribuován ve dvou základních verzích, Apache 1.3 a 2.0, druhá zmíněná je prozatím nejlepší, oproti předchůdci umožňuje nastavit chod serveru jako procesově orientovaného nebo čistě vláknového. Použití vláken (součást procesu, zpracovávají se paralelně) přineslo odlehčení zatížení, na úkor tohoto klesla spolehlivost, v případě chyby může dojít k ovlivnění dat či jiných vláken. Apache ve verzi 1.3 i 2.0 disponuje modulární architekturou, prostřednictvím níž lze aktivovat moduly nové, případně deaktivovat staré, a tím rozšířit funkcionalitu. U novější verze došlo k implementaci filtrů, které umí kódovat/modifikovat data získaná například z jiných modulů.

1 Netcraft: Anglická společnost provádějící průzkum a analýzy mnoha aspektů internetu. http://news.netcraft.com

2 Open-Source: Software s otevřeným zdrojovým kódem, do kterého lze nahlédnout, případně provést úpravy. http://cs.wikipedia.org/wiki/Open_source

1.1.2 Logování

Logování je souhrn činností zajišťujících sběr dat o určitém procesu (požadavky, úspěšnost zpracování, chybové stavy).

Webový server Apache umožňuje ukládání neboli logování dat o stavu a chování systému pomocí svých modulů. V našem případě se jedná o doplňkový modul **mod_log_config**, z něhož lze získat mnoho informací o aktivitě jednotlivých uživatelů zasílajících požadavky na webový server.

Modul se dělí na 3 základní části:

- *TransferLog* vytvářející soubor se záznamy
- *LogFormat* definice formátu záznamu
- *CustomLog* definice formátu záznamu, umístění souboru

Po aktivaci modulu dochází k zaznamenávání všech požadavků, které jsou serverem zpracovávány. Složení záznamů je plně konfigurovatelné pomocí formátovacího řetězce. Vytváření vlastních pravidel pro záznam událostí je doporučováno pouze pro vlastní analýzu dat, takto získaná data již nelze zpracovávat v dostupných softwarových řešení určených pro analýzu logovacích souborů.

Mezi nejpoužívanější standardizované formáty patří *Common Log Format* a *Combined Log Format*. Výhodou standardizace je jednoznačnost získaných dat, které se dají například analyzovat pomocí externích nástrojů. Argumenty používané v *LogFormat* jsou popsány v Tabulce 1 a představují skupinu informací, které lze o každém návštevníkovi získat. Parametry „%x“ jsou při zpracování nahrazovány jednotlivými hodnotami.

Tabulka 1: Výpis dostupných argumentů pro konfiguraci logovacího souboru

Parametr	***Popis***
%a	IP adresa[3] žadatele
%A	lokální IP adresa
%b	odeslaná data (byte), vyjma HTTP[4] hlaviček
%B	odeslaná data (byte), vyjma HTTP hlaviček, CLF formát (v případě 0 B, nahrazeno „-“)
%c	stav spojení po zpracování požadavku „X“ - spojení ukončeno před úplným zpracováním požadavku „+“ - spojení možná zachováno i po odeslání odpovědi „-“ - spojení bylo ukončeno po odeslání odpovědi
%{Foobar}e	obsah proměnné Foobar
%f	jméno žádaného souboru
%h	jméno (IP) žadatele
%H	protokol žádosti
%{Foobar}i	obsah proměnné Foobar (hlavní data odeslaná v požadavku serveru)
%l	vzdálené jméno logu (získáno z identd, pokud je podporováno)
%m	metoda žádosti
%p	kanonický port, na kterém server obsluhuje požadavky
%P	identifikační číslo procesu potomka, který zpracovává požadavek
%q	řetězec dat zaslaných v požadavku (pokud neexistuje, vrací se prázdný řetězec)
%r	první řádek žádosti
%>s	status žádosti
%t	čas zpracování žádosti (standardní anglický tvar)
%{format}t	čas zpracování žádosti (lze formátovat)
%T	čas obsluhy žádosti ve vteřinách
%u	jméno vzdáleného uživatele (pokud neexistuje, vrací „-“)
%U	URL[5] žádosti (neobsahuje žádné další požadavky)
%v	jméno serveru obsluhující žádost v kanonickém tvaru (používáno při aplikaci virtuálních webových serverů)
Pokud chceme v záznamu použít uvozovky, musí předcházet zpětné lomítko, např. \“%r\“, "Nějaký text", jinak by mohlo dojít k předčasnému ukončení formátování.	

3 IP adresa – Jednoznačná identifikace uživatele internetu
4 HTTP - Hyper Text Transfer Protocol, internetový protokol určený pro výměnu hypertextových dokumentů
5 URL - textový řetězec přesně specifikující umístění zdrojů dokumentů nebo služeb na internetu

Common Log Format

Standardizovaný formát používající následující nastavení:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" logname
CustomLog logs/access_log logname
```

První řádek definuje způsob formátování záznamu a pro další identifikaci je pojmenován jako *logname*, následuje vytvoření souboru *access_log* v adresáři */logs/* s formátováním definovaným pomocí *logname*. Jak již bylo zmíněno, jedná se o standardizovaný formát, který bývá někdy označován zkratkou *CLF*. Reálný záznam může vypadat následovně (jednotlivé položky jsou popsány v Tabulce 2):

```
213.29.7.70 - admin [15/Feb/2007:18:29:33 +0100] "GET
/pictures/ball.gif HTTP/1.1" 200 1541
```

Tabulka 2: Rozpis údajů Common Log Format

213.29.7.70	**(%h)** – IP adresa klienta (případně jméno), který zaslal požadavek na server, pokud je klient umístěn za proxy serverem, je identifikována pouze IP adresa proxy serveru
-	**(%l)** – informace, kterou se nepodařilo získat, identita nebyla zaslána klientským počítačem
admin	**(%u)** – uživatelské ID (jméno) získané pomocí HTTP autentifikace, v případě nezabezpečené stránky je hodnota „-“
[15/Feb/2007:18:29:33 +0100]	**(%t)** – čas úspěšného zpracování požadavku, formát dle [den/měsíc/rok:hodina:minuta:vteřina +zóna]
"GET /pictures/ball.gif HTTP/1.1"	**(\“%r\“)** - metoda požadavku, žádaný objekt a protokol, kterým klient žádá
200	**(%>s)** – kód stavu požadavku, který je zaslán zpět klientovi
1541	**(%b)** – velikost objektu v Byte[6] zasláná zpět klientovi, pokud není nic vráceno, hodnota je „-“

6 Byte – jednotka množství informace (1024 bitů)

Combined Log Format

Další z často používaných formátů, vychází z *Common Log Format* a je rozšířen o dvě položky v záznamu, první z nich je adresa (HTTP request header) definující zdroj požadavku a druhý je uživatelský agent (identifikace používaného internetového prohlížeče na klientské straně). Obecný zápis a konkrétní příklad záznamu:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
agent}i\"" combined
CustomLog log/acces_log combined
```

```
84.181.30.33 - - [12/Feb/2007:12:28:16 +0100] "GET
/pictures/bottom.gif HTTP/1.1" 200 2792
"http://webserver.cz/style.css" "Mozilla/5.0 (Windows; U; Windows
NT 5.1; cs; rv:1.8.0.9) Gecko/20061206 Firefox/1.5.0.9"
```

Prvních sedm položek záznamu je shodných s *Common Log Format*, následující dvě jsou popsány v Tabulce 3. Kombinovaný formát je vhodnější pro získání více informací o klientech navštěvujících webový server.

Tabulka 3: Rozpis údajů Combined Log Format

"http://webserver.cz/style.css"	**(\"%{Referer}i\")** - HTTP hlavička žádosti - adresa, kde je uveden žádaný objekt (na objekt je odkazováno nebo je součástí)
"Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.8.0.9) Gecko/20061206 Firefox/1.5.0.9"	**(\"%{User-agent}i\")** - User-agent HTTP hlavička žádosti – identifikace klientského systému, prohlížeče

Logování virtuálních serverů

Pokud používáme více webových serverů na jedné instanci Apache, je vhodné provést úpravu formátování. K dipozici máme dvě metody:

- ➔ vytvoření logovacího souboru pro každý z virtuálních serverů
- ➔ vytvoření univerzálního logovacího souboru, který pojme všechny spouštěné virtuální servery pomocí zápisu (%v je jméno serveru):

```
LogFormat "%v %l %u %t \"%r\" %>s %b" comonvhost
CustomLog logs/access_log comonvhost
```

1.2 Statistická data

Na získané logovací soubory lze aplikovat statistické analýzy a získat několik důležitých ukazatelů, vhodných pro následnou optimalizaci webových stránek. Optimalizace může být zaměřena na spolehlivost respektive chybovost stránek s využitím HTTP odpovědí či sledování návštěvnosti určitých částí webu. S nárůstem obliby internetu a rozšiřování reklamy a marketingu bylo nutné zavést obecné standardy statistických dat, z tohoto důvodu dobrovolné sdružení IFABC[7] vytvořilo normy pro analýzu a auditování návštevnosti webových serverů. V následujícím přehledu jsou uvedeny možné statistické ukazatele:

- počet záznamů
- počet unikátních návštěv, unikátních IP adres
- délka návštěv rozlišených dle IP adres
- počet odeslaných stránek (specifikovaných dle koncovky souboru)
- počet a typ odeslaných souborů vč. jejich velikosti
- nejvíce žádané stránky
- reference (odkazovače)
- používané prohlížeče a operační systémy
- přehled robotů světových a národních vyhledávačů
- HTTP odpovědi, chybová hlášení

Mezi nejpoužívanější ukazatele patří *unikátní návštevník, unikátní IP adresa, zhlédnutí, reference a HTTP odpovědi,* detailněji jsou rozepsány v následujícím textu.

Zhlédnutí, návštěva, hit

Základní ukazatel, který lze z logovacích souborů získat. Jeho hodnota je inkrementována s každým novým řádkem v souboru či načtením stránky bez ohledu na odesílatele požadavku. K označení se používá především slov „hit“, „záznam“, „page impression“ či „page view“. Tento typ je často používán v internetovém

7 IFABC – mezinárodní organizace definující standardy v oblasti auditu tisku, internetu. http://www.ifabc.org

marketingu a reklamě, především z důvodu vysoké hodnoty oproti unikátním ukazatelům.

