

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleš Čadež

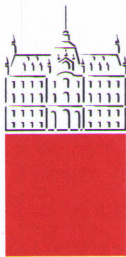
**SISTEM ZA ANALIZO OBISKOV SPLETNEGA
MESTA**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

LJUBLJANA, 2009

Št. naloge: 01543/2009

Datum: 15.02.2009



Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ALEŠ ČADEŽ**

Naslov: **SISTEM ZA ANALIZO OBISKOV SPLETNEGA MESTA
SYSTEM FOR ANALYSIS OF WEBSITE TRAFFIC**

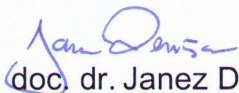
Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Pripravite sistem za analizo obiskov spletnega mesta. Temelji naj na analizi dnevniških datotek, kakršne je mogoče pripravljati s spletnim strežnikom Apache. Pri zasnovi sistema se posvetite predvsem naprednim možnostim analize, torej različnim možnim filtrom in kombinacijam le-teh, različnim možnim načinom združevanja v skupine in podobno. Ob tem naj bo uporaba sistema vseeno kar se da enostavna in primerna tudi za manj izkušene uporabnike. Sistem naj ponuja čimveč grafičnih predstavitev zbranih podatkov.

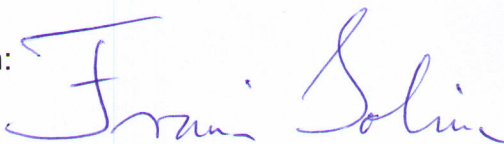
Sistem naj temelji na odprtokodnih rešitvah in naj ga bo po možnosti preprosto namestiti na različnih operacijskih sistemih.

Mentor:


doc. dr. Janez Demšar



Dekan:


prof. dr. Franc Solina

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleš Čadež

**SISTEM ZA ANALIZO OBISKOV SPLETNEGA
MESTA**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

Mentor:

doc. dr. Janez Demšar

LJUBLJANA, 2009

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Aleš Čadež,

z vpisno številko 63020023,

sem avtor/-ica diplomskega dela z naslovom:

Sistem za analizo obiskov spletnega strežnika

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

doc. dr. Janez Demšar

in somentorstvom (naziv, ime in priimek)

/ _____

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 23.3.2009

Podpis avtorja/-ice: _____

ZAHVALA

Zahvaljujem se mentorju doc. dr. Janezu Demšarju za strokovno svetovanje, potrpežljivost in dobro voljo pri nastajanju diplomskega dela.

Hvala družini in Evi, za vsestransko podporo v času študija in pisanja diplomskega dela.

Hvala pa tudi vsem nenavedenim, ki so mi kakorkoli pomagali.

KAZALO

KAZALO SLIK.....	IX
KAZALO TABEL	X
SEZNAM UPORABLJENIH KRATIC.....	XI
POVZETEK	1
ABSTRACT	2
1 UVOD.....	3
1.1 NAMEN	3
1.2 ZGRADBA	3
2 OPREDELITEV SPLETNE ANALIZE	5
2.1 POJEM SPLETNE ANALIZE	5
2.2 TEHNOLOŠKE ZNAČILNOSTI SPLETNE ANALIZE	6
2.2.1 Zbiranje podatkov.....	6
2.2.2 Oblike namestitve aplikacije.....	13
2.3 RAZVOJ SPLETNE ANALIZE	14
2.4 PRIHODNOST SPLETNE ANALIZE.....	15
3 PRIMERJAVA Z ŽE OBSTOJEČIMI APLIKACIJAMI	17
3.1 BREZPLAČNE APLIKACIJE ZA ANALIZO DOSTOPOV DO STREŽNIKA.....	18
3.1.1 Analog	18
3.1.2 AWStats.....	19
3.1.3 OneStatFree.....	19
3.1.4 CrazyEgg.....	20
3.1.5 Google Analytics.....	20
3.2 PLAČLJIVE APLIKACIJE ZA ANALIZO DOSTOPOV DO STREŽNIKA	21
3.2.1 WebTrends – Web Analytics 8.....	22
3.2.2 Omniture - SiteCatalyst	23
3.2.3 Coremetrics – Online Analytics.....	24
4 TEHNOLOŠKA PLATFORMA	25
4.1 RELACIJSKA BAZA: MYSQL	25
4.2 SPLETNI STREŽNIK: APACHE HTTP SERVER.....	26
4.2.1 Cronolog	26

4.3	PROGRAMSKI JEZIK SLOJA POSLOVNE LOGIKE: PYTHON	26
4.3.1	<i>Knjižnica MySQLdb</i>	27
4.3.2	<i>Knjižnica Matplotlib</i>	27
4.3.3	<i>Knjižnica PIL – Python Imaging Library</i>	27
4.3.4	<i>Knjižnica Basemap toolkit</i>	27
4.3.5	<i>Knjižnica Numpy</i>	28
4.3.6	<i>Knjižnica ReportLab toolkit</i>	28
4.4	POVEZAVA PYTHON – APACHE: MODUL MOD_PYTHON	28
4.5	PROGRAMSKI JEZIKI SLOJA UPORABNIŠKEGA VMESNIKA.....	29
4.5.1	HTML	29
4.5.2	Javascript.....	29
4.5.3	AJAX.....	30
4.5.4	CSS.....	32
5	ARHITEKTURA SISTEMA	33
5.1	PODATKOVNI SLOJ	34
5.1.1	<i>Dnevniška datoteka</i>	34
5.1.2	<i>Shranjevanje podatkov</i>	35
5.1.3	<i>Tabele podatkov relacijske baze</i>	35
5.2	SLOJ POSLOVNE LOGIKE	38
5.2.1	<i>WebLAT_DB.py</i>	38
5.2.2	<i>WebLAT_Util.py</i>	39
5.2.3	<i>WebLAT.py</i>	41
5.2.4	<i>WebLAT_Scheduler.py</i>	45
5.3	SLOJ UPORABNIŠKEGA VMESNIKA.....	45
5.3.1	<i>WebLAT.html</i>	45
5.3.2	<i>WebLAT_js_funkcije.js</i>	48
5.3.3	<i>WebLAT_css.css</i>	54
5.3.4	<i>WebLAT_help.html</i>	54
5.3.5	<i>Calendar.css</i>	54
5.3.6	<i>Calendar.js</i>	54
6	NAMESTITEV SISTEMA	55
6.1	POTREBNA PROGRAMSKA OPREMA.....	55
	• <i>Relacijska podatkovna baza MySQL</i>	55
	• <i>Strežnik Apache</i>	55
	• <i>Cronolog</i>	55
	• <i>Python</i>	55

6.2	KONFIGURACIJA STREŽNIKA APACHE	57
	<i>Definicija dnevniške datoteke</i>	57
	<i>Določitev »handlerja« strežnika</i>	57
	<i>Konfiguracija »mod_python« modula</i>	57
6.3	POSTAVITEV FIZIČNE PODATKOVNE BAZE.....	57
6.4	NAMESTITEV SKRIPT IN DATOTEK.....	58
	<i>Konfiguracija datoteke »filters«</i>	58
	<i>Konfiguracija dostopa do podatkovne zbirke</i>	59
	<i>Nastavitev dnevnega vpisa – »task«</i>	59
7	DELOVANJE APLIKACIJE	60
7.1	ORODNA VRSTICA	60
7.2	NAČIN »SIMPLE«	61
7.3	NAČIN »ADVANCED«	62
7.4	NAČIN »COMPLEX«	64
7.5	IZVOZ PODATKOV.....	64
7.6	POMOČ, KONTAKT IN TISKANJE	64
8	ZAKLJUČEK	65
9	LITERATURA	67
	PRILOGE	69
9.1	VSEBINA PRILOŽENE ZGOŠČENKE	69

KAZALO SLIK

Slika 1: Prikaz zbiranja podatkov v načinu spletnih dnevnikov	7
Slika 2: Prikaz zbiranja podatkov v načinu spletnih signalov	9
Slika 3: Prikaz zbiranja podatkov s pomočjo Javascript označb.....	10
Slika 4: Zbiranje podatkov s pomočjo paketnega vohljanja.....	12
Slika 5: Analog v0.9.....	14
Slika 6: Primer gostote klikanja	15
Slika 7: Primer uporabe toplotne karte.....	16
Slika 8: Tržni delež aplikacij za spletno analizo (januar 2008) [16].....	18
Slika 9: Prikaz delovanja aplikacije Analog	18
Slika 10: Prikaz delovanja aplikacije AWStats.....	19
Slika 11: Prikaz povezave do OneStat aplikacije.....	19
Slika 12: Google Analytics nadzorna plošča.....	21
Slika 13: WebTrends - Web Analytics 8 nadzorna plošča.....	22
Slika 14: Omniture SiteCatalyst nadzorna plošča	23
Slika 15: Coremetrics nadzorna plošča	24
Slika 16: Primerjava spletne komunikacije	32
Slika 17: Trinivojska arhitektura.....	33
Slika 18: Entitetni model.....	36
Slika 19: Struktura WebLAT.html	45
Slika 20: Sekcija »Top«	46
Slika 21: Sekcija »Simple«	46
Slika 22: Primer prikaza sekcije "SimpleResults"	47
Slika 23: Skica "Advanced" sekcije	47
Slika 24: Prikaz sekcije "Complex"	48
Slika 25: Prikaz delovanja metode showTBody	51
Slika 26: Orodna vrstica	60
Slika 27: Način "Simple"	61
Slika 28: Način "Advanced".....	62
Slika 29: Prikaz filtriranja podatkovne skupine	63
Slika 30: Način "Complex".....	64

KAZALO TABEL

Tabela 1: Prednosti treh oblik aplikacij [4 – str 74]	14
Tabela 2: Najbolj pogoste aplikacije za spletno analizo	17
Tabela 3: Razčlenitev definicije dnevniške datoteke	34
Tabela 4: Seznam vseh kod statusa	37
Tabela 5: Parametri funkcije Call_Advanced	49
Tabela 6: Seznam in opis datotek in map	58

SEZNAM UPORABLJENIH KRATIC

- AJAX** »Asynchronous JavaScript and XML« ali Asinhroni JavaScript in Xml je skupina medsebojno povezanih spletnih razvojnih tehnik, uporabljenih za ustvarjanje interaktivnih spletnih aplikacij. Z AJAX-om si lahko spletne aplikacije izmenjujejo podatke s strežnikom asinhrono v ozadju, brez potrebe po ponovnem nalaganju strani.
- CDI** »Costumer-driven innovation« so uporabniško vodene inovacije, t.j. inovacije, ki pridejo preko zahtev uporabnika.
- CSS** »Cascading Style Sheets« so predloge, ki določajo izgled spletnih strani. Z njimi določamo pisavo, velikosti črk ter vizualno predstavitev spletne strani. HTML naj bi predstavljal semantično strukturo in smiselno hierarhijo dokumenta, CSS pa predstavitevno vlogo.
- HTML** »Hypet Text Markup Language« je označevalni jezik za izdelavo spletnih aplikacij, ki predstavlja osnovo vsakega spletnega dokumenta. Z njegovo pomočjo ustvarimo strukturo in semantično ureditev dokumenta. Pišemo ga lahko v vsakem urejevalniku besedil in je dokaj preprost za učenje in uporabo.
- IP** »Internet Protocol« naslov je številka, ki natančno določa računalnik v omrežju Internet. Število je 32-bitno, za bolj pregleden prikaz pa je običajno zapisano s štirimi osembitnimi vrednostmi v desetiški obliki, npr. 193.95.198.35. Del teh bitov je možno uporabiti za ustvarjanje podomrežij znotraj nekega omrežja.
- KPI** »Key performance indicators« so indikatorji (matrike, grafikoni), najpomembnejši za poslovanje ali druga opazovanja uspešnosti. Kratica se uporablja tako v poslovnem kot računalniškem žargonu.
- PDI** »Possibility-driven innovation« so inovacije na podlagi tehnično-podatkovnih zmožnosti.
- SQL** »Structured Query Language« ali strukturirani povpraševalni jezik za delo s podatkovnimi bazami je najbolj razširjen in standardiziran povpraševalni jezik za delo s podatkovnimi zbirkami ter programskimi stavki, ki posnemajo ukaze v naravnem jeziku. Določen je z ANSI/ISO SQL standardom. Standard SQL se je razvijal od leta 1986 in danes obstaja več različic.
- URL** »Uniform Resource Locators« ali enolični krajevnik vira je naslov spletnih strani v svetovnem spletu.

POVZETEK

Analiza obiskov spletnega mesta, oz. na kratko spletna analiza, je zelo mlada veja računalništva, ki pa v sodobnem svetu postaja vedno pomembnejša. Vse več oglaševanja in poslovanja podjetij namreč poteka prek spleta, zato je spletna analiza eden ključnih dejavnikov, ki prispevajo h konkurenčnosti podjetja na trgu, saj ponuja sredstva optimizacije in pritegnitve uporabnika glede na njegove želje in potrebe, hkrati pa podjetjem omogoča spremljanje dogajanja na straneh v daljšem časovnem obdobju.

Analiza lahko poteka tako na tehnični kot tudi na poslovni ravni. Tehnična analiza ponuja merjenje in predstavitev podatkov o dogajanju na strežniku, kot so npr. količine prenešenih podatkov, število obiskov, lokacije s katerih dostopajo uporabniki, odstotek uspešnih/neuspešnih zahtev, itd. Po drugi strani pa se poslovna analiza ukvarja predvsem z obnašanjem uporabnika na naših straneh - kam je uporabnik klikal, katere podstrani so ga vzpodbudile k nakupu izdelkov, s katerih strani je zaključil obiskovanje našega portala, itd. Glavni koncepti in pojmi spletne analize so temeljito predstavljeni v prvih dveh poglavjih diplomskega dela.

Izdelan odprtokodni sistem ponuja analiziranje na tehnični ravni. Zajema podatke in jih prek spletne aplikacije prikazuje administratorju, kar omogoča tehnično analizo dogajanja na spletnem strežniku. Administratorju preko prijaznega uporabniškega vmesnika omogoča pregledovanja statističnih podatkov v obliki grafikonov in matrik.

Z namenom ponujati sistem v uporabo ali nadaljni razvoj, so vse uporabljene tehnologije, kot tudi aplikacija, odprtokodne ter prenosljive.

Ključne besede:

Analiza, Splet, Spletna analiza, AJAX, Python, Odprta koda

ABSTRACT

Analysis of web site traffic, or web analysis for short is a very young branch in computer science, which is becoming increasingly important in the modern world. The share of sales and advertising via the Internet is increasingly growing and web analysis offers ways to optimize the site to be more user friendly and consequently keep user on site longer.

The analysis can be done on technical and business level. Technical analysis provides a measurement and presentation of events on the server, such as the bandwidth, number of hits, referer, the percentage of successful / unsuccessful requests etc. On the other hand, the business analysis deals with the behavior of the user on site. Where a user clicks, the pages from which he purchases products, pages from which he leaves our site, etc. The main ideas and concepts of web analysis are presented in the first two chapters of this diploma thesis.

The main emphasis of the diploma thesis was on the development of the system, which collects data from the server and presents them via a web application for chart and matrix based analysis.

In order to offer the system for use and further development, all of the technologies, as well as the application are open source based.

Key words:

Analysis, Web analysis, Website traffic, AJAX, Python, Open source

1 UVOD

1.1 Namen

Namen diplomske naloge je izdelati sistem, ki omogoča pregledovanja dostopov do spletnega strežnika in izdelati uporabniku prijazno aplikacijo, ki na različne načine omogoča pregledovanje statistike.

Gre za sistem, ki zajema celoten proces, od zajema pa do prikaza podatkov uporabniku. V ta namen je ustvarjenih več oblik grafikonov in matrik, na katerih je mogoče poljubno izbirati grupiranje in filtriranje podatkov (datum, država, kontinent, protokol, IP, brskalnik, operacijski sistem, itd.), kar omogoča podrobno analizo dostopov na strežnik.

Sistem je bil v skladu z nameni nameščen na strežnik Laboratorija za umetno inteligenco (LUI) Fakultete za računalništvo in informatiko Univerze v Ljubljani.

Poleg namestitve sistema na strežnik LUI, je bil eden izmed namenov tudi možnost prenosljivosti sistema na druge strežnike in tudi ta vidik je realiziran.

1.2 Zgradba

Diplomsko delo je vsebinsko razdeljeno na dva dela.

Prvi del je namenjen predstavitvi sistemov spletne analize nasploh in z ozirom na sistem, razvit v tej diplomski nalogi. Prvi del tako sestavlja poglavje »OPREDELITEV SPLETNE ANALIZE«, kjer je podana definicija in ključni pojmi spletne analize, njene zmožnosti, cilji, zgodovina in nadaljni razvoj ter poglavje »PRIMERJAVA Z ŽE OBSTOJEČIMI APLIKACIJAMI«, v kateri aplikacijo lastne diplomske naloge primerjam z drugimi obstoječimi aplikacijami različnih brezplačnih ali plačljivih ponudnikov.

Drugi del diplomskega dela natančneje opisuje sestavo in delovanje izdelanega sistema. Sestavljen je iz štirih poglavij. Poglavje »TEHNOLOŠKA PLATFORMA« vsebuje opis vseh uporabljenih tehnologij, programov in knjižnic ter načinov njihovega povezovanja in

vključevanja v delovanje sistema. Poglavje »ARHITEKTURA SISTEMA« opisuje večnivojsko zasnovo sistema. Razdeljeno je na tri podpoglavja (Podatkovni sloj, Sloj poslovne logike in Sloj uporabniškega vmesnika), ki predstavljajo nivoje arhitekturne zasnove. V vsakem podpoglavju so opisani vsi modeli ter znotraj njih metode in funkcije spletne aplikacije. Poglavje »NAMESTITEV SISTEMA« opisuje vse korake in sistemske zahteve za namestitev sistema na druge strežnike. Opis delovanja aplikacije v obliki uporabniškega priročnika pa najdemo v poglavju »DELOVANJE APLIKACIJE«.

V zaključku (»ZAKLJUČEK«) so predstavljeni temeljni izzivi, s katerimi se soočajo razvijalci tovrstnih sistemov in s katerimi sem se soočil tudi sam, poleg tega pa so predstavljene tudi možnosti nadaljnega razvoja in optimizacije sistema glede na potrebe posameznega uporabnika.

2 OPREDELITEV SPLETNE ANALIZE

2.1 Pojem spletne analize

Čeprav je spletna analiza obstajala že v devetdesetih letih prejšnjega stoletja, je bila prva uradna definicija sprejeta šele v letu 2006.

»Analiza spleta je merjenje, zbiranje, analiziranje in predstavljanje internetnih podatkov z namenom razumevanja in optimiziranja uporabe spleta«¹[9]

Iz definicije sledi, da je spletna analiza sestavljena iz štirih bistvenih delov. Prvi je zbiranje podatkov, ki lahko poteka – kot bo razloženo v nadaljevanju – na različne načine. Drugi del predstavlja merjenje in analiziranje podatkov, tretji pa zajema predstavitev rezultatov v obliki matrik in grafikonov različnih oblik, ki uporabniku pomagajo pri razumevanju obnašanja in optimizaciji (četrti del) spletne strani.

V poslovnem svetu lahko optimizacija spletne strani pomeni tudi v večji dobiček, ki izhaja iz naslova večje prodaje ali reklamiranja prek spletnih strani. Ker je v zadnjih letih prodaja prek spletnih strani v velikem porastu, vedno več podjetij vlaga ne samo v razvoj spletnih strani, ki omogočajo prodajo izdelkov, pač pa tudi v samo analizo obnašanja uporabnikov na strani. Kako obdržati uporabnike na strani, ter kako uporabnika pripeljati čez proces spletnega nakupa s čim večjo verjetnostjo. Ni namreč dovolj zgolj štetje uporabnikov na strani, pač pa predvsem odstotek uporabnikov, ki se odloči za nakup.

Sodobna spletna analiza je torej veliko več, kot le pregledovanje števila gostov na strani ali časa obiska strani, kot je bila pogosta praksa v preteklosti.

¹ Ang. - »Web Analytics is the measurement, collection, analysis and reporting of Internet data for the purposes of understanding and optimizing Web usage«.

2.2 Tehnološke značilnosti spletne analize

2.2.1 Zbiranje podatkov

Zbiranje podatkov je mogoče na več načinov, od katerih ima vsak svoje prednosti in slabosti. Skupen jim je princip zbiranja podatkov na podlagi klikanja uporabnika po spletni strani (angl. clickstream). Izbira načina zbiranja podatkov je prvi korak pri načrtovanju aplikacije za spletno analiziranje, zato mora temeljiti na poznavanju vseh možnih načinov zbiranja podatkov in na željenem delovanju aplikacije [1 – str. 25-33].