Unikátní návštěva, unikátní návštevník

Unikátní návštěva/návštěvník, neboli „unique-user“, představuje statistickou veličinu počítající každého uživatele pouze jednou za určitý časový úsek. V případě přístupů na webový server se standardně používá limit v délce 30 minut.

Identifikace jednotlivých návštevníků může probíhat dvěma způsoby:

- ➔ pomocí IP adresy
- ➔ uložením Cookies[8] na uživatelském počítači s následným kontrolováním dat v databázi na serveru poskytovatele monitorovacích služeb

Identifikace pomocí IP adresy nepatří mezi nejvhodnější metody, především z důvodu masivního rozšiřování sdílených veřejných IP adres, kde je pomocí NAT[9] připojeno více uživatelů (počítačů) pouze jednou veřejnou IP adresou. Tím mohou být data zkreslena, např. nezapočtením druhého uživatele sítě. K dalšímu zkreslení dochází v případě, kdy poskytovatelé internetu dynamicky přidělují IP adresy. Uživatel získá veřejnou IP adresu dočasně, existuje zde možnost její změny v intervalu 30 minut od předchozí návštěvy a poté započtení nové unikátní návštevy, i když se jedná o shodného uživatele. I přes nevýhody je způsob identifikace dle IP často používán, především u systémů analyzujících zpětně logovací soubory webových serverů („offline“ systémy), kde jiná možnost výpočtu unikátních návštevníků není.

Další způsobem, jak určit jedinečného návštevníka, je využití cookies. Během první návštevy serveru je odeslána cookies na stranu uživatele, kde je uložena. Při dalším požadavku si server vyžádá cookies zpět a provede ověření časového intervalu, případně navýší hodnotu ukazatele a aktualizuje hodnotu v cookies. Výhodou je rozlišení více uživatelů používajících jeden počítač, případně jednu IP adresu. S rozšiřováním prvků pro zabezpečení internetových prohlížečů bohužel dochází v některých případech k potlačení cookies (přibližně 9-11 % uživatelů používá blokování cookies), a tak i následnému zkreslení výsledků. V porovnání s identifikací

8 Cookies – malé množství dat odeslané webovým serverem klientovi, při další návštěvě serveru klient odesílá data zpět.

9 NAT -Translation, technologie umožňující skrytí více uživatelů/počítačů za jednu věřejnou IP adresu

dle IP adresy jsou však získaná data stále přesnější. Pro používání tohoto způsobu je nutná existence databáze jednotlivých uživatelů a zásah do webových stránek. Cookies využívají především „online systémy“, které jsou popsány spolu s „offline systémy“ v následující kapitole.

Unikátní IP adresa

Unikátní IP adresa neboli „unique IP“ je definována jako počet jedinečných IP adres, ze kterých byly zaslány požadavky na server za určitý časový interval (lze zvolit v aplikaci zpracovávající záznamy o provozu webového serveru).

Pomocí IP adres lze získat např. informace o geografické pozici uživatele či poskytovatele jeho internetového přípojení. V analýze intranetové sítě lze rozlišovat jednotlivá oddělení ve firmě a sledovat například jejich aktivitu na interních stránkách společnosti. Tato data mohou být dále zpracována v oblasti optimalizace, marketingu a reklamy.

Stránky, soubory

V případě potřeby lze získat typy, počty žádaných (přenesených) souborů a objem. Rozeznávání je řešeno pomocí jednotlivých přípon souborů, které bývají jedinečné a lze je tak jednoduše rozlišit.

Každá odpověď na požadavek obsahuje i velikost přenesených dat, pomocí této informace lze počítat objem přenesených dat v určitém časovém úseku.

Žádosti, reference

Žádosti uživatelů serveru jsou vhodným ukazatelem oblíbenosti určitých stránek na webu. Pomocí těchto informací můžeme optimalizovat obsah webu, přizpůsobit důležitá data a umístit je na nejžádanější stránky, čímž získáváme větší „produktivitu“ webu.

Reference patří mezi nejžádanější ukazatele z pohledu reklamy a marketingu. Vlastník či správce serveru má k dispozici přesné informace o tom, odkud návštevníci přicházejí a lze snadno ověřit účinnost, případně neúčinnost reklamních kampaní, SEO[10] optimalizace a dalších.

10 SEO – SearchEngineOptimalization, upravování obsahu webových stránek s cílem co nejlepšího umístění ve výsledcích internetových vyhledávačů, http://cs.wikipedia.org/wiki/SEO

Operační systémy, prohlížeče

Každý návštevník se při svém požadavku na server identifikuje typem internetového prohlížeče včetně informací o operačním systému. Tyto informace nepatří mezi nejdůležitější, ale s ohledem na obsah webových stránek mohou být užitečné. Například ve srovnání poměru operačních systémů Unix, Linux vs. Microsoft Windows nebo k optimalizaci stránek pro internetové prohlížeče.

HTTP odpovědi, chyby

Webový server odpovídá na každý požadavek návštevníka HTTP kódem, který je jednoznačný a odpovídá situaci na stránkách a nastavení daného serveru. Správci serverů využívají takto zaznamenaná data pro minimalizaci chyb (chybějící stránky, obrázky, chybné přesměrování apod.). Úplný přehled kódů je uveden v Příloze A.

1.3 Dostupná řešení

V dnešní době se zabývá monitorováním chování uživatelů na webových stránkách mnoho organizací, které své služby nabízejí na komerční i nekomerční bázi. Placené služby jsou z větší části určeny především pro společnosti vlastnící webové servery s vysokou návštevností, pro něž je nutné mít okamžitý přehled o uživatelích a stavu stránek, aniž by musely vynakládat svůj čas na správu a organizaci monitorování. Majitelé či správci menších serverů mají k dispozici bezplatná řešení v podobě aplikací vyvíjených skupinově za účelem kvalitního a dostupného řešení pro kohokoliv. Obě varianty, tj. komerční i nekomerční, lze rozdělit na „online“ a „offline“ systémy.

1.3.1 Online systémy

„Online“ monitorovací systémy se vyznačují nezávislostí na platformě webového serveru. Získaná data jsou ukládána do souborů či databází na straně provozovatele dané služby. Umístění informací na vnější straně umožňuje okamžitý přístup ke generovaným datům z celého světa pomocí internetu.

Monitorování je u „online“ systémů realizováno převážně umístěním Javascriptu[11] na webových stránkách (část webu, která je zobrazována neustále).

11 Javascript – interpretovaný programovací jazyk pro webové stránky, používá se především pro vkládání interaktivních prvků

Načtením stránky s tímto skriptem dojde k odeslání uživatelských dat na stranu provozovatele sledování, kde dochází k následnému zpracování. Nevýhodou tohoto řešení je možnost zablokování Javascriptů ze strany uživatele a následné nemožnosti monitorování.

Mezi populární a často vyhledávané „online“ monitorovací služby v České republice patří portály toplist.cz, navrcholu.cz, Google Analytics, případně rozšířené řešení společnosti WebTrends Analytics.

„Online“ monitorovací služby bývají voleny z důvodu jejich nezávislosti/nestrannosti na uživateli. Klient může takto získaná důvěryhodná data prezentovat např. při nabídce reklamního prostoru.

Portál toplist.cz

Služby toplist.cz se dynamicky rozvíjejí od roku 1997 a nyní patří mezi nejvýznamnější na českém internetu. Portál nabízí kvalitní, uživatelsky příjemné a jednoduché rozhraní. Monitorování uživatelů je k dispozici ve dvou variantách, základní a rozšířené. První z jmenovaných je bezplatná a částečné omezená, druhá nabízí komplexní řešení pro náročnější uživatele, ovšem za určitý poplatek.

Monitorování probíhá pomocí viditelného, či neviditelného prvku v podobě obrázku s Javascriptovým kódem, umístěného v kódu webové stránky. Sledovány jsou všechny přístupy prohlížečů podporujících zobrazování grafiky.

toplist.cz má definovány tři základní ukazatele:

- návštevy (visits) – jedná se o uživatele, který webový server navštívil a je rozlišován pomocí IP adresy a cookies za určitý čas, dle obecných pravidel používaných ve světě tento čas činí 900 vteřin
- zhlédnutí (pageviews) – počítá se každé načtení stránky včetně „znovunačtení“ či „reload“ stránky
- unikátní IP – přináší počet jedinečných IP adres uživatelů (v případě uživatelů skrytých za NAT/proxy serverem se využívají cookies)

Možnosti a výhody monitorování:

- hodinové, týdenní a měsíční statistiky (zaznamenávání návštěv a zhlédnutí)
- výpis návštěv podle adres a domén 2. úrovně
- podrobný výpis návštěv (čas, IP adresa, prohlížeč, operační systém, stránka, technické vybavení)
- vstupní stránky, reference (odkazovače), výstupní stránky
- vyhledávače odkazující na daný web (za den, měsíc, rok)
- cesta návštěvníka po serveru za den
- měsíční export získaných dat do PDF
- ovládací prvek jako neviditelný bod (bez nutnosti zobrazení loga monitorovací služby)

- porovnání s konkurenčními weby ve vlastním katalogu

Nevýhody:

- detailnější statistiky dostupné pouze po zaplacení určitého poplatku
- nemožnost exportu dat do formátů *.csv či *.xls vhodných pro další zpracování
- nemožnost rozlišovat typy uživatelů v zabezpečené části webových stránek systémem Apache

Portál navrcholu.cz

Služby navrcholu.cz se z větší části shodují s předcházejícím portálem. Na českém internetu působí přibližně shodnou dobu. Na rozdíl od toplist.cz nabízí navrcholu.cz čtyři tarify monitorování, od bezplatné až po individuální verzi dle přání zákazníka.

Měření probíhá stejnou metodou jako u předešlého konkurenta, na webové stránky je umístěn neviditelný, či viditelný obrázek, pomocí něhož se zasílají data. Výhodou je nezapočítávání přístupů robotů celosvětových vyhledávačů.