Večina aplikacij za spletno analizo uporablja enega od spodaj opisanih načinov.

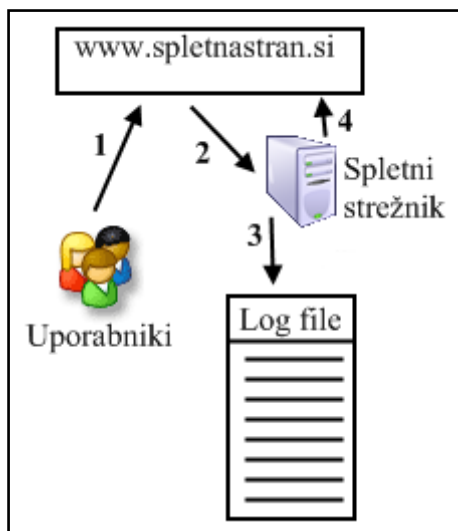
2.2.1.1 Spletni dnevniki

Spletni dnevniki² (angl. web log files) so bili prvi razvit način zbiranja podatkov, s katerim se dogajanje na strežniku beleži v datoteki. Razviti so bili predvsem za potrebe beleženja napak na strežnikih, sčasoma pa so začeli beležiti tudi vedno več podatkov o zahtevah na strežnikih. Danes zbirajo ogromne količine informacij praktično o vsem, kar se na strežniku dogaja. Zbiranje podatkov na ta način se danes najpogosteje uporablja za analiziranje dostopov spletnih iskalnikov³ do naših strežnikov in za analiziranje tehničnih značilnost delovanja spletnega strežnika. Za analiziranje poslovno-marketinško pomembnih informacij o dogajanju na spletnih straneh, pa se pogosteje uporabljajo druge metode.

Podatke o dogajanju na spletnem strežniku je mogoče zajemati tudi v kombinaciji z drugimi načini, toda ker to lahko vodi do kompleksnih problemov pri tolmačenju podatkov, se takšna rešitev uporablja le redko.

² Spletni dnevnik (angl. log file) je tip datotek, kamor se beležijo vsi dogodki. (npr. med instalacijo programa ali med izvajanjem na določenem strežniku).

³ Spletni iskalniki (npr. Google, Yahoo, MSN Search, itd...) brskajo po svetovnem spletu s tako imenovanimi iskalnimi roboti (angl. search bots), ki neprestano posodablajo njihovo bazo. Več kot teh dostopov prihaja na naš strežnik, bolj so naše strani posodobljene med iskalniki in večja je možnost, da bodo naše strani najdene med uporabniki, ki iščejo ključne besede po spletnih iskalnikih.



Slika 1: Prikaz zbiranja podatkov v načinu spletnih dnevnikov

Proces zbiranja podatkov (Slika 1):

- 1) Uporabnik vpiše spletni naslov (URL⁴) v svoj brskalnik.
- 2) Na strežnik prispe zahteva za dostop do strani. Zahtev je lahko več in se lahko beležijo v samostojni ali skupni spletni dnevnik.
- 3) Strežnik prejme zahtevo in zapiše podatke v spletni dnevnik. Oblika spletnega dnevnika je definirana na strežniku. Običajni podatki, ki se zapisujejo, so čas dostopa, verzija brskalnika, verzija operacijskega sistema, IP⁵ in ostali manj pomembni podatki.
- 4) Spletni strežnik pošlje spletno stran brskalniku uporabnika. V primeru, da na strežniku pride do napake, zaradi katere zahtevane strani ne more poslati, se tak dogodek prav tako zabeleži v spletni dnevnik.

Glavne prednosti metode:

- Takšen način zbiranja podatkov je med vsemi metodami najbolj enostaven.
- To je edini način za zbiranje podatkov o dostopih spletnih iskalnikov na strežniku.
- Zbrani podatki so vedno v naši lasti, medtem ko pri ostalih metodah podatke zbirajo ponudniki storitev analiziranja podatkov.

⁴ URL ali enolični krajevnik vira (angl. »Uniform Resource Locators«) je naslov spletnih strani v svetovnem spletu.

⁵ IP-naslov je številka, ki natančno določa računalnik v omrežju Internet. Kratica IP označuje Internet Protocol. Število je 32-bitno, za bolj pregleden prikaz je običajno zapisano s štirimi osembitnimi vrednostmi v desetiški obliki npr. 193.95.198.35. Del teh bitov je možno uporabiti za ustvarjanje podomrežij znotraj nekega omrežja.

Glavne slabosti metode:

- Metoda je primerna le za zbiranje tehničnih podatkov o dogajanju na strežniku. S temi podatki je skoraj nemogoče predstaviti, kaj uporabnik počne na spletni strani. Za te namene so veliko bolj primerne ostale metode.
- Ker se v spletne dnevnike beležijo tudi avtomatizirani dostopi spletnih iskalnikov, zahteve za CSS⁶ datoteke, slikovne datoteke in napake na strani, je potrebno zapise pazljivo filtrirati, saj lahko sicer prihaja do nepravilnosti pri predstavitvi natančnih podatkov.

Ta način zbiranja podatkov uporablja tudi sistem WebLAT, saj je cilj sistem analiziranje tehničnih podatkov o dogajanju na strežniku. Poleg tega je to edini način, pri katerem lahko zbrane podatke hranimo kar na lastnem strežniku. V nasprotju s preostalimi načini ni potrebna sprememba same HTML⁷ kode spletnih strani, ki tečejo na strežniku.

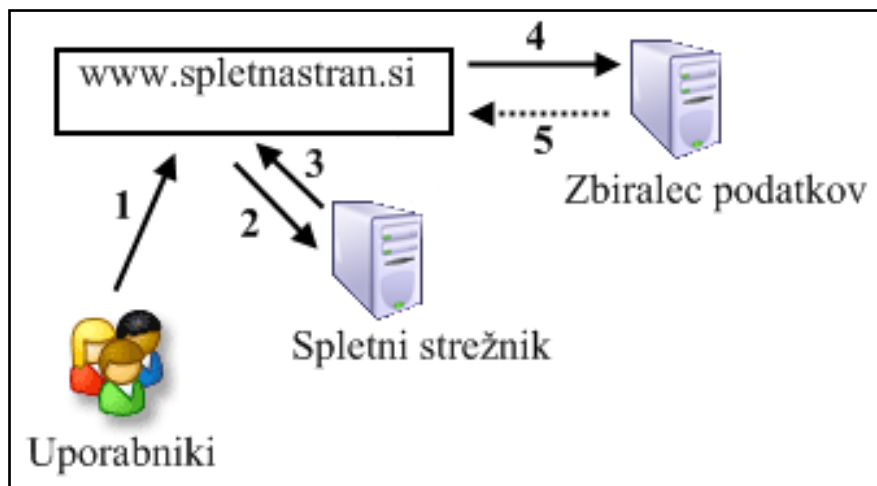
2.2.1.2 Spletni signali

Način zbiranja podatkov s pomočjo spletnih signalov (angl. web beacons) se je uveljavil v času, ko so se na spletu razvila reklamna sporočila v obliki naslovnice (angl. banners). Podjetje je lahko oglaševalo na več različnih spletnih straneh in tako na svojem strežniku (»zbiralec podatkov«) zbiralo podatke o uporabnikih, ki so si sporočila ogledali in na njih kliknili. Tak način zbiranja podatkov je mogoče implementirati tudi pri pošiljanju reklamnih sporočil preko spletne pošte. Strežnik (»zbiralec podatkov«) prejme podatke, ko se spletna pošta naloži v brskalniku spletne pošte uporabnika.

Z uvedbo zbiranja podatkov z »označbami Javascript«, se temu načinu znižuje priljubljenost..

⁶ CSS (angl. Cascading Style Sheets) so predloge, ki določajo izgled spletnih strani. Z njimi določamo pisavo, velikosti črk ter vizualno predstavitev spletne strani. HTML naj bi predstavljal semantično strukturo in smiselno hierarhijo dokumenta, CSS pa predstavitevno vlogo.

⁷ Hyper Text Markup Language (slovensko jezik za označevanje nadbisedila, kratica HTML) je označevalni jezik za izdelavo spletnih strani. Predstavlja osnovo spletnega dokumenta. S pomočjo HTML-ja ustvarimo strukturo in semantično ureditev dokumenta.



Slika 2: Prikaz zbiranja podatkov v načinu spletnih signalov

Proces zbiranja podatkov (Slika 2):

- 1) Uporabnik vpiše spletni naslov (URL) v svoj brskalnik.
- 2) Na strežnik prispe zahteva za dostop do strani.
- 3) Spletni strežnik vrne zahtevano stran skupaj z »GET« zahtevo za 1x1 točko veliko sličico⁸ strežniku (»zbiralec podatkov«). Ta zahteva lahko vsebuje tudi piškotke, ki jih je brskalnik dobil kdaj prej (točka 5) na tej ali kateri drugi spletni strani, ki ji sledi isti strežnik (»zbiralec podatkov«).
- 4) Ko se stran nalaga, pošlje zahtevo (»GET« zahteva iz točke 3) za 1x1 sličico, ter tako pošlje podatek o strani strežniku (»zbiralcu podatkov«).
- 5) Strežnik (»Zbiralec podatkov«) pošlje sličico brskalniku.

Omeniti velja, da je opisan način zbiranja podatkov zelo preprost, saj zbira le podatke o številu dostopov do določene strani. Aplikacije, ki uporabljajo ta način zbiranja podatkov, spletne signale kombinirajo z označbami Javascript in tako lahko dobijo večje število relevantnih podatkov, kot npr. IP naslov, podatke o piškotkih uporabnika, čas, porabljen na strani, itd.

Glavne prednosti metode:

- Spletne signale je lahko implementirati, saj zahtevajo le nekaj vrstic kode.

⁸ 1x1 točko velika sličica, ki je vedno tudi prozorna, je na spletni v sklopu HTML označbe *img src*. Običajno prihaja iz drugega strežnika (angl. third-party server).

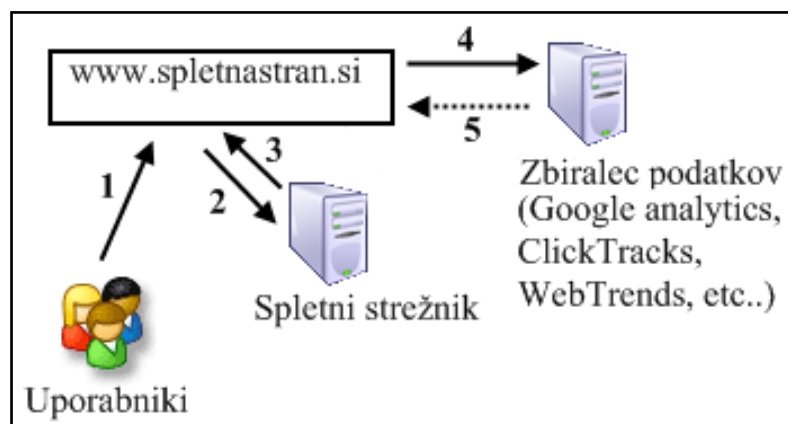
- Enostavno se določa, katere podatke bomo zbirali, poleg tega pa tudi nimamo skrbi s spletnimi roboti, saj le ti ne nalagajo slik.
- Metoda je zelo učinkovita, kadar hočemo analizirati veliko število spletnih strani.

Glavne slabosti metode:

- Če so v programih za prebiranje spletne pošte (Microsoft Office Outlook, Google Gmail) zahtevki za prikaz slik avtomatsko zavrjeni (kar je vedno bolj pogosta nastavitve), ne moremo zbirati podatkov. Enak problem so lahko tudi onemogočeni piškotki v spletnem brskalniku.
- Zajema manjše število podatkov kot označbe Javascript (naslednje poglavje).

2.2.1.3 Označbe JavaScript

Trenutno najbolj pogosto uporabljena metoda zajema podatkov je metoda s pomočjo označb Javascript⁹ (angl. javascript tags). Pri tej metodi so lahko spletne strani na kateremkoli strežniku in ni potrebno, da ga imamo v lasti, ali nam vrača kakršnekoli podatke. Zbrani podatki se zbirajo pri ponudnikih storitev spletne analize. Te nam nato posredujejo podatke v obliki grafikonov, matrik ali obrazcev kar preko spleta.



Slika 3: Prikaz zbiranja podatkov s pomočjo Javascript označb

Proces zbiranja podatkov (Slika 3):

- 1) Uporabnik vpiše spletni naslov (URL) v svoj brskalnik.
- 2) Na strežnik prispe zahteva za dostop do strani.

⁹ »Javascript« – programski jezik, uporabljen za izvajanje programske kode na uporabniški strani spletne strani. »Tag« je označba določenega dela spletne strani. Na podlagi »tag-a« lahko ugotovimo, kam je uporabnik kliknil.

- 3) Strežnik vrne spletno stran skupaj s kodo Javascript.
- 4) Ko se stran nalaga, izvrši kodo v Javascriptu, ki je pripeta strani in pošlje zajemane podatke zbiralcu podatkov oz. ponudniku storitev analize.
- 5) V nekaterih primerih lahko zbiralec podatkov zahteva še določene dodatne podatke o uporabniku.

Glavne prednosti metode:

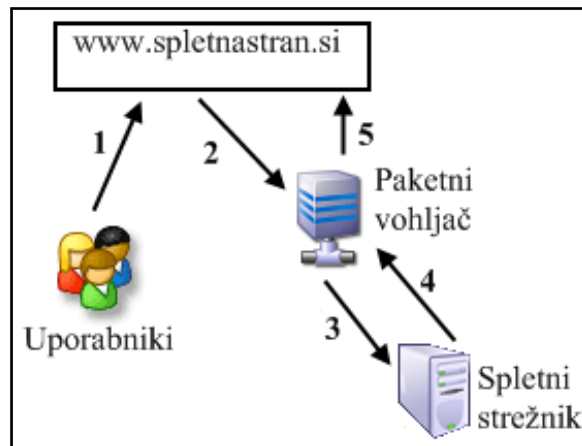
- Poleg spletnih dnevnikov je to metodo najlažje implementirati, saj je vsaki strani potrebno dodati le nekaj vrstic kode Javascript.
- Če administrator oz. razvijalec nima dostopa do spletnega strežnika, je to tudi edini možni način za izvajanje spletne analize. Ravno zato je ta način analize najbolj pogost pri malih in srednjih podjetjih.
- Na različnih spletnih straneh (spletna prodaja) je možno spremljati tudi obnašanje kupca (kdaj je kaj dodal v košarico, kdaj jo je izpraznil, na katerih delih strani je največ klikal, itd.).

Glavne slabosti metode:

- Nekateri uporabniki imajo zaradi varovanja zasebnosti na spletu možnost uporabe Javascript-a izkopljeno. V takem primeru prometa teh uporabnikov ni mogoče spremljati (2-6%). [1 - str. 33].
- Zelo težko je izvajati zbiranje podatkov o datotekah formata PDF, EXE, ZIP, itd.
- Če je na spletni strani preveč kode Javascript, lahko prihaja do konfliktov.

2.2.1.4 Paketno vohljanje

Paketno vohljanje (angl. packet sniffing) je tehnično najbolj dovršen način zbiranja podatkov, ki pa se kljub temu ne uporablja pogosto.



Slika 4: Zbiranje podatkov s pomočjo paketnega vohljanja

Proces zbiranja podatkov (Slika 4):

- 1) Uporabnik vpiše spletni naslov (URL) v svoj brskalnik.
- 2) Zahteva je poslana spletnemu strežniku, a jo na poti ujame strojna ali programska oprema (paketni vohljač). Le-ta shrani podatke, ki jih zbira.
- 3) Paketni voljač pošlje zahtevo naprej spletnemu strežniku.
- 4) Strežnik zahtevo obdela in pošlje stran uporabnikovemu brskalniku, vendar jo na poti ponovno prestreže paketni vohljač in shrani podatke, ki jih zbira.
- 5) Paketni vohljač pošlje zahtevo naprej do uporabnikovega brskalnika.

Glavne prednosti metode:

- Ker gredo vsi podatki preko paketnega vohljača, ni potrebno pisanja kode Javascript.
- Pri postavitvi ni potrebno spreminjati nastavitvev strežnika ali spletnih strani.
- Zbirajo se lahko velike količine informacij, vključno s tehničnimi podatki (podobno kot spletni dnevniki), ki jih z drugimi metodami ni mogoče zbrati.

Glavne slabosti metode:

- Večini podjetij je težko implementirati paketnega vohljača, saj to predstavlja dodatni sloj programske oz. strojne opreme.
- Pri pravilnem zbiranju podatkov bi tudi ta metoda potrebovala delčke kode Javascript, saj na primer shranjene strani (angl. cached pages) ne podajajo zahteve na server, tako da jih vohljač ne more zaznati.

- Paketno vohljanje je najdražja oblika zbiranja podatkov, saj v primeru večih omrežij zahteva posebno instanco programske oz. strojne opreme na vsakem posameznem omrežju.

2.2.2 Oblike namestitve aplikacije

2.2.2.1 Aplikacije na strežniku

Orodja za analizo, ki so nameščena na isti strežnik kot spletna stran (angl. server side), so najlažja za uporabo in so ponavadi že nameščena (brezplačne aplikacije) na strežnik ponudnikov internetnih storitev.

Prednosti aplikacij tega tipa je več. Do njih je mogoče dostopati iz kateregakoli računalnika (za razliko od aplikacij pri klientu), zagotovljena je zasebnost podatkov (za razliko od gostujočih aplikacij), poleg tega pa so hitre ter zanesljive. Omeniti je potrebno še dejstvo, da skoraj vse aplikacije tega tipa uporabljajo spletni dnevnik.

2.2.2.2 Aplikacije pri klientu

Aplikacije pri klientu (angl. client side) so nameščene na računalniku klienta. Njihova prednost je v možnosti hkratnega nadziranja več spletnih domen na različnih strežnikih. Prednosti so tudi v možnosti analiziranja brez uporabe interneta (vsi podatki so že na našem računalniku) ter hitrejši obdelavi podatkov. Hramba vseh podatkov na lastnem računalniku pa lahko prinese tudi negativne posledice, in sicer prostor, ki ga velike količine zbranih podatkov lahko zasedejo, poleg tega pa je vse podatke potrebno prenesti iz spletnega strežnika, kjer se zbirajo. Zbiranje podatkov na lastnem računalniku nam hkrati omogoča večjo zasebnost, po drugi strani pa tudi možnost izgube vseh podatkov v primeru okvare trdega diska računalnika.

2.2.2.3 Gostujoče aplikacije

Gostujoče aplikacije (angl. hosted solutions) so nameščene na strežnik našega ponudnika aplikacij za spletno analizo. So povsem neodvisne od strežnika naših spletnih strani ali od našega računalnika. Vse podatke zbira naš ponudnik, kar ogroža varovanje zasebnosti. Če je družba pripravljena zaupati lastne podatke ponudniku storitve, lahko te podatke veliko lažje zbira, razčlenjevanjuje (angl. parsing), vzdržuje in prenaša, saj za vse skrbi ponudnik. Tak tip aplikacij za zbiranje podatkov v skoraj vseh primerih uporablja označevanje Javascript. Trenutno na trgu najbolj opazna aplikacija tega tipa je Google Analytics, ki je brezplačna.

Tabela 1: Prednosti treh oblik aplikacij [4 – str 74]

Prednost	Apl. na strežniku	Apl. pri klientu	Gostujoča apl.
Brezplačna orodja	•		•
Neodvisna od spletnih dnevnikov			•
Dostop s spletno povezavo	•		•
Dostop brez spleta		•	
Brez spremembe spletnih strani	•	•	
Spletna stran na več strežnikih			•
Neodvisnost od ponudnika	•	•	

2.3 Razvoj spletne analize

Prvi objavljeni spletni program, ki je omogočal spletno analizo je bil »Analog v0.9«, ki je bil izdelan leta 1995. Program je omogočal osnovni pregled matrik zadetkov¹⁰ po urah, dnevih (Slika 5), mesecih in letih.