Základní ukazatele při monitorování (vycházejí ze standardů organizace IFABC):

- návštevník (unique user) – jedinečný návštevník identifikovaný pomocí IP adresy a cookies
- návšteva (visit) – připočítává se každá návštěva, ne však, pokud se navrátí návštevník v intervalu 900 vteřin od své poslední návštevy
- zobrazená stránka (page impression) – každá stránka zobrazená návštevníkovi

Možnosti a výhody monitorování:

- až 53 specifických statistik, možnost individuálního tarifu
- archivace dat po dobu až 12 měsíců
- počet návštevníků (denní, měsíční a roční přehledy)

- počet aktivních uživatelů na webu (denní, týdenní a měsíční)
- počet návštev (denní, týdenní a měsíční)
- vstupní stránky, reference, výstupní stránky, technické vybavení
- časové zdržení uživatele na webových stránkách
- měsíční statistiky do formátu PDF
- porovnání s ostatními weby ve vlastním katalogu

Nevýhody:

- bezplatná verze omezena 100 000 zobrazenými stránkami měsíčně

Google Analytics

Služby celosvětového vyhledávače google.com se rozrůstají každým dnem a nyní nabízejí i možnost sledovat a monitorovat počínání návštevníků na webových stránkách. Výhodou je mezinárodní působení služby, bezplatnost a silná základna, spravující tuto službu, zaručující kvalitu a dostupnost.

Podobně jako předcházející služby i Google Analytics využívá vloženého Javascriptu do kódu webových stránek. Data mohou být opět zkreslena zamezením spouštění skriptů na straně uživatelského počítače.

Portál umožňuje tvůrcům webových stránek optimalizovat a upravovat obsah dle získaných statistických dat. Administrátor může analyzovat provedené úpravy či realizovat reklamní a marketingové strategie se službou Google AdWords. Mimo podporu reklamy a marketingu internetových stránek jsou počítáni návštevníci, sledováni odkud přišli, jak dlouho zůstali a zda-li provedli to, co od nich očekáváme (např. dokončení objednávky v elektronickém obchodování).

Výhodou je podpora reklamních a marketingových strategií, jednoduché rozhraní, celosvětová dostupnost, bezplatnost, možnost nasazení od malých až po velké webové servery.

1.3.2 Offline systémy

„Offline“ monitorování ve většině případů spočívá ve využívání nadstaveb webových serverů nebo samostatných aplikací, které přistupují přímo k logovacím souborům serveru, což vytváří výhodu oproti „online“ systémům. Nehrozí zde omezení v podobě zablokování Javascriptu či nenačtení obrázku na straně uživatele (návštěvníka). Logovací soubory obsahují všechny požadavky jednotlivých uživatelů na námi monitorovaný webový server. Obsah logovacích souborů je ovlivněn nastavením serveru, příklad je uveden v kapitole 1.1.2 o logování serveru Apache.

Provoz a nastavení těchto aplikací není již tak komfortní jako v případě „online“ systémů, je nutné být více znalý, především v nastavení serveru, na kterém běží i webový server (vyjma externích aplikací, které využívají pouze vytvořené logovací soubory). Dostupnost dat je omezena připojením serveru k síti internet. Nelze získat aktuální data z dané chvíle, vždy se provádí zpětná analýza vytvořených dat, v tomto směru je výhodné použití „online“ služeb, které provádí okamžité sledování.

Existuje celá řada monitorovacích systémů pro různé webové servery, vzhledem k velkému podílu „open-source“ serveru Apache ve světě internetu, budou zmíněny nadstavby či aplikace právě pro uvedený server.

AWStats

AWStats - bezplatný software pro analýzu webových služeb, umožňující zpětně posuzovat požadavky jednotlivých návštevníků (webové požadavky, streamovaná média[12], emaily nebo služby na bázi FTP[13]). AWStats přistupuje přímo k logovacím souborům serveru, ze kterých jsou generována data pro pozdější zobrazení webové prezentace získaných výsledků.

U systému je zakomponována podpora nejpoužívanějších serverů v podobě Apache, WebStar nebo IIS (Internet Information Services společnosti Microsoft). Aplikace je napsána pomocí programovacího jazyka Perl, lze ji tak případně upravovat na většině systémů. AWStat lze spustit formou CGI skriptu přímo na serveru nebo jako samostatnou aplikaci na pracovní stanici a provádět analýzu uloženého logovacího

12 Stream – protokoly pro přenos multimedální dat, např. video sekvencí, audio sekvencí
13 FTP – FileTransferProtocol, protokol určený pro přenos souborů po síti internet, intranet, výhodou je nezávislost na platformě

souboru ze vzdáleného serveru.

Aktuální verze AWStat 6.6 naznačuje neustálý vývoj (od verze 6.5 došlo k výraznému pokroku) a i žádanost produktu. Aplikaci nelze snad nic vytknout a je težké jí konkurovat, nevýhodou je stále absence sledování jednotlivých uživatelů.

Možnosti a výhody:

- ➔ analýza logovacích záznamů webových, FTP a e-mailových serverů
- ➔ možnost obnovovat statistiky pomocí příkazové řádky, případně přes internetový prohlížeč
- ➔ monitorování návštevníků, unikátních návštěvníků, času stráveného na serveru (založeno na načtených stránkách)
- ➔ monitorování robotů (integrovaná databáze existujících), autorizovaných uživatelů
- ➔ rozlišování návštevníků dle území, měst, internetových poskytovatelů
- ➔ monitorování zatížitelnosti jednotlivých hodin, dnů v týdnu, často dotazovaných stránek, vstupních a výstupních stránek
- ➔ detekování jednotlivých typů souborů, operačních systémů a internetových prohlížečů návštěvníků
- ➔ monitorování vyhledávacích služeb na webu (použitá klíčová slova, řetězce)
- ➔ generování denní, týdenní a měsíční statistiky
- ➔ generování statistik chybových odpovědí webového serveru
- ➔ možnost analyzovat stažený logovací soubor na pracovní stanici
- ➔ distribuováno pod GPL (General Public License)
- ➔ dobrá uživatelská podpora, příjemné uživatelské rozhraní

Nevýhody:

- ➔ nemožnost sledování konkrétního uživatele
- ➔ omezenost dostupných dat daných obsahem logovacího souboru

WebAlizer

WebAlizer - aplikace pro analýzu přístupu a využití webového serveru. Disponuje jednoduchým grafickým rozhraním realizovaným pomocí webových stránek, kde jsou zobrazovány i výsledky. I přes ukončený vývoj (roku 2002) je stále často využívána správci serverů.

Pro rychlé zpracování je aplikace napsána v programovacím jazyce C a plně podporuje logovací soubory serveru Apache.

Možnosti a výhody:

- ➔ jednoduchost, rychlost, bezplatnost
- ➔ hodinové, denní a měsíční statistiky návštěvnosti
- ➔ monitorování propustnosti dat za určité období, nejžádanější stránky
- ➔ monitorování uživatelských IP adres
- ➔ vstupní stránky, reference, výstupní stránky
- ➔ distribuováno pod GPL (General Public License)

Nevýhody:

- ➔ zastavení vývoje aplikace
- ➔ slabší uživatelské rozhraní

Webtrends

Webtrends Analytics je komplexní komerční řešení analýzy chování uživatelů webových serverů určené pro velké a střední společnosti. Nabízí několik řešení „na klíč“, zaměřených určitým směrem optimalizace a sledování (elektronické obchodování, marketing, apod.).

Výhodou tohoto řešení je výběr z „online“ i „offline“ řešení. V prvním případě stačí použít Javascriptový kód na webových stránkách, vše ostatní je řešeno na domacím serveru společnosti nabízející tento produkt. V druhém je nabízeno softwarové řešení, zde je již nutné pořízení vlastního hardwaru, kde probíhá zpracování dat. Poté lze analyzovat jak pomocí online Javascriptového kódu, tak i z logovacích

souboru serveru. Obě služby jsou dostupné na dálku z internetu, což umožňuje okamžitě zasáhnout do obsahu stránek a provádět případné změny či optimalizace.

Standardní možnosti:

- ➔ analýza reklamy na vlastních stránkách
- ➔ statistiky vyhledávačů, návštevníků, domén
- ➔ geografické statistiky návštevníků
- ➔ statistiky vyhledávání na vlastních stránkách
- ➔ přehledy žádaných stránek, souborů, procházení stránek
- ➔ statistiky prohlížečů, systémového prostředí návštevníků
- ➔ sledování výkonnosti webového serveru

2 Praktická část

2.1 Návrh aplikace

Cílem praktické části diplomové práce je návrh aplikace, umožňující zpětnou analýzu logovacího souboru webového serveru Apache, který byl popsán v kapitole 1.2. Vzhledem k předchozímu dělení systémů pro analýzu získaných dat se jedná o „offline“ aplikaci, která bude přistupovat k vytvořeným logovacím souborům a následně je analyzovat podobně jako AWStat či Webalizer. Podpora je zajištěna pro stávající standardizováné logovací soubory *Common Log File* a *Combined Log File*.

Pro realizaci aplikace jsem zvolil volně dostupná řešení. Uživatelské prostředí je vytvořeno pomocí značkovacího jazyka pro tvorbu hypertextových dokumentů XHTML a grafické zobrazení kaskádovými styly CSS. Tvorba, složení webové prezentace a výpočty jsou tvořeny skriptovacím jazykem PHP za podpory databáze MySQL, do které jsou převedeny logovací soubory. Export dat je k dispozici ve formátu XML. Jednotlivá řešení jsou popsána v následujícím textu.

Nevýhodou realizace pomocí databáze MySQL je rychlost práce s daty v databázi, kdy jsme omezeni výkonem serveru a objemem dat, zmíněná konkurenční řešení jazykem Perl jsou rychlejší, ale uživateli neposkytují možnost náhledu do získaných logovacích dat, což nám databáze v plné míře umožňuje.