Daily Summary	
(Go To: Top: Monthly report: Hourly summary: Directory report: Request report)	
Each + represents 200 requests, or part thereof.	
day: #reqs	
---	-----
Sun: 6191:	+++++
Mon: 9488:	+++++
Tue: 9112:	+++++
Wed: 9390:	+++++
Thu: 9329:	+++++
Fri: 8697:	+++++
Sat: 6986:	+++++

Slika 5: Analog v0.9

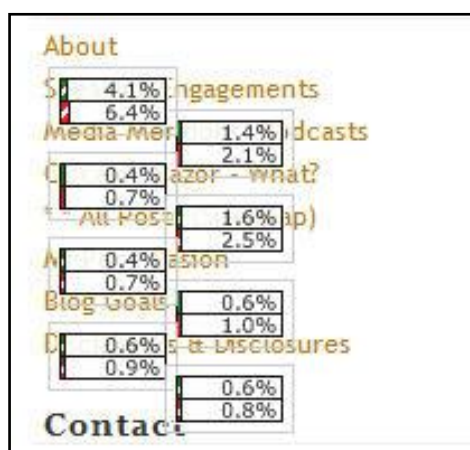
Spletna analiza je postala široko uporabna z začetkom uporabe t.i. »spletnih števecv« (angl. counters). V letih 1995 in 1996 so se števci začeli pojavljati na vseh spletnih straneh in lahko trdimo, da so bili prvi zametki komercialne uporabe spletne analize, saj so dokazovali obiskanosti določene spletne strani.

V naslednjih letih se je uveljavilo že veliko komercialnih, brezplačnih ter odprtokodnih (angl. open source) aplikacij, ki so omogočale bolj zahtevne operacije in delo z velikimi količinami

¹⁰ Zadelek (angl. hit rate) - vsak obisk se zabeleži kot zadelek.

podatkov, ki spremljajo spletno analizo. Najbolj znana podjetja tistega časa, ki delujejo še danes, so WebTrends, Accrue, WebSideStory.

Novo poglavje v spletni analizi je omogočila analize na podlagi označb JavaScript. Do tedaj je bila vsa analiza opravljena na podatkih spletnega strežnika, ki je v posebno datoteko na strežniku (spletni dnevnik) beležil vsako zahtevo. Tehnologija označb Javascript pa omogoča zbiranje podatkov o samem dogajanju (klikanju) uporabnika na strani. To je omogočilo analizo strani na podlagi gostote klikov uporabnikov (Slika 6: Primer gostote klikanja).



Slika 6: Primer gostote klikanja

Prelomno leto na področju spletne analize je bilo leto 2006, ko je Google izdal svojo brezplačno orodje za spletno analizo Google Analytics. Po ocenah, naj bi v prvih 6 mesecih orodje začelo uporabljati pol milijona uporabnikov [1 – str 5]. Napoveduje pa se tudi prihod Microsoftovega orodja za spletno analizo imenovan MS Gatineau.

2.4 Prihodnost spletne analize

Tehnološko gledano nam bo prihodnost spletnega analiziranja prinesla vedno več vizualno-sporočilnih aplikacij, ki bodo administratorjem in razvijalcem pomagale pri ustvarjanju uporabniku prijaznejših spletnih strani. Primer ene izmed takih aplikacij je t.i. toplotna karta (angl. heat map) spletne strani (Slika 7), ki grafično prikazuje kje vse se je zadrževal uporabnik (kliki, premiki miške) na strani.



Slika 7: Primer uporabe toplotne karte

V dosedanjem razvoju orodij za analizo zabeleženih podatkov so razvijalci uporabljali zgozlj »inovacije na podlagi tehnično-podatkovnih zmožnosti« (angl. Possibility-driven innovation – PDI) [1 – str. 7], v prihodnosti pa bo večji poudarek na željah uporabnikov – »uporabniško vodene inovacije« (angl. Customer-driven innovation – CDI) [1 – str. 7]. Aplikacije za spletno analizo bodo postale več kot le orodje za pregledno analiziranje. Postale bodo orodje, ki bo, v želji po večjem obisku, odločilno vplivalo na razvoj spletnih strani podjetij.

3 PRIMERJAVA Z ŽE OBSTOJEČIMI APLIKACIJAMI

Na spletu obstaja veliko aplikacij/sistemov z možnostjo pregleda dostopov do spletnega strežnika. Poleg ločevanja glede na tehničnih značilnostih (zajem podatkov in tip namestitve), jih lahko ločimo tudi glede na vrsto ponudbe, v tri razrede, in sicer na odprtokodne (brezplačni z možnostjo lastne nadgradnje), brezplačne ter plačljive.

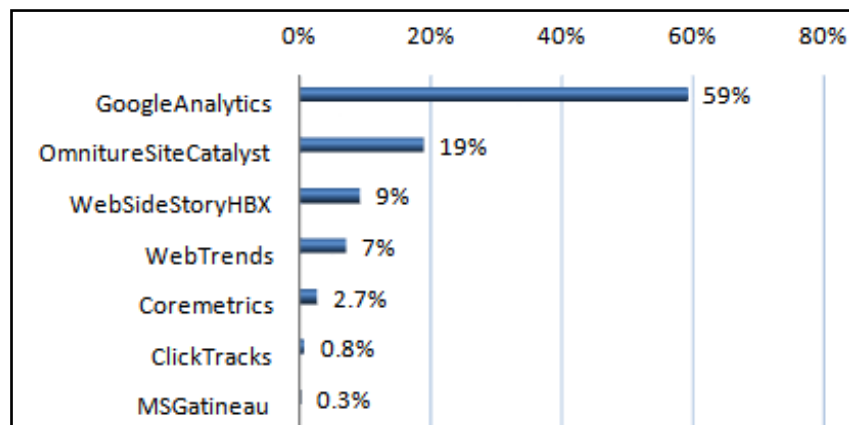
Plačljive aplikacije se naprej razvrščajo glede na način plačevanja (enkratno, po obdobju, po količini podatkov, po številu strani, itd.).

V tem poglavju bodo na kratko predstavljene najpogosteje uporabljene aplikacije/sistemi ter njihove značilnosti, tehnologije in posebnosti, saj bo tako v nadaljnih poglavjih razumevanje sistema te diplomske naloge lažje.

Omeniti velja, da področje spletne analize ni podobno področju urejevalnikom besedil, kjer imamo eno (MS Office - plačljiv) oz. dve (OpenOffice – brezplačen) aplikaciji, ki obvladujeta skoraj celoten tržni delež. Na področju aplikacij za spletno analizo je tržni delež razpršen (Slika 8: Tržni delež aplikacij za spletno analizo (januar 2008) [16]), čeprav v zadnjem času (na tržišče je prišel leta 2006) Google Analytics vedno bolj izstopa, na tržišče pa z velikimi koraki prihaja tudi Microsoftova aplikacija MS Gatineau, za katero je pričakovati, da bo prevzela velik tržni delež.

Tabela 2: Najbolj pogoste aplikacije za spletno analizo

Naziv aplikacije	Zbiranja podatkov	Oblika namestitve	Cena aplikacije	Domača spletna stran
Analog	Spletni dnevnik	Na strežniku	Brezplačna	www.analog.cx
AWStats	Spletni dnevnik	Na strežniku	Odprtokodna	www.awstats.org
Google Analytics	Označbe Javascript	Gostuje	Brezplačna	www.google.com/analytics
OneStatFree	Spletni dnevnik	Gostuje	Brezplačna (omejeno)	www.onestatfree.com
CrazyEgg	Označbe Javascript	Gostuje	Brezplačna (omejeno)	www.crazyegg.com
WebTrends [9]	Hibridno	Opcijsko	Približno 27.000\$/leto	www.webtrends.com
Omniure	Označbe Javascript	Opcijsko	Približno 30.000\$/leto	www.omniure.com
WebLAT	Spletni dnevnik	Na strežniku	Brezplačna	



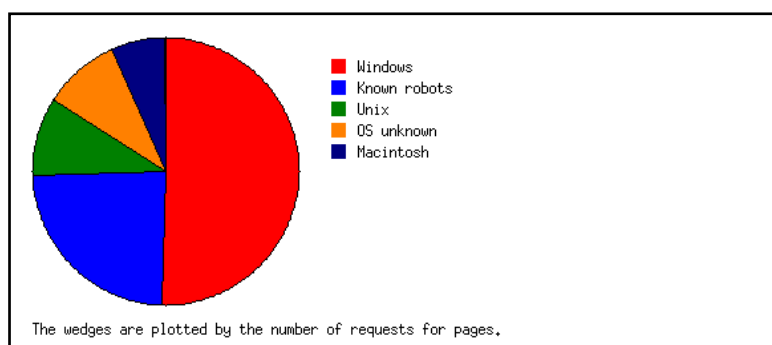
Slika 8: Tržni delež aplikacij za spletno analizo (januar 2008) [16]

3.1 Brezplačne aplikacije za analizo dostopov do strežnika

3.1.1 Analog

Verzija 0.9 te aplikacije je bila prva aplikacija namenjena spletni analizi. Priljubljena je predvsem zaradi hitre ter preproste analize (le najbolj pomembne matrike in grafi - prikazi števila zahtev po dnevih, mesecih, urah, brskalnikih, operacijskih sistemih, itd.). Deluje na vseh operacijskih sistemih in na vseh brskalnikih, poleg tega pa je izpis analize mogoč kar v 32 jezikih.

Analog nam omogoča pregledno osnovno analizo (KPI¹¹) dogajanja na naših spletnih straneh, a od te aplikacije ne smemo pričakovati preveč (Slika 9: Prikaz delovanja aplikacije Analog). Brezplačnost gotovo govori v njen prid, a v tej skupini lahko najdemo tudi aplikacije z večjim naborom funkcionalnih zmožnosti.

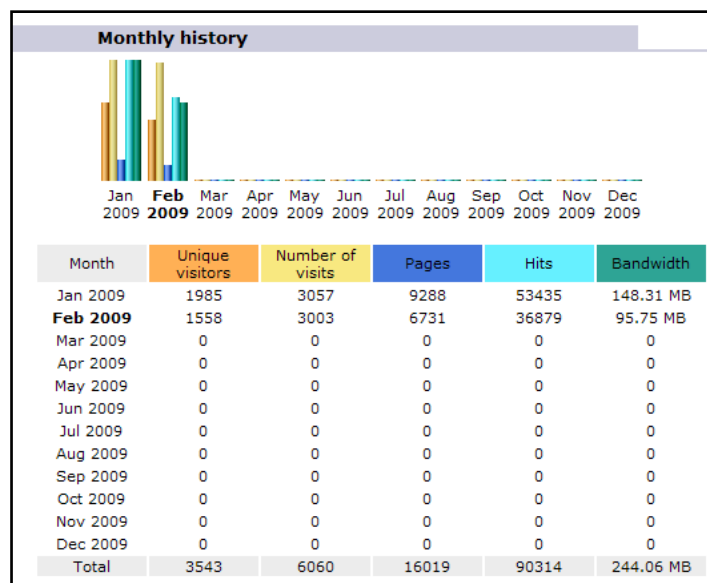


Slika 9: Prikaz delovanja aplikacije Analog

¹¹ KPI (angl. key performance indicators) kratica, uporabljena v svetu spletne analize, ki predstavlja najpomembnejše matrike in grafikone, kot so število obiskovalcev, število obiskov po dnevih, mesecih, itd.

3.1.2 AWStats

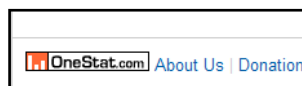
Ta program je zelo podoben prej opisani aplikaciji Analog, le da poleg analiziranja spletnih dnevnikov spletnih strani, omogoča tudi analizo spletnih dnevnikov FTP¹² ter spletne pošte. Poleg tega, da je brezplačen, je tudi licenciran pod GNU¹³ licenco, kar pomeni, da je odprtokodni program. Tudi grafično ponuja več kot Analog (Slika 10: Prikaz delovanja aplikacije AWStats), medtem ko je nabor statističnih podatkov podoben.



Slika 10: Prikaz delovanja aplikacije AWStats

3.1.3 OneStatFree

Aplikacija obstaja v plačniški ter brezplačni obliki. Posebnost te aplikacije, je primerjava statistik s podobnimi (prodajalne podobnih izdelkov, enake teme, itd.) spletnimi stranmi. Vse, kar podjetje, ki nudi aplikacijo zahteva od svojega naročnika, je povezava opremljena z njihovim logotipom nekje na naši spletni strani (Slika 11).



Slika 11: Prikaz povezave do OneStat aplikacije

¹² FTP (angl. - file transfer protocol) protokol za prenos datotek, je programski standard za prenos datotek med računalniki z različnimi operacijskimi sistemi. Spada v aplikacijsko raven internetnega nabora protokolov.

¹³ GNU je tričrkovni rekurzivni akronim: GNU's Not Unix; izbrano je bilo zaradi podobnosti sistema z Unixom, vendar se od njega razlikuje po temu, da GNU vsebuje prosto programje in ne vsebuje kakršnekoli Unixove programske kode. Razvoj projekta GNU je začel Ameriški programer Richard Stallman.

3.1.4 CrazyEgg

Posebnost te aplikacije je popolnoma nov način spremljanja obnašanja uporabnika na naši strani (Slika 7: Primer uporabe toplotne karte). Poleg te pa ponuja še vrsto drugih funkcij, ki jih lahko opazimo tudi pri podobnih tovrstnih aplikacijah, npr. Google Analytics, kot so vsi pomembni KPI-ji ter gostota klikanja uporabnikov po straneh (Slika 6: Primer gostote klikanja).

Program je brezplačen do 5000 obiskov na mesec, nato cena raste glede na število obiskovalcev.

Ker uporablja označbe Javascript, je potrebno vsem stranem, ki jih želimo analizirati, dodati eno vrstico kode Javascript. Dodatne programske opreme ni potrebno namestiti.

3.1.5 Google Analytics

Trenutno najbolj popularna aplikacija za analiziranje dostopov do spletnih strani (Slika 12: Google Analytics nadzorna plošča). Uporablja označevanje Javascript (Označbe JavaScript na strani 10), kar pomeni, da moramo vsako spletno stran ki jo želimo analizirati opremiti z nekaj vrsticami kode Javascript (GATC¹⁴).

Google omogoča brezplačno uporabo z omejitvijo petih milijonov obiskov (na maksimalno 50 različnih straneh) na mesec (dva zadetka na sekundo). Uporaba je možna tudi nad to mejo, vendar Google od nas zahteva uporabo sistema »Adwords«¹⁵.

Google Analytics ima veliko prednosti pred tekmeci. Prva je brezplačnost, a kljub temu ponuja rešitve, katerim je kos le redkokatera plačljiva aplikacija.

Če pri aplikaciji Google Analytics iščemo napake, jih bomo tudi našli. Ena izmed njih je osveževanje podatkov, saj se to zgodi le enkrat na 24 ur, kar pomeni, da podatki niso ažurni. Problematično je tudi varstvo zasebnosti, saj se vsi podatki hranijo na Googlovih strežnikih.

Kot vodilo pri izbiri statističnih podatkov, ki jih prikazuje WebLAT, me je vodil prav sistem Google Analytics. Razlika med sistemoma je v zajemu podatkov (WebLAT – Spletni dnevnik, Google Analytics – označevanje Javascript), zato vsaka aplikacija ponuja prednosti (in slabosti), ki jih način zajema podatkov prinaša.

¹⁴ GATS – Google Analytics Tracking Code. Kratica za Google Analytics koda za spremljanje spletne strani.

¹⁵ »Adwords« - Googlov program za reklamiranje. Glavni vir zaslužka podjetja Google (2007, 16.4 milijarde \$).



Slika 12: Google Analytics nadzorna plošča

3.2 Plačljive aplikacije za analizo dostopov do strežnika

V poglavju brezplačnih aplikacij so omenjene nekatere aplikacije, ki so brezplačne do določene količine prometa, nad tem pa je potrebno plačevati določene prispevke za nadaljno uporabo.

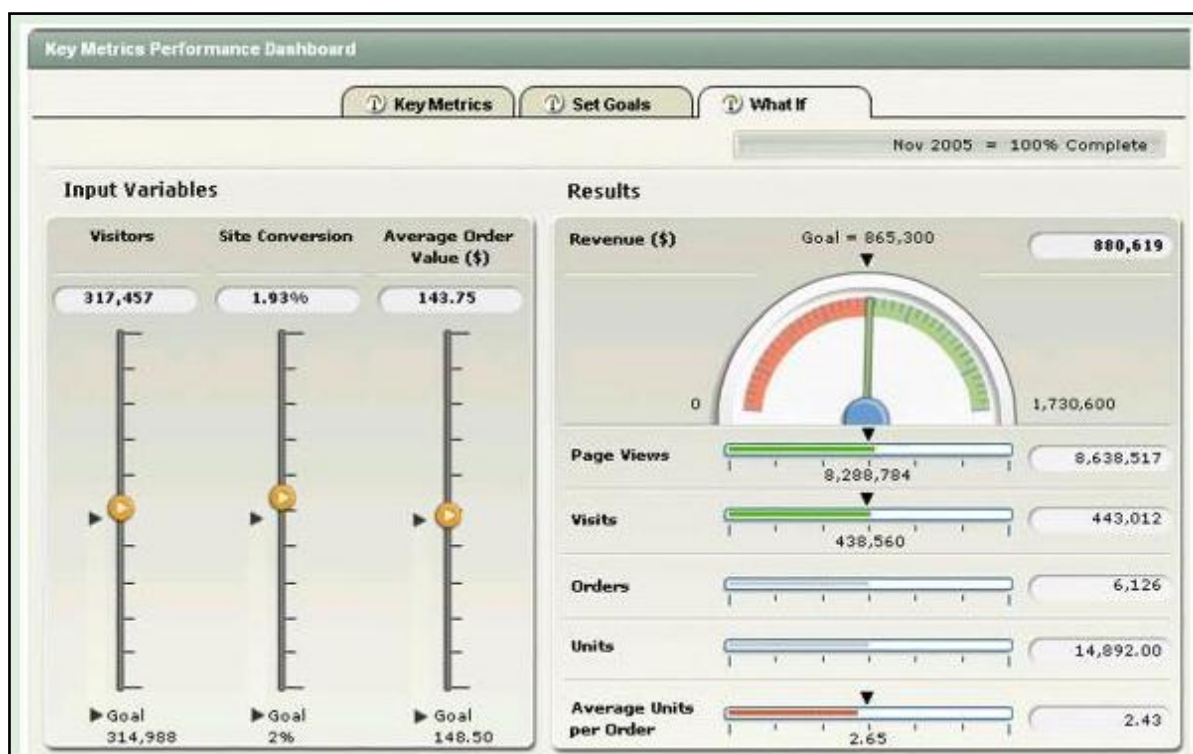
V poglavju plačljivih aplikacij za analizo dostopov so predstavljena nekatera največja podjetja, ki v sklop svojih ponudb poleg aplikacij za spletno analizo vključujejo tudi razvoj spletnih strani in ostalo podporo spletnemu poslovanju. Poleg nudenja samih statistik ponujajo tudi storitve svojih specialistov in izdelave posebnih poročil (angl. customized

reports). Spletna analiza je tako le majhen del ponudbe. Temu primerne so tudi cene, ki se lahko v primeru individualnih pogodb merijo celo v milijonih dolarjev.

3.2.1 WebTrends – Web Analytics 8

Sistem ponuja hibridni zajem podatkov (spletni dnevnik in označevanje Javascript), namestitev na lastnem ali gostujočem strežniku ter ogromno količino KPI matrik in grafikonov. Poleg tega pa nam ponuja »kaj-če« (angl. what-if) analizo, kjer analiziramo, kaj bi se dogajalo na naši strani, če bi jo spreminjali ali imeli drugačne skupine uporabnikov, ali pa le več oz. manj le-teh (Slika 13: WebTrends - Web Analytics 8 nadzorna plošča).