XML

eXtensible Markup Language, rozšiřitelný značkovací jazyk, lze považovat za fenomén dnešní doby. S pokračujícím nárůstem informací ve světě internetu je nutné třídit informace již u zdroje. Označování obsahu usnadňuje a zefektivňuje práci vyhledávačů. Toto přináší XML vytvořené pro popsání obsahu dat (oproti HTML, kde je kladen důraz na zobrazení dat a na způsob, jak data vypadají). Jednotlivé tagy (značky) nejsou předdefinované, každý uživatel vytváří své, čímž se přesně popisují zobrazovaná data. V případě tvorby webových stránek pomocí XML je nutné používat kaskádové styly pro konečné definování způsobu zobrazení obsahu. XML lze používat nejen v oblasti hypertextových dokumentů, ale i v oblasti průmyslu, např. při předávání dat mezi aplikacemi ve výrobě.

Pro představu uvádím příklad kódu XML, kde je vidět jednoduchost a čitelnost záznamu:

```
<?xml version="1.0"?>
<record>
      <ip>10.1.1.4</ip>
      <name>comp.internet.org</name>
      <user>john day</user>
</record>
```

Při tvorbě XML dokumentů je nutné dbát několika pravidel, základním je vytvoření hlavního počátečního a ukončovacího tagu (značky) dokumentu, který bude obsahovat jednotlivé záznamy (např. record). Každý tag musí být uzavřený, např. IP adresa uzavřená v předcházejícím příkladě značkou <ip> a </ip>. Dále uvádíme definici verze XML dokumentu. Vzhledem k tomu, že je XML citlivá na velikosti písmen, doporučuje se používat pouze malá písmena, předchází se tak případným problémům.

XHTML

eXtensible Hypertext Markup Language, neboli rozšiřitelný hypertextový značkovací jazyk, sloužící k tvorbě hypertextových dokumentů jako jsou internetové prezentace. Vychází z XML a HTML, s nimiž si zachovává kompatibilitu. Zavedením XHTML došlo k upřesnění pravidel a standardů pro tvorbě hypertextových dokumentů, čímž se urychlilo a automatizovalo zpracování (prohlížeči, vyhledávači, mobilními zařízeními apod.).

CSS

CSS neboli Cascading Style Sheets, jedná se o souhrn metod a vlastností pro úpravu vzhledu webových stránek. Hlavní výhodou je nastavení formátování jednotlivých prvků v externím souboru. Zásah do jediného souboru se promítne v celém vzhledu dokumentu. Dalším vylepšením je očištění zdrojového kódu webové stránky od formátovacích atributů a zlepšení čitelnosti takového kódu.

Rozdíl standardního formátování a CSS je uveden u definice odstavce, první část značí standardní HTML kód, druhá XHTML s náhledem do souboru se styly:

```
<p style="color:red;font-weight:bold">Text</p> - HTML
<p class="cerveny">Text</td> - XHTML
p { color:red; font-weight:bold; font-family:Arial;} - CSS
```

PHP, TemplatePower

Hypertext Preprocessor – skriptovací jazyk pracující na straně serveru, klient vždy obdrží pouze výsledky. Výhodou je jednoduchost (již se základy lze vytvářet zajímavé skripty), dostupnost a podpora ze strany vývojářů a uživatelů. Samostatný skriptovací jazyk je doplněn o TemplatePower, šablonovací systém, který odděluje kód jazyka PHP od XHTML, samotný zdrojový kód je pro případnou kontrolu či úpravu přehlednější. PHP je využíván na webových serverech právě s Apache.

MySQL

Relační databázový systém využívá tabulek, ve kterých jsou umístěné záznamy v podobě řádků a sloupců. Každý sloupec v tabulce je jedinečný a obsahuje data. Při nutnosti získání dat jsou tabulky procházeny a dle kritérií jsou vybírána data nebo prováděny operace. Výhodou je šíření pod licencí open-source a podpora několika platforem, např. C, C++, Java, Perl, PHP, Python, Tcl, Visual Basic, .NET.

PHP disponuje již vytvořenými funkcemi pro práci s databází MySQL a opět je zde zaručena dobrá kompatibilita chodu PHP, MySQL a serveru Apache.

2.2 Funkce aplikace

2.2.1 Statistiky

Aplikace disponuje obecnými statistikami získanými při samotném importu dat do databáze, především se jedná o obecné statistické údaje a užitečné informace o stavu webových stránek.

- ➔ celkový počet záznamů (požadavků)
- ➔ celkový počet unikátních návštěv – využita metodika s časovým koeficientem 30 minut, započtena je pouze návštěva, u které uběhlo více než 30 minut od posledního požadavku daného uživatele
- ➔ celkový počet unikátních IP adres s možností zobrazení výpisu
- ➔ celkový počet odeslaných stránek
- ➔ celkový počet odeslaných dat v Byte

- jednotlivé počty HTTP odpovědí webového serveru s možností detailního výpisu
- časové rozmezí dostupných záznamů
- výpis nejpoužívanějších internetových prohlížečů (prvních 5)
- výpis odkazovačů na webové stránky (prvních 5)
- výpis nejžádanějších požadavků (prvních 5)

Dále statistiky rozdělují data dle jednotlivých měsíců nalezených v databázi:

- celkový počet záznamů, unikátních návštěv, unikátních IP adres, odeslaných stránek, dat v Byte
- výpis HTTP odpovědí odeslaných v daném měsíci
- výpisy nejpoužívanějších prohlížečů, odkazovačů (referencí) a žádostí s možností nastavení počtu zobrazených dat
- výpis počtu unikátních návštěv pro jednotlivé dny zvoleného měsíce

2.2.2 Sledování uživatelů

Monitorování uživatelů je klíčovou funkcí, která nám umožňuje získat výpis všech jedinečných uživatelů dle IP adres včetně počtu žádostí. Výpis disponuje IP adresou, DNS jménem, pod kterým vystupuje IP adresa na internetu, počtem záznamů pro danou IP a možností aktivace sledování uživatele, čímž je údaj automaticky zobrazen na prvních místech výpisu s jednoznačnou identifikací. V seznamu dostupných adres lze provést hledání na základě IP adresy, případně klíčového slova vyskytujícího se v záznamu identifikace prohlížeče (funkce je dostupná pouze pro logovací formáty typu Combined Log File), tuto volbu lze využít k hledání robotů vyhledávačů apod.

Ke každému jedinečnému záznamu IP adresy získáváme tyto možnosti:

- výpis údajů o datu, času záznamu, požadavku, typu odpovědi, velikosti dat odeslaných uživateli, referenci a druhu prohlížeče
- doménový název pro vybranou IP adresu
- možnost třídit data dle času, data, klíčového slova prohledávajícího

požadavky či typu HTTP kódu

- ➔ statistiky pro vybranou IP adresu, počet záznamů, odeslaných dat a počet jednotlivých odpovědí HTTP kódu, pro případné hledání chyb ve struktuře stránek

2.2.3 Import, nastavení

Pro organizaci jednotlivých logovacích souborů je připravena sekce import a nastavení, je zde možnost nadefinování názvu tabulky, pod kterou budou data uschována v databázi, a uvedení přístupové cesty k logovacím souboru umístěného na webovém serveru včetně určení formátu souboru (podpora Common Log File a Combined Log File). V případě existence záznamu je uživatel dotázán, zda-li data přidat či vytvořit nová.

Dostupné tabulky s převedenými daty jsou zobrazeny v uceleném výpisu. Aktivní tabulka je označena a je zde možnost volby ostatních záznamů či jejich kompletní odstranění z databáze.

Sekce nastavení poskytuje možnost volby počtu řádků k zobrazení v jednotlivých sekcích aplikace jako jsou detailní výpisy záznamů, seznamy IP adres, statistiky NEJ.

2.2.4 Export dat

Pro další zpracování získaných dat je k dispozici možnost exportu. Základem je generování vybraných dat do formátu XML, který je následně nabídnut ke stažení uživateli. Exportování dat lze provádět v sekci sledování uživatelů (celkový výpis dostupných jedinečných IP adres nebo detailní výpis záznamů zvolené IP adresy). Další možností je export záznamů jednotlivých HTTP kódů a jejich detailních záznamů.

2.3 Realizace

2.3.1 Použitá řešení

Aplikace je programována a optimalizována na bázi skriptovacího jazyka PHP, testována a provozována na verzi PHP 4.4.4. Pro oddělení jednotlivých kódů jazyků XHTML a PHP jsem využil TemplatePower. Toto řešení je distribuováno pod licencí GPL, umožňuje tak bezplatné nasazení a uživatelský komfort. Existují i jiná podobná řešení, ale práce s TemplatePower je dle mého názoru jednoduchá a rychlá.

TemplatePower

TemplatePower umožňuje komfortní programování webových stránek složených z jazyků PHP a HTML či XHTML. Základem je tisknutí šablon doplňených o proměnné. Šablony mohou obsahovat dynamické bloky, které se využívají např. v tabulkách. Programátor tak může upravovat zdrojový kód v PHP bez nutnosti sledovaní a upravování XHTML kódu. Opačně lze upravovat vzhled stránek pomocí kaskádových stylů nebo změn v hypertextovém kódu bez nutnosti zásahu do hlavního skriptovacího kódu.

Základním kamenem je objektová třída *class.TemplatePower.inc.php*. Obsahuje všechny potřebné procedury a funkce pro tvorbu šablonovacích objektů. Je nutné ji incializovat při každém načtení skriptovacího kódu, který bude realizovat tisk hypertextové šablony. V případě mé aplikace je incializace zajištěna skriptem *index.php* následující formou: *include_once('./classes/class.TemplatePower.inc.php')*. Při používání je nutné znát několik základních funkcí, první z nich je vytvoření objektu s výběrem šablony, realizace probíhá voláním *$tpl = new TemplatePower('./cesta_k_sablone/sablona.tpl')* čímž vytvoříme nový objekt s vybranou šablonou. Následuje inicializace objektu (volání konstruktoru) pomocí *procedury $tpl->prepare()* a přiřazení proměnných příkazem *$tpl>assign(„jmeno_promenne“,hodnota)* lze přiřazovat i více hodnot naráz zápisem *$tpl->assign(Array(„prom_1“ => hod_1,„prom_2“ => hod_2))*, po přiřazení všech potřebných proměnných lze šablony předat internetovému prohlížeči pomocí procedury *$tmp->printToScreen()*.