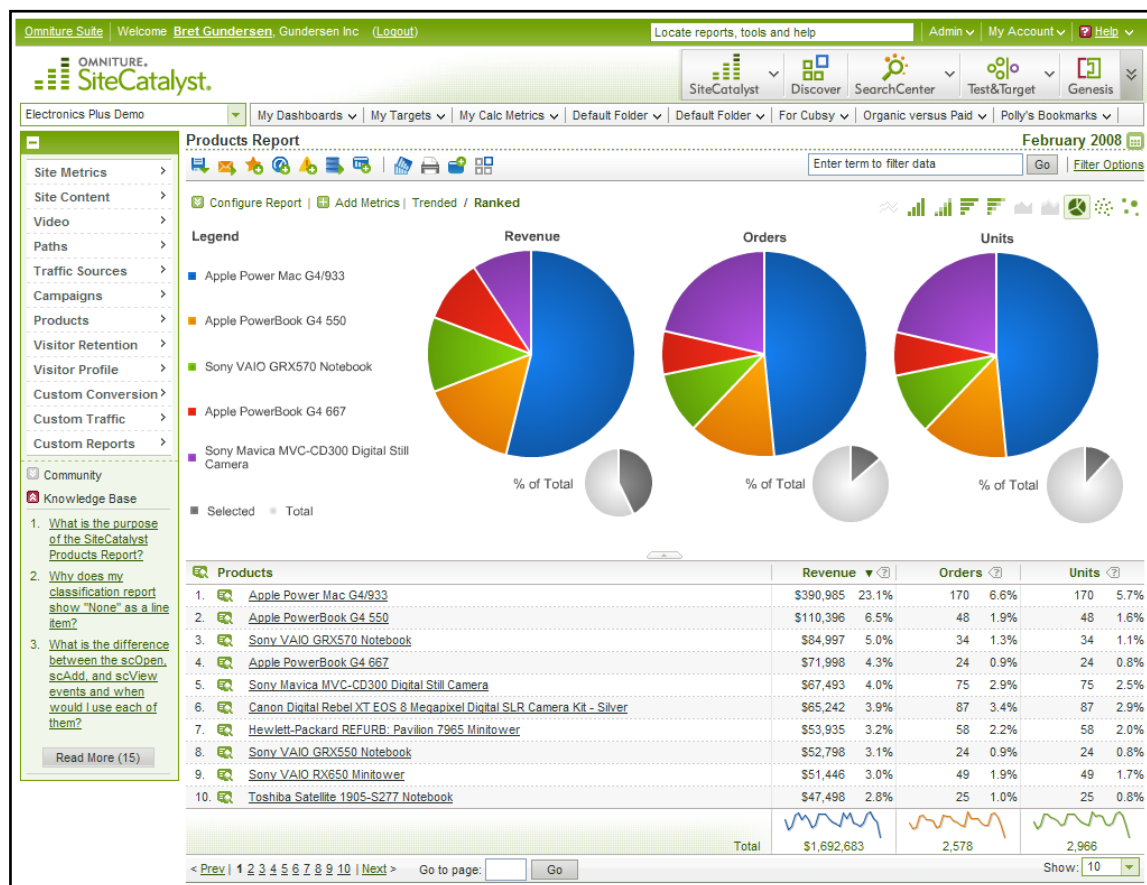
Slabosti so predvsem: potencialna zmeda pri razvoju spletnih strani zaradi kompleksnega modela poročil, zapleteno licenciranje ter dejstvo, da aplikacija ni realnočasna (angl. real time).



Slika 13: WebTrends - Web Analytics 8 nadzorna plošča

3.2.2 Omniture - SiteCatalyst

Sistem (Slika 14: Omniture SiteCatalyst nadzorna plošča) že vsebuje podporo za »Web 2.0«¹⁶, prav tako za posamezne analize socialnih spletnih portalov, blogov, profiliranje uporabnikov ter »bogatih spletnih aplikacij« (angl. rich internet applications).

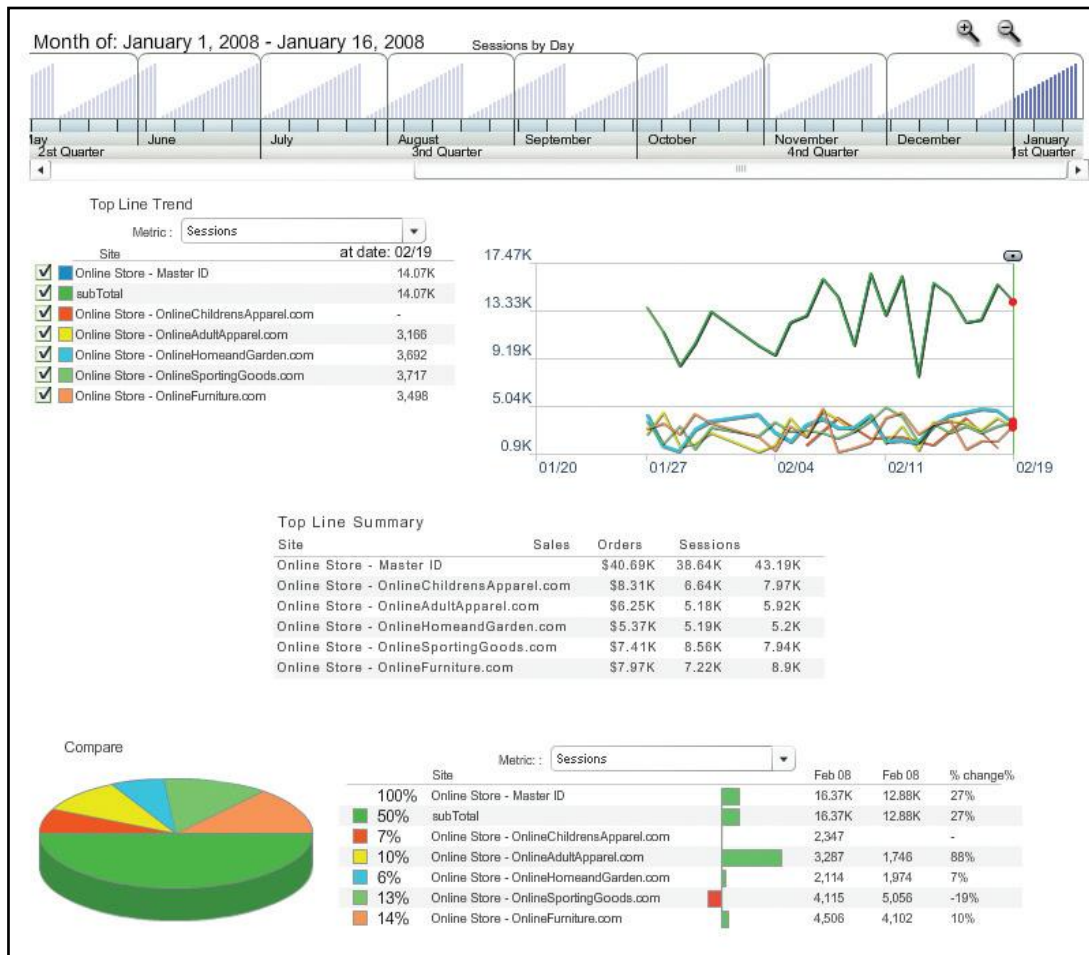


Slika 14: Omniture SiteCatalyst nadzorna plošča

¹⁶ Web 2.0 je del tranzicije svetovnega spleta iz skupka spletnih strani v obširno platformo, ki bo združila spletne ter namizne aplikacije[6 – str 94].

3.2.3 Coremetrics – Online Analytics

Posebnost tega ponudnika je profiliranje »v živo«, kar pomeni spremljanje dogajanja na strani med samim dogajanjem (Slika 15: Coremetrics nadzorna plošča).



Slika 15: Coremetrics nadzorna plošča

4 TEHNOLOŠKA PLATFORMA

Pri izbiri tehnološke platforme je bilo potrebno upoštevati dve zahtevi.

Prva je zahteva naročnika: kompatibilnost z že obstoječimi tehnologijami, na katerih tečejo spletne strani LUI (projekt Orange), druga pa je idejna: odprtokodne tehnologije.

Ti dve zahtevi sta pogojevali uporabo naslednjih strežnikov, programov za podporo relacijskim bazam, programskih jezikov, ter knjižnice uporabljene za grafično oblikovanje uporabniškega vmesnika.

4.1 Relacijska baza: MySQL

Razvojna verzija: MySQL 5.0.67-community-nt via TCP/IP

MySQL [12] je sistem, namenjen upravljanju z relacijskimi podatkovnimi bazami. Gre za odprtokodno implementacijo relacijske podatkovne baze, ki za delo s podatki uporablja jezik SQL¹⁷.

Deluje na principu odjemalec - strežnik, pri čemer lahko strežnik namestimo kot sistem, porazdeljen na več strežnikov. Obstaja veliko število odjemalcev, zbirk ukazov in programskih vmesnikov za dostop do podatkovne baze MySQL.

Glavne značilnosti podatkovnega strežnika MySQL so deklarativnost (povemo kaj in ne kako), relacijski model podatkov (množice), funkcionalna preprostost, enostaven nabor ukazov, zmogljivost (dovolj dobra za večino spletnih aplikacij) in brezplačnost.

Pri povezavi s podatkovnim strežnikom moramo podati osnovne podatke, kot so:

- uporabniško ime
- geslo
- ime podatkovne baze
- gostitelj (angl. host)

¹⁷ SQL ali strukturirani povpraševalni jezik za delo s podatkovnimi bazami (angl. Structured Query Language) je najbolj razširjen in standardiziran povpraševalni jezik za delo s podatkovnimi zbirkami, s programskimi stavki, ki posnemajo ukaze v naravnem jeziku. Določen je z standardom ANSI/ISO SQL. Standard SQL se je razvijal od leta 1986 in danes obstaja več različic. Oznaka SQL-92 se navezuje na standard izdan v letu 1992, SQL:1999 se navezuje na standard izdan leta 1999, SQL:2003 se navezuje na različico iz leta 2003 in tako naprej. Izraz standard SQL uporabljamo za poimenovanje trenutne različice SQL standarda v vsakem časovnem obdobju.

4.2 Spletni strežnik: Apache HTTP server

Razvojna verzija: Apache HTTP server 2.2

Spletni strežnik Apache¹⁸ [15] igra ključno vlogo pri širjenju spleta. Bil je prva alternativa Netscapeovemu spletnemu strežniku, trenutno znanemu kot spletni strežnik Sun Java System. Od aprila 1996 je Apache najpogosteje uporabljen strežnik HTTP na celem spletu, oktobra 2007 je bilo na strežniku Apache postavljenih približno 48% vseh spletnih strani.

Sistem Apache je ustvarila in vzdrževala skupnost neodvisnih razvijalcev, ki so se kasneje povezali v skupnost Apache Software Foundation. Aplikacija je bila prvotno zgrajena za operacijska sistema Unix in FreeBSD, dandanes pa je na voljo tudi za širše uporabljene operacijske sisteme, kot so Microsoft Windows, Solaris, MAC OS-X idr.

4.2.1 Cronolog

Razvojna verzija: Cronolog 1.6.1

Cronolog je preprost program, ki filtrira spletni dnevnik (angl. log file) na podlagi definicije, zapisane v konfiguracijski datoteki strežnika (v našem primeru spletni strežnik Apache in datoteka »httpd.conf«). Program vedno deluje v sodelovanju oz. kot modul spletnemu strežniku, na kateremkoli operacijskem sistemu. Če je tako določeno v definiciji, lahko program zapisuje v več datotek, ločenih glede na datumske vrednosti (v našem primeru se vsak dan odpre nova datoteka (spletni dnevnik) poimenovana po tekočem datumu).

4.3 Programski jezik sloja poslovne logike: Python

Razvojna verzija: Python 2.5.2 (r252:60911)

Python¹⁹ je dinamičen objektno-orientiran programski jezik, ki se ga lahko uporablja za raznovrsten programski razvoj. Velik poudarek jezika je na integraciji z drugimi programskimi jeziki ter orodji.