Příklad použití systému TemplatePower je ukázán na následující ukázce pro tisk tabulky v XHTML, první částí je skriptovací kód PHP *test.php*:

```
<?php
  // načtení třídy TemplatePower
  include_once('./classes/class.TemplatePower.inc.php');

  // vytvoření a vykreslení objektu
  $tpl = new TemplatePower( './templates/test.tpl' );
  $tpl->prepare();
  $tpl->assign("id",$tb_name);
  for ($i=0;$i<3;$i++) {
    $tpl->newBlock("row");
    $tpl->assign(Array("num" => $i,"value" => $foo[$i]));
  $tpl->printToScreen();
?>
```

druhou částí je šablona umístěná v souboru *test.tpl*:

```
<table id="{id}" cellpadding="0">
  <!-- START BLOCK : row -->
  <tr>
    <td>{num}</td><td>{value}</td>
  </tr>
  <!-- END BLOCK : row -->
</table>
```

výsledkem je výpis XHTML kódu pro internetový prohlížeč v podobě:

```
<table id="tabulka" cellpadding="0">
  <tr>
    <td>0</td><td>car</td>
  </tr>
  <tr>
    <td>1</td><td>type</td>
  </tr>
  <tr>
    <td>2</td><td>model</td>
  </tr>
</table>
```

Pro programování hypertextových stránek jsem zvolil standard XHTML 1.0 Strict s podporou kaskádových stylů CSS 1.0 s cílem úplné validity zdrojových kódů, kterou lze ověřit na domovských internetových stránkách konsorcia W3C http://www.w3c.org, které se zabývá zaváděním standardů jazyků používaných při tvorbě webových stránek a dokumentů.

Úložiště převedených dat z logovacích souborů obstarává databáze MySQL, jak jsem již zmínil v představení jednotlivých řešení, výhodou je dostupnost

a především časté nasazení v kombinaci se serverem Apache s podporou PHP. Aplikace je provozována na verzi MySQL 4.0.27.

Případný export dat je zajištěn pomocí formátu XML verze 1.0, při volbě exportu je uživateli nabídnut soubor ke stažení.

Databáze

Databáze je uspořádaná množina informací neboli dat, které jsou v našem případě uloženy v tabulkách.

Pro další práci s logovacími soubory bylo nutné připravit tabulky do databáze MySQL pro budoucí importovaná data, vypočtená statistická data a nastavení. Základní logovací soubory webových serverů Apache většinou obsahují velké množství dat ve velikosti desetitisíců až miliónů řádků (záleží na vytíženosti daného serveru), proto jsem pro každý takový logovací soubor zvolil jednu tabulku, obsahující všechna potřebná data, která lze získat ze zvoleného logovacího souboru. Rozdělení souboru do více tabulek nepovažuji za výhodné, vznikla by potřeba je následně spojovat při komplexním vyhledávaní dat. Všechny tabulky, které jsou vytvořeny nebo budou vytvořeny v průběhu používání aplikace, jsou pojmenovány „log_jmenotabulky“, čímž jsou jednoznačně rozeznatelné od ostatních tabulek v databázi.

Základní struktura tabulky pro data vypadá dle rozvržení v Tabulce 4

Tabulka 4: Rozvržení databázové tabulky pro logovací data

Název sloupce	***Datový typ***	***Popis***
id	int(11)	jedinečné označení každého záznamu
ip	varchar(25)	IP adresa návštevníka
accesstime	datetime	čas a datum požadavku
request	varchar(255)	žádost
requesttype	varchar(25)	typ žádosti (GET, POST,..)
code	char(3)	HTTP odpověď serveru
bytes	int(11)	počet odeslaných dat v Byte
reference	varchar(255)	odkazovač, reference (odkud návštevník dorazil)
browser	varchar(255)	typ prohlížeče návštevníka

Každá tabulka v databázi musí mít definovaný primární klíč a index. V případě tabulky pro uložení logovacích dat je primárním klíčem zvolen sloupec *id*, který slouží zároveň jako index. Jelikož se v záznamech často hledá pomocí času, dalším indexem je sloupec *accesstime*. Indexy obecně urychlují práci databáze nad zvolenou tabulkou. Pro odlišení tabulek se záznamy od ostatních je do názvu přidáno písmeno X, jméno tabulky má tvar „log_Xjmeno“.

Další podmínkou je přítomnost tabulky umožňující uložení generovaných statistických dat získaných z tabulek logovacích dat. Základní struktura je vyobrazena v Tabulce 5, mající v databázi pevně dáno jméno *log_statdata*. V tabulce je vyhledáváno pouze pomocí sloupce *id*, který nabývá jedinečných hodnot, proto je zvolen jako primární klíč a index.

Tabulka 5: Rozvržení databázové tabulky pro statistická data

Název sloupce	***Datový typ***	***Popis***
id	varchar(20)	identifikace, shodné s názvem tabulky zdrojových dat
records	bigint(20)	počet záznamů
visits	smallint(6)	počet unikátních návštěv
ip	smallint(6)	počet unikátních IP adres
bytes	int(11)	počet odeslaných dat v Byte
pages	smallint(6)	počet odeslaných stránek
http	varchar(200)	jednotlivé HTTP kódy a jejich počet
day_data	text	statistická data k jednotlivým dnům

Předchozí databázovou tabulku *log_statdata* doplňuje tabulka nejžádanějších a nejpoužívanějších informací (žádosti, reference, prohlížeče) s názvem *log_statdata_top*. Složení je uvedeno v Tabulce 6. Při hledání je používáno výhradně sloupce *tab*, který je indexován. Primárním klíčem je opět sloupec *id*.

Tabulka 6: Rozvržení databázové tabulky pro rozšířená statistická data

Název sloupce	***Datový typ***	***Popis***
id	smallint(6)	identifikace
tab	varchar(20)	jméno hlavní tabulky
type	varchar(20)	druh záznamu
value	varchar(200)	hodnota
count	mediumint(9)	počet

Pro dostupnost doménových jmen jednotlivých IP adres jsem přidal tabulku *log_ip2name*, toto řešení jsem zvolil z důvodu časové náročnosti funkcí, které umožňují převod z IP adresy na její doménové jméno. Tabulka je využívána i pro funkci „Sledování“, indexem a primárním klíčem je zvolen sloupec *ip*. Struktura je uvedena v Tabulce 7.

Tabulka 7: Rozvržení databázové tabulky pro IP adresy a jejich doménová jména

Název sloupce	***Datový typ***	***Popis***
ip	varchar(15)	IP adresa
ipname	varchar(40)	doménové jméno pro IP adresu
ipactive	tinyint(4)	aktivace sledování

Poslední potřebná tabulka pro chod aplikace nese název *log_configuration* a obsahuje všechny potřebné informace týkající se nastavení. Vyznačuje se pouze jedním řádkem, v němž jsou hodnoty průběžně nastavovány. Struktura je v Tabulce 8. Primárním a jedinečným klíčem je sloupec *id*.

Tabulka 8: Rozvržení databázové tabulky pro nastavení

Název sloupce	***Datový typ***	***Popis***
id	int(11)	identifikace nastavení
tab	varchar(32)	název aktivní tabulky
max_row_ip	smallint(6)	maximální počet řádků u výpisu IP adres
max_row_record	smallint(6)	maximální počet řádků u obecného výpisu informací
max_row_top	smallint(6)	maximální počet řádků u TOP statistik

Spojení s databází je řešeno pomocí *connectDB()* ze souboru *functions.php*, tato funkce naváže spojení s databází uvedenou v konfiguračním souboru */conf/configuration.php*, v případě neúspěchu je oznámeno chybové hlášení. Při potřebě dat z databáze je spojení navázáno a po ukončení práce opět odpojeno pomocí funkce *mysql_close()*.

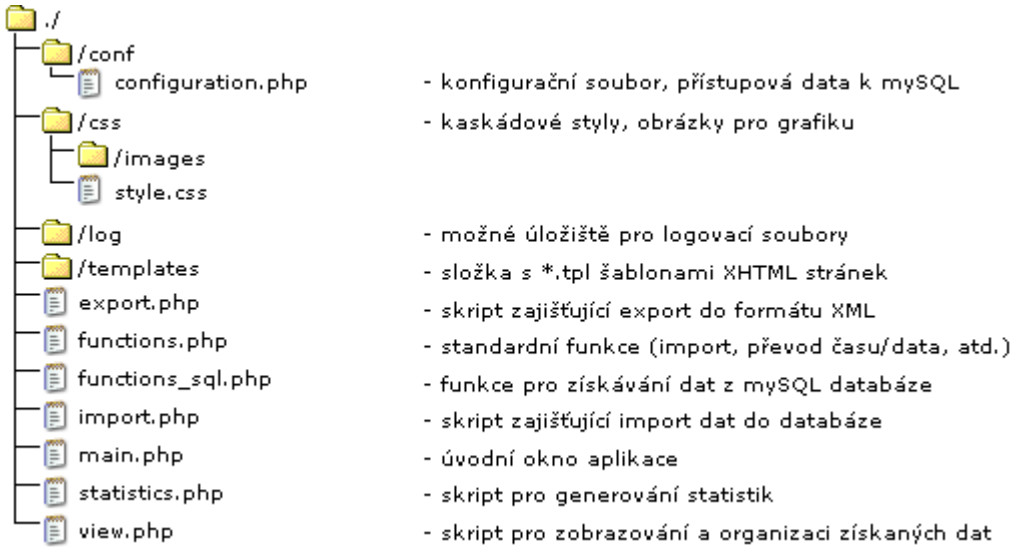
2.3.2 Zabezpečení

Aplikace není v zakladním nastavení zabezpečena z hlediska přístupu uživatelů, obsah dat není tak vysoce citlivý. Předpokládá se nasazení v rámci správcovské oblasti na serveru, kde je přístup umožněn pouze určitým osobám. Jelikož je aplikace připravena pro použití na serveru Apache, existuje jednoduchá možnost zamezení přístupu nepovolaným osobám umístěním souboru *.htaccess*[14] do kořenové struktury aplikace, kdy při každém pokusu o načtení stránky následuje zobrazení formuláře pro přihlášení uživatele. Zabezpečení pomocí HTTP autentifikace doporučuji použít v případě složky */conf*, kde je uložen konfigurační soubor s přístupy do databáze.

14 HTTP autentifikace: jednoduchý a účinný způsob zabezpečení přístupu využívající dotazovací formuláře daného prohlížeče bez potřeby jej programovat, záleží na nastavení serveru Apache

2.3.3 Struktura

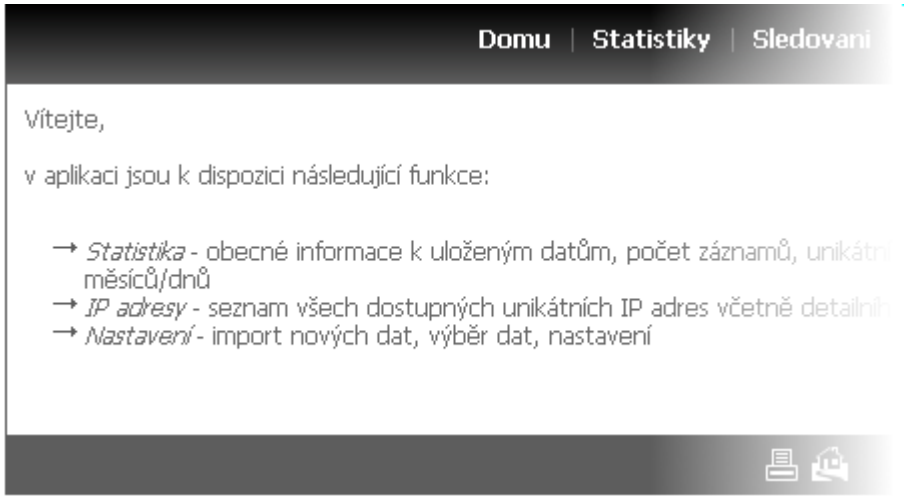
Umístění jednotlivých skriptů a souborů je vyobrazeno na stromové struktuře viz Obrázek 1. Z uvedené struktury je dle názvů zřejmé, co jednotlivé skripty dělají. Rád bych zmínil soubor *functions.php*, který obsahuje řadu vytvořených funkcí, především zpracování a převod dat z logovacího souboru do formy vhodné pro databázi, úpravu řetězců či časových údajů. Soubor *functions_sql.php* obsahuje převážně funkce pro hledání dat v databázi, jejich zpracování pro statistické účely apod., jednotlivé funkce jsou okomentovány ve zdrojovém kódu.



Obrázek 1: Souborová struktura aplikace

2.3.4 Spouštění

Základní spuštění aplikace probíhá načtením skriptu *index.php* internetovým prohlížečem. Při jeho volání je provedeno získání základních parametrů uložených v databázi. Podrobný rozpis způsobu uložení dat je uveden v následující kapitole *Import,nastavení*.



Obrázek 2: Náhled domovské stránky aplikace

Po načtení konfiguračních proměnných je zobrazeno základní prostředí aplikace (viz Obrázek 2). Ta se skládá ze 3 částí (hlavička, tělo, patička). První z nich je hlavička aplikace, která je generována do hypertextového kódu pomocí TemplatePower s použitím šablony */templates/app_top.tpl*. Obsahuje standardní hypertextové odkazy na funkční části aplikace. Poté je načítáno tělo aplikace, definované pomocí proměnné *$action.* Každým znovunačtením stránek dochází k její kontrole a v případě shodnosti s uloženými hodnotami je inializováno načtení daného skriptu funkcí *include(),* v opačném případě je volána základní domovská stránka (chráněno proti případnému pokusu o načtení cizího skriptu využitím funkce *include*()). Např. pro volání nastavení aplikace je zápis odkazu realizován *index.php?action=import*. Poslední částí je generování patičky stránky opět pomocí TemplatePower s využitím šablony */templates/app_bottom.tpl*.

2.3.5 Import, nastavení

Základním skriptem pro zpracování importu a nastavení je *import.php*, který je po nalezení proměnné *$action=import* přidán jako tělo aplikace.

Převod existujících dat

Základním a jediným zdrojem dat pro import jsou logovací soubory webového serveru Apache, jedná se o textové soubory obsahující záznamy o aktivitě uživatelů (návštevníků) internetových stránek. Každý záznam je uveden na jednom řádku, náhled souboru vypadá následovně:

```
213.29.7.70 - - [01/Jan/2007:00:00:57 +0100] "GET / HTTP/1.1"
404 1054 "-" "holmes/3.10 (http://morfeo.centrum.cz/bot)"

62.245.71.231 - - [01/Jan/2007:00:01:05 +0100] "GET / HTTP/1.0"
200 15762 "-" "check_http/1.81 (nagios-plugins 1.4)"

62.245.71.231 - - [01/Jan/2007:00:06:05 +0100] "GET / HTTP/1.0"
200 15762 "-" "check_http/1.81 (nagios-plugins 1.4)"
```

Význam jednotlivých položek byl vysvětlen v kapitole 1.1.2, z uvedeného příkladu je vidět problematičnost „rozškatulkování“ jednotlivých záznamů pro další uložení do databáze. Jednotlivé části záznamu nejsou specificky odděleny, až na čas, typ požadavku či verzi prohlížeče. V tomto případě jsem se rozhodl využít regulárních výrazů pro získání jednotlivých údajů.

Přístup k souborům je zajišťován běžnými funkcemi skriptovacího jazyka PHP, otevření probíhá pomocí *$fp = fopen("/log/v_logdata.log","r")*, kde funkce *fopen()* zajistí otevření souboru */log/v_logdata.log* pro čtení, začne načítání jednotlivých řádků souboru až po jeho konec. Každý získaný řádek je pomocí funkce *parseLog()* analyzován a postupně rozdělen do pomocného pole $x[]:

- ➔ vyhledání IP adresy je realizováno pomocí regulárního výrazu *"/([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)/"* a *funkce preg_split*[15], kde je hledána čtveřice trojic oddělených třemi tečkami, získaná hodnota je uložena do pomocného pole *$x[0]*, zbývající řetězec do proměnné *$all*.

15 preg_split($reg,$string,p,q) – funkce PHP, vrací hledaný řetězec shodný s regulárním výrazem $reg v řetězci $string

- ➔ následuje vyhledání času/data záznamu, výhodou je oddělení pomocí hranatých závorek od ostatních záznamů. Zde využívám regulárního výrazu *"/\[([^\]]*).+/"* spolu s funkcí *preg_match()*[16], získanou hodnotu času/data je nutné před uložením do databáze převést na formát shodný s SQL, tzv. datový typ *datetime* (pro další bezproblémové používání v databázi). K tomuto účelu jsem vytvořil funkci *parseTime()*, ta umožňuje převod „01/Jan/2007:00:01:05 +0100“ na formát „2007-01-01 00:01:05“, nově formátovaný čas je opět uložen do pole *$x[1]* a zbývající řetězec do proměnné *$all*.
- ➔ dalším záznamem v pořadí je požadavek uživatele s verzí protokolu, jelikož se vyskytuje více požadavků (GET, POST, HEAD), je nutné je rozlišit. Regulárním výrazem je *"/ \"GET (.[^\"]+)/"*, kde je GET postupně nahrazen ostatními typy žádostí do doby než *preg_match()* vrátí hodnotu TRUE (záznam nalezen), výsledek je opět uložen do pole, žádost do *$x[2][0]* a druh žádosti do *$x[2][1]*.
- ➔ hledání pokračuje kódem HTTP odpovědi serveru, výhodou je jednoznačnost v podobě třímístného čísla, regulární výraz je "/ ([0-9]{3}) /", nalezený kód je uložen do *$x[3]* a zbývající řetězec do proměnné *$all* jako v předcházejících případech.
- ➔ pokud máme k dispozici logovací soubor formátu *Combined Log File*, probíhá další analýza řetězce v proměnné *$all*, hledáme výraz uzavřený uvozovkami (reference, následně prohlížeč), formou reg. výrazu *"/\"(.[^\"]+|-)/"*, výsledky jsou uloženy postupně do pole *$x[4] a $x[5]*.

Po dokončení rozložení řádku logovacího souboru je realizováno vložení do databáze, každý záznam je přidáván jako jedinečný. Po dosažení konce zdrojového souboru následuje ukončení jeho čtení voláním funkce *fclose($fp)*.

16 preg_match($reg,$string,$match) - funkce PHP, vrací hledaný řetězec $match shodný s regulárním výrazem $reg v řetězci $string

Import dat

Nahrání nových dat do databáze je přístupné po zvolení odkazu „Nastavení, import“ v horním menu aplikace. K dispozici je jednoduchý formulář (Obrázek 3) s možností zadání jména tabulky (pro pozdější identifikaci, např. mujweb1) a cesty k logovacímu souboru na webovém serveru (např. /log/mujweb1.log). V případě existence tabulky v databázi je uživatel vyzván zda-li si přeje přidat nová data ke starým, nebo stávající data nahradit novými. Není-li tabulka k dispozici, je vytvořena nová s parametry uvedenými v kapitole „Databáze“.

Obrázek 3: Import nových dat

Potvrzením formuláře tlačítkem „Importovat“ dochází ke spojení s databází (realizováno pomocí funkce *connectDB()*, která vychází z uložených dat v souboru *configuration.php*, a vytvoří trvalé spojení s databází), následuje zpracování logovacího souboru dle postupu uvedeného v předcházející kapitole o převodu stávajících dat. Při získání jednotlivých dat z řádku logovacího souboru je pole hodnot *$x[0]* až *$x[5]* vždy vloženo do tabulky zvolené ve formuláři. Příklad vložení:

```
$sql_query = "INSERT INTO " . $table_name . " (id, ip,
accesstime, request, requesttype, code, bytes, reference,
browser) VALUES (NULL,'" . $x[0] . "','" . $x[1] . "','" .
htmlentities($x[2][0]) . "','" . $x[2][1] . "','" . $x[3] . "','"
. $x[4] . "','" . htmlentities($x[5]) . "','" . $x[6] . "')";

$result_query = mysql_query($sql_query);
```

Proměnná *$sql_query* obsahuje příkaz jazyka SQL a funkce *mysql_query()* jej vykoná na spojené databázi. Jak již bylo uvedeno, logovací soubory obsahují i statisíce řádků, proto již samotný proces převodu dat ze souboru do databáze trvá např. pro 500 000 řádků přibližně dvě minuty (záleží na výkonu a vytížení daného serveru). Po úspěšném dokončení převodu je v databázi dostupná nová tabulka „novatabulka“.

Po samotném importu dat do databáze se realizují výpočty jednotlivých

statisických dat. Realizace přímého výpočtu za chodu, kdy by se každá hodnota přepočítávala „online“, je vhodná pouze pro malé logovací soubory o velikosti 10 000 řádků. Pro větší je již proces časově náročný a uživatel je nucen čekat. Z tohoto důvodu jsem zvolil generování těchto dat ihned po importu.

První částí statistických výpočtů je získání hodnot celkového počtu návštěv (záznamů), unikátních návštev, unikátních IP adres, odeslaných stránek, odeslaných dat, HTTP kódů a časového rozmezí dostupnosti dat. Využívá se funkcí *getCount()*, *getVisitCount()*, *getHTTPcode()*, které jsem vytvořil a jsou k dispozici v souboru *functions_sql.php*. Skript pokračuje hledáním TOP údajů v podobě nejpoužívanějších prohlížečů, nejžádanějších požadavků nebo nejčastějších referencí, realizuje se pomocí *getTop()*. Následuje analýza dat z časového hlediska s použitím funkce *getAllDate()*, nejdříve jsou hledány roky, pro každý nalezený rok jsou vyhledány dostupné měsíce a pro ně se již opět volají funkce pro výpočet statistických dat včetně časového omezení na daný měsíc. Pro nalezené měsíce se ještě dodatečně generují statistiky jednotlivých dní. Získané hodnoty jsou ukládány do proměnných a ty následně převedeny do databáze. Obecné statistiky pro kompletní záznamy jsou uloženy dle vyobrazení Tabulky 8, ukládání probíha do tabulky *log_statdata*.

Tabulka 9: Příklad záznamu obecných statistických dat

Sloupec v tabulce	***Hodnota***
id	log_Xnovatabulka
records	335567
visits	10729
ip	4056
bytes	2147483647
pages	25506
http	200,251091;304,74669;302,8375;206,1432;
day_data	1,128,39646839;2,160,50645833;3,178,43859561;

Rozložení v Tabulce 9 udává, jaká data jsou ukládána, u sloupce *http* jsem zvolil sloučení všech dostupných kódů a jejich počet do jedné hodnoty, identifikace kódu a jeho počet je oddělen čárkou, další kód středníkem, při zpětném načítání dat je řetězec rozdělen dle uvedených dělících znaků. Podobně jsou ukládána data o jednotlivých dnech - první číslo udáva den, druhé počet unikátních návštěv a třetí počet odeslaných dat. V případě ukládání informací pro jednotlivé měsíce se do sloupce *id* ukládá název tabulky vč. časové informace, pokud budeme ukládat data pro leden roku 2007, *id* bude mít tvar *log_Xnovatabulka_200701*.

Složení záznamů pro TOP statistiky je uvedeno v Tabulce 10, hlavním údajem je sloupec *tab*, obsahující název tabulky, ke které data patří, dalším důležitým sloupcem je *type* udávající druh záznamu (browser – prohlížeč, request – žádost, reference – odkazovač), údaje jsou ukládány do *tabulky log_statdata_top* v databázi.

Tabulka 10: Příklad záznamu pro TOP ukazatel (prohlížeče)

Sloupec v tabulce	***Hodnota***
id	1
tab	log_Xnovatabulka
type	browser
value	MSIE 6.0
count	87202

Po kompletním provedení importu a generování dat je nová tabulka nastavena jako výchozí, její jméno je uloženo do konfigurační tabulky *log_configuration* a lze s ní ihned pracovat. Celý proces importu a generování statistických dat může zabrat i několik minut, záleží na počtu řádků logovacího souboru.

Nastavení

Aplikace umožňuje uživateli nastavit několik využitelných dat v oblasti vypisování informací, náhled nastavení je k dispozici na Obrázku 4, první nastavitelnou

Obrázek 4: Náhled formuláře pro nastavení zobrazovacích proměnných

položkou je počet řádků při sledování (max_row_ip), tzn. počet IP adres, které se vypíší v seznamu po otevření odkazu „Sledování“ v menu aplikace. Následuje možnost změny počtu detailních záznamů (max_row_record), které jsou vypisovány např. u vybraného záznamu (IP adresy) v sekci „Sledování“. Poslední položkou je počet záznamů u TOP statistik (max_row_top). Údaje jsou načítány a ukládany z/do tabulky *log_configuration*.

Při existenci více záznamů je uživateli zobrazen přehled s dostupnými tabulkami v databázi (viz Obrázek 5) a nabídnuta možnost aktivace a odstranění. Aktivní tabulka je vyznačena zeleným kroužkem za jejím názvem (v případě Obrázku 5 se jedná o záznam *alumi*).

Obrázek 5: Náhled přehledu dostupných tabulek v databázi

2.3.6 Statistiky

Otevřením odkazu „Statistiky“ je inicializována proměnná *$action=statistics*, následuje přiložení skriptu *statistics.php*, pomocí něhož získává uživatel okamžitý přehled o uložených datech v databázi. Data jsou získána několika kroky:

- načtením řádku z tabulky *log_statdata*, kde sloupec *id* odpovídá názvu aktivní tabulky. Získaná data se ukládají do proměnných určených pro šablonovací systém, hodnoty sloupců *http* a *day_data* jsou rozloženy do pole (využití funkce *explode()* pro rozdělení řetězce), u jednotlivých HTTP kódů jsou generovány odkazy pro detailní výpis záznamů s daným kódem (volání skriptu *view.php* pomocí *$action=view*), zobrazení dat uživateli je realizováno pomocí TemplatePower a šablony */templates/statistics/stat_main.tpl*, jíž jsou předána nalezená data.

- ➔ dalším krokem je získání TOP statistických dat, vycházíme z tabulky *log_statdata_top* a hledáme řádky obsahující položku *tab* rovnu názvu aktivní tabulky se záznamy a odlišujeme dle typu záznamu (sloupec *type*). Zobrazení využívá šablony */templates/statistics/stat_top.tpl*.
- ➔ posledním krokem je výpis jednotlivých nalezených měsíců, data jsou uložena v tabulce *log_statdata*, hledáme ve sloupci *id* hodnotu rovnu aktivní tabulce se záznamy a poté zjišťujeme přítomnost časového údaje přidaného do názvu (např. log_Xnovatabulka_200601), pokud je úspěšně nalezen, jsou načtena jednotlivá statistická data a poté zobrazena pomocí šablony */templates/statistics/stat_main.tpl*

Obecna statistika	
Celkovy pocet zaznamu (navstev)	3893
Celkovy pocet unikatnich navstev	542
Celkovy pocet unikatnich zaznamu (IP adres)	371
Celkovy pocet odeslanych stranek (*.htm, *.html, *.php, *.asp)	46
Celkovy pocet odeslanych dat [B]	28489628
Pocet HTTP odpovedi typu 200 (detail)	2604
Pocet HTTP odpovedi typu 404 (detail)	690
Pocet HTTP odpovedi typu 304 (detail)	580
Pocet HTTP odpovedi typu 206 (detail)	19
Zaznamy jsou k dispozici v intervalu	01.12.2006 15.12.2006
TOP uzivatelske prohlizece	
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)	664
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)	312
TOP reference	
http://www.inrex.cz/	1572
http://www.inrex.cz/?akce=products	670
TOP zadosti	
/	236
/pictures/kamion.jpg	164
Statistika pro mesic 12/2006 (detail)	
Celkovy pocet zaznamu (navstev)	3893
Celkovy pocet unikatnich navstev	542
Celkovy pocet unikatnich zaznamu (navstev)	371
Celkovy pocet odeslanych (*.htm, *.html, *.php, *.asp)	28489628
Celkovy pocet odeslanych dat [B]	46

Obrázek 6: Náhled a přehled statistik

Statistiky nabízejí detailní informace, přehled s náhledem na aplikace je k dispozici na Obrázku 6. První část zaujímá obecná statistika, následuje přehled Top

statistik po dvou záznamech a posledním blokem je statistika pro nalezený prosinec roku 2006 s možností detailního výpisu.

2.3.7 Sledování

Funkci monitorování lze spustit odkazem v horním menu „Sledování“, kdy je do proměnné *$action* přiřazena hodnota *view* a tím načten skript *view.php* obsahující následující možnosti, které se volí změnou proměnné *$detail*:

- Základním režimem je zobrazení výpisu všech dostupných IP adres, proměnná *$detail* je nulová. Hledání záznamů je relizováno spojením tabulek *logX_novatabulka* a *log_ip2name* pomocí SQL příkazu. Získáváme IP adresu, její možný překlad na doménové jméno a počet záznamů nalezených pro danou IP. Realizace SQL příkazu je následující:

```
SELECT a.ip, b.ipname,b.ipactive, COUNT(a.id) AS ip_count FROM
logX_novatabulka AS a LEFT JOIN log_ip2name AS b USING (ip)
GROUP BY a.ip ORDER BY b.ipactive DESC"
```

  Získaná data jsou zobrazena pomocí šablony */templates/view/view_ip_address.tpl* . Omezení výpisu je zajištěno konfigurační proměnnou *max_row_ip*, v případě většího počtu záznamů je k dispozici stránkování nebo možnost vyhledávání pomocí formuláře. Příklad vyhledávání je v Tabulce 11.

Hledaná adresa	***Řetězec pro hledání***	***Výsledek***
192.168.x.x	192.168.	**192.168.**1.10, **192.168**.68.1, ...
xx**.87.0.**x	%.87.0.	90.**87.0**.12, 10.**87.0**.1, ...

- Poslední možností je aktivace sledování uživatelů dle IP adresy. Aktivované záznamy jsou vypisovány na prvních pozicích včetně označení. Při aktivaci/deaktivaci probíha úprava tabulky *log_ip2name* následujícím SQL příkazem:

```
REPLACE INTO log_ip2name (log_ip,log_ipname,log_ipactive)
VALUES ('10.1.1.10','test.comp.com',1)
```

➔ Detailní informace o IP adrese jsou získány pomocí nastavení proměnné *$detail=ip_address*, následuje vyhledání všech dostupných dat k zvolené adrese pomocí SQL příkazu:

```
SELECT * FROM log_Xnovatabulka WHERE ip = '10.1.1.10'
```

Získaná data jsou zobrazena šablonou */templates/view/view_ip_detail.tpl*, k dispozici je i filtrování v podobě formuláře pro zadání časového rozmezí, klíčového slova či výběru HTTP kódu. Filtrování je přidáno k SQL příkazu, např. při nutnosti vyhledání záznamů s HTTP kódem 200 vypadá příkaz pro databázi následovně:

```
SELECT * FROM log_Xnovatabulka WHERE ip = '10.1.1.10' AND code
= 200
```

2.3.8 Export

Aplikace umožňuje ukládat data pro další zpracování pomocí odkazu v podobě tiskárny umístěné v patičce. Pro ukládání dat jsem zvolil formát XML, celý export zajišťuje skript *export.php*. Ukládat lze detailní výpisy a obecná statistická data ze sekce „Statistiky“.

Skript pro export využívá přístupu do databáze, kdy vyhledává data dle posledního SQL příkazu, který byl použit pro zobrazení dat v aplikaci. Následuje generování XML souboru a automatická nabídka k uložení. Náhled získaného souboru:

```
<?xml version="1.0" encoding="iso-8859-2"?>
<CONTAINER>
      <RECORD>
            <IP>167.206.235.5</IP>
            <ACCESSTIME>2006-12-04 16:53:44</ACCESSTIME>
            <REQUEST>/index_de.php?akce=contacts</REQUEST>
            <CODE>200</CODE>
            <BYTES>11324</BYTES>
            <REFERENCE>http://www.google.de/</REFERENCE>
            <BROWSER>Mozilla/4.0 (compatible...</BROWSER>
      </RECORD>
</CONTAINER>
```

Závěr

Dokument seznamuje čtenáře se způsoby a možnostmi logování webového serveru. Stručný přehled dostupných systémů určených analýze logovacích dat ukazuje jejich výhody a nevýhody. Za hlavní hledisko při výběru řešení pro monitorování bych zvolil účel získaných dat. Pokud jsou získaná data určena pro vlastní optimalizaci, stačí využít „offline“ systému. Řešení „online“ bych volil v případě nutnosti prezentovat data v rámci nabídky našich stránek pro reklamu, je vhodné převším pro svou nezávislost při měření dat, možnost zkreslení dat v náš prospěch je minimální.

V rámci diplomové práce jsem vytvořil webovou aplikaci pro analýzu logovacích souborů dle zadání. Aplikace je plně funkční a hlavní cíl ve formě sledování jednotlivých uživatelů lze využít například ke sledování jednotlivých robotů celosvětových vyhledávačů (analýzu chování, přizpůsobování webu) díky nimž se naše webové stránky dostávají hlouběji do světa internetu. Další vhodné využití vidím v nasazení u podnikové sítě, kde jsou provozovány interní internetové stránky. Administrátor či vedení společnosti má možnost sledovat využívání internetových stránek zaměstanci (jednotlivými odděleními) a v návaznosti na získaná data provádět optimalizaci struktury či obsahu.

Ve srovnání s existujícími aplikacemi je tento projekt vhodný pro nasazení na menší servery (přibližně 1 milión řádků v logovacím souboru). Toto doporučení vychází ze zkušeností získaných při používání aplikace. Přenos velkého množství dat do databáze MySQL je časově náročnější a ukazuje výhodu programovacího jazyka Perl, který umí zpracovávat data až dvojnásobně rychleji. Výhodou oproti stávajícím řešením je jednoduchost instalování aplikace a přístup k získaným datům. Především případný náhled na data a dostupnost v databázi přináší možnost dalšího vývoje. Uživatel může rozšířit stávající aplikaci nebo vytvořit novou a analyzovat data dle aktuálních potřeb.

Literatura

[1] http://httpd.apache.org/ - dokumentace serveru Apache

[2] http://templatepower.codocad.com/ - dokumentace šablonovacího systému TemplatePower

[3] http://www.php.net/manual/cs/ - dokumentace skriptovacího jazyka PHP

[4] http://www.w3schools.com/ - přehled pravidel pro tvorbu webových stránek

[5] http://www.wikipedia.org/ - všeobecná encyklopedie

[6] Kolektiv autorů: *Linux Dokumentační Projekt 3. vydání*. Computer Press Praha 2003, ISBN 80-7226-761-2

[7] Kosek, J.: *PHP, tvorba interaktivních internetových aplikací, podrobný průvodce*. Grada Publishing Praha 1999, ISBN 80-7169-373-1

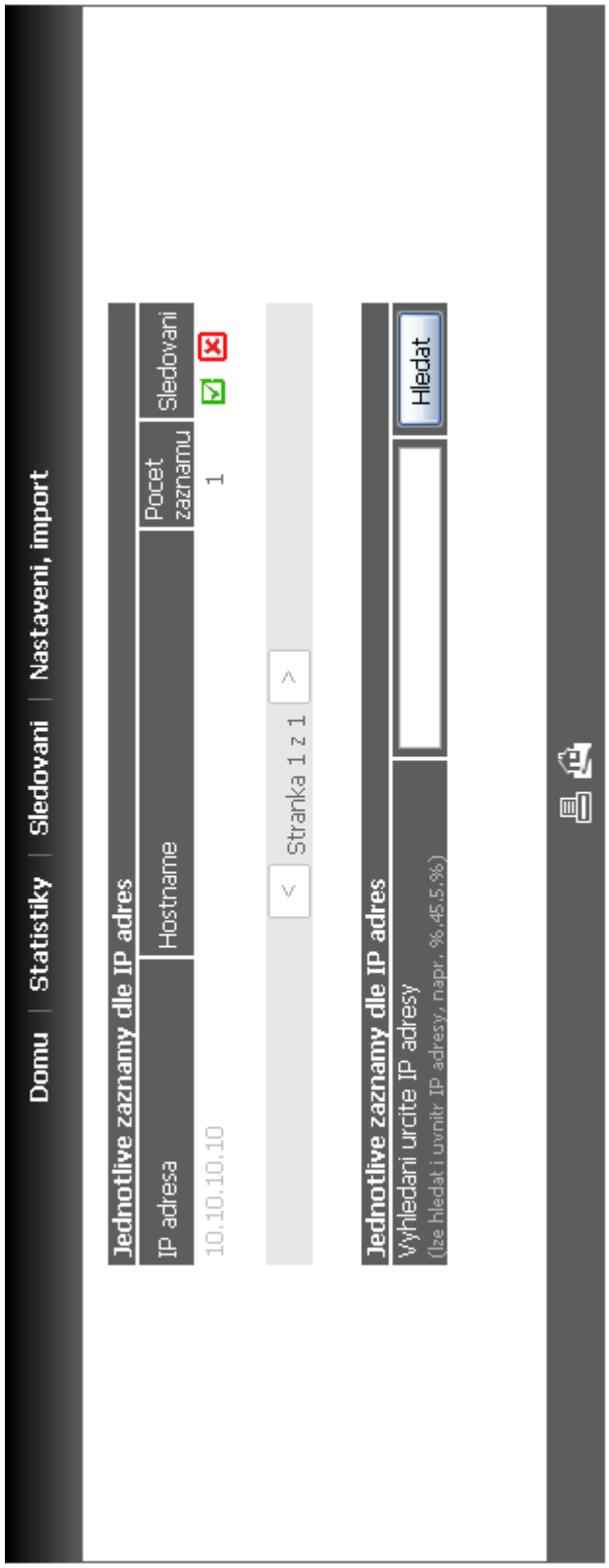
[8] Maslakowski, M.: *Naučte se MySQL za 21 dní*. Computer Press Praha 2001, ISBN 80-7226-448-6

Příloha A – Přehled HTTP kódů

Druh	***Kód***	***Popis***
Informační (100-199)	100 Continue	Klient může pokračovat v zasílání požadavků
	101 Switching Protocols	Server mění komunikační protokol
Úspěch vykonání (200-299)	200 OK	Požadavek byl úspěšně vykonán
	201 Created	Výsledkem požadavku nově vytvořený objekt
	202 Accepted	Přijat asynchronní požadavek, akceptován, odpovídající činnost se nemusela zatím provést
	204 No content	Požadavek byl úspěšně vykonán, žádná data pro klienta
Přesměrování (300-399)	300 Multiple choises	Požadovaná data jsou dostupná z více míst, možnosti jsou vráceny klientovi
	301 Moved Permanently	Požadovaná data jsou trvale přesunuta na jinou adresu, další odkazy musí použít tuto novou adresu
	302 Moved Temporarily	Požadovaná data jsou dočasně přesunuta na jinou adresu, další odkazy mohou použít původní adresu
	304 Not Modified	Podmíněný požadavek úspěšně zpracován, avšak od udané doby nebyl modifikován
Chyby klienta (400-499)	400 Bad Request	Chybný požadavek, server nerozumí, klient jej musí zaslat znovu
	401 Unauthorized	Pokud je požadavek anonymní, musí se autorizovat, v opačném případě přístup odepřen
	403 Forbidden	Autorizace nebyla úspěšná
	404 Not Found	Žádaná adresa nenalezena
	406 Not Acceptable	Požadovaná data nejsou k dispozici ve formátu, který je požadován klientem
	408 Request Timeout	Klient nestihl odeslat požadavky ve stanoveném časovém intervalu
Chyby serveru (500-599)	500 Internal Server Error	Vnitřní chyba serveru
	501 Not Implemented	Požadavek není podporován
	503 Service Unavailable	Server nemůže dočasně obsloužit požadavek

Příloha B – Náhled aplikace - statistiky

Příloha C – Náhled aplikace - sledování



Příloha D – Náhled aplikace – sledování, detail