¹⁸ Ime »Apache« je bilo izbrano iz dveh razlogov:

- po indijanskem plemenu Apačev, znanem po njihovih vojnih sposobnostih
- po setih dodatkov (angl. patches), pa je prišlo ime »a patchy« strežnik

¹⁹Ustvaril ga je Guido van Rossum leta 1990. Jezik je dobil ime po priljubljeni angleški televizijski nanizanki Leteči cirkus Montyja Pythona (Monthy Python's Flying Circus).

[13] Python ima popolnoma dinamične podatkovne tipe, samodejno upravlja s pomnilnikom in podpira funkcionalno, imperativno oziroma proceduralno, strukturirano in objektno orientirano računalniško programsko paradigmo. Zaradi dinamičnih podatkovnih tipov je podoben jezikom Perl, Ruby, Scheme, Smalltalk in Tcl. Razvili so ga kot odprtokodni projekt, ki ga je upravljala neprofitna organizacija Python Software Foundation.

4.3.1 Knjižnica MySQLdb

Razvojna verzija: MySQL-python 1.2.2

Knjižnica, ki omogoča povezavo Python skript s MySQL podatkovno bazo.

4.3.2 Knjižnica Matplotlib

Razvojna verzija: matplotlib 0.98.5.2

Knjižnica Matplotlib, ki omogoča risanje grafov v programskem jeziku Python in pomeni razširitev numerične matematične knjižnice NumPy. Omogoča API²⁰ »pylab«, ki je zelo podoben MATLAB-u, ter je tako primeren za vse, ki jim je domače okolje MATLAB. Matplotlib je napisal in ga vzdržuje John Hunter in je distribuiran pod licenco BSD²¹.

4.3.3 Knjižnica PIL – Python Imaging Library

Razvojna verzija: PIL 1.1.6

Knjižnica zagotavlja podporo za odpiranje, obdelavo in shranjevanje datotek vseh formatov, v katerih se shranjujejo slike.

Podprti formati datotek: PNG, JPEG, GIF, TIFF, BMP.

4.3.4 Knjižnica Basemap toolkit

Razvojna verzija: basemap 0.99.3

»Matplotlib basemap toolkit« je knjižnica namenjena risanju dvodimezionalnih podatkovnih tipov na zemljevide v programskem jeziku Python. Basemap ne opravlja risanja sam, pač pa omogoča pretvorbo koordinat na enega od 22 različnih projekcij zemljevidov. Knjižnica Matplotlib je nato uporabljena za obrisovanje, risanje grafov, linij ali točk na podane koordinate. Podane so tudi linije obal, rek ter političnih meja, skupaj z metodami za njihovo risanje.

²⁰ API - kratica za (angl. application programming interface) vmesnik uporabniškega programa.

²¹ BSD (Berkeley Software Distribution) licenca skoraj nima omejitev. Vsakdo lahko kodo spremeni in jo ni dolzan dati na voljo drugim, kot je to pri GPL (General public licence).

GEOS knjižnica je uporabljena interno za dodajanje političnih meja, rek ter obal na izbrano projekcijo zemljevida ter regije.

4.3.5 Knjižnica Numpy

Razvojna verzija: numpy 1.2.1

NumPy je temeljni paket, potreben za matematične namene programiranja s programskim jezikom Python.

4.3.6 Knjižnica ReportLab toolkit

Razvojna verzija: reportlab 2.3

Knjižnica ReportLab toolkit v programskem jeziku Python podpira delo z datotekami PDF. Omogoča ustvarjanje datotek PDF z možnostjo dodajanja vseh gradnikov (tabel, različnih stilov črk, vektorske grafike, slike v različnih tipih datotek (PNG, JPG, BMP, itd.)).

4.4 Povezava Python – Apache: modul mod_python

Razvojna verzija: mod_python 3.3.1

Mod_python [11] je Apachejev modul, ki vsebuje vmesnik do programskega jezika Python znotraj strežnika, kar omogoča izvajanje skripte Python na strežniku. Mod_python omogoča pisanje spletnih aplikacij v jeziku Python, ki tečejo veliko hitreje kot tradicionalni CGI²² in ima napredne funkcije, kot npr. možnost držanja povezave (angl. connection handler) s podatkovno bazo.

Pri opisu mod_python modula je potrebno opozoriti, da je napisan za spletni strežnik Apache in na drugih spletnih strežnikih ne deluje.

²² CGI je tričrkovna kratica za Common Gateway Interface, ki označuje skupek pravil, katera določajo, kako spletni strežnik komunicira z ostalo programsko opremo in obratno. Navadno CGI program bere podatke iz baze ali kakega drugega vira, jih predela, ter pošlje v obliki HTML na izhod. Skripte, ki uporabljajo ta vmesnik kot orodje, so bolj znane kot CGI-skripte. Na CGI-ju tečejo kot aplikacije tudi programski jeziki kot so Perl, MIVA, PHP in Python.

4.5 Programski jeziki sloja uporabniškega vmesnika

Sloj uporabniškega vmesnika je spletna stran, ki pa ima zaradi želje po prijaznem ter hitrim uporabniškim vmesnikom vgrajene najnovejše tehnologije na področju programiranja spletnih strani.

4.5.1 HTML

HTML (angl. Hypet Text Markup Language) [5] je označevalni jezik, namenjen spletnih aplikacij in predstavlja osnovo vsakega spletnega dokumenta. S pomočjo HTML-ja ustvarimo strukturo in semantično ureditev dokumenta. Pišemo ga lahko v vsakem urejevalniku besedil in je dokaj preprost za učenje in uporabo.

Jezik HTML so osnovali znanstveniki pod vodstvom Tima Berners-Leeja leta 1990 v Švici, v Evropskem centru za fiziko osnovnih delcev (CERN). Ker so si hoteli najnovejša spoznanja med seboj izmenjavati kar najhitreje, so potrebovali dovolj hiter način za izmenjavo podatkov - besedil, slik in zvoka. Tega pa dotlej internet ni omogočal, saj je bilo do takrat namreč mogoče uporabljati le določene ukaze ali pošiljati preprosta tekstovna sporočila. Za osnovo jezika HTML so uporabili jezik SGML (angl. Standard Generalized Markup Language). Tako so se v začetku devetdesetih let začele pojavljati prve barvne spletne strani in povezave med besedili. S pomočjo obogatenih besedil (angl. hypertext) je bilo mogoče med seboj povezati tekstovne ali slikovne strani. HTML jezik nam je tako omogočil izdelavo obogatene besedila in je močno povezan z razvojem svetovnega spleta.

Gradniki HTML jezika so oznake (angl. tags). Te so dveh tipov:

- Samostojne značke (samostojna značka - npr. Vrstica
 Nova vrstica)
- Dvojne značke (začetna in končna značka - npr. krepka pisava)

4.5.2 Javascript

[10] Objektni skriptni programski jezik JavaScript, ki ga je razvil Netscape z namenom pomagati programerjem pri ustvarjanju interaktivnih spletnih strani.

Jezik je bil razvit neodvisno od Jave, vendar si z njo deli številne lastnosti in strukture. JavaScript lahko sodeluje s HTML-kodo in s tem poživi stran z dinamičnim izvajanjem.

JavaScript je podprt s strani velikih programskih podjetij in kot odprt jezik ga lahko uporablja vsakdo, ne da bi pri tem potreboval licenco. Podpirajo ga vsi novejši spletni brskalniki.

Sintaksa jezika JavaScript ohlapno sledi programskemu jeziku C. Prav tako kot C JavaScript nima vgrajenih vhodno izhodnih funkcij, zato je izvedba teh funkcij odvisna od gostitelja.

JavaScript se veliko uporablja za ustvarjanje dinamičnih spletnih strani. Program se vgradi ali pa vključi v HTML z namenom, da opravlja naloge, ki niso mogoče s samo statično stranjo, na primer odpiranje novih oken, preverjanje pravilnost vnešenih podatkov, enostavni izračuni, itd. Na žalost različni spletni brskalniki izpostavijo različne objekte za uporabo, zato je za podporo vseh brskalnikov potrebno napisati več različic funkcij.

Zunaj spleta se JavaScript uporablja v različnih orodjih. Adobe Acrobat in Adobe Reader podpirata jezik v PDF-datotekah. Podpirata ga tudi operacijska sistema Microsoft Windows in Mac OS X.

Ti programi oz. operacijski sistemi imajo svoje objektne modele, ki zagotavljajo dostop do gostiteljevega okolja, samo jedro jezika JavaScript pa je v vseh programih večinoma enako.

4.5.3 AJAX

[7] Ajax - (asinhroni JavaScript in Xml) je skupina medsebojno povezanih spletnih razvojnih tehnik, namenjenih ustvarjanju interaktivnih spletnih aplikacij. Z AJAX-om si lahko spletne aplikacije izmenjujejo podatke s strežnikom asinhrono v ozadju, brez potrebe po ponovnem nalaganju strani. S tem je mogoče tekoče in hitrejše spremljanje ter spreminjanje vsebine na spletni strani. Podatki se prenašajo s pomočjo objektov razreda XMLHttpRequest ali s pomočjo »Remote Scripting-a« (v starejših brskalnikih, ki ne podpirajo tehnologije Ajax). Uporaba tehnologij AJAX je tipična za Web 2.0. Navkljub imenu uporaba tehnologij Javascript in XML ni pogoj za izvajanje Ajax-a.

Aplikacije Ajax dajejo vtis, kot da v celoti tečejo na računalniku uporabnika. Običajna spletna aplikacija namreč za vsako spremembo na strani pošlje HTTP zahtevo, strežnik pa kot odgovor pošlje celotno stran. Brskalnik mora zato osvežiti celotno stran, s tem pa pride do

motečega obnavljanja strani. Spletna aplikacija je zato počasna in prav nič podobna namiznim (Slika 16: Primerjava spletne komunikacije).

Ajax aplikacije so prirejene za generiranje poizvedb za strežnik, tako da pošljejo samo tiste podatke, ki jih dejansko potrebujejo. Klic se opravi kot asinhrona komunikacija, torej lahko uporabnik medtem, ko aplikacija čaka podatke iz strežnika, nemoteno uporablja spletno stran. Ko so podatki pripravljene, določena funkcija v JavaScript prikaže podatke na strani brez potrebe po ponovnem nalaganju celotne strani.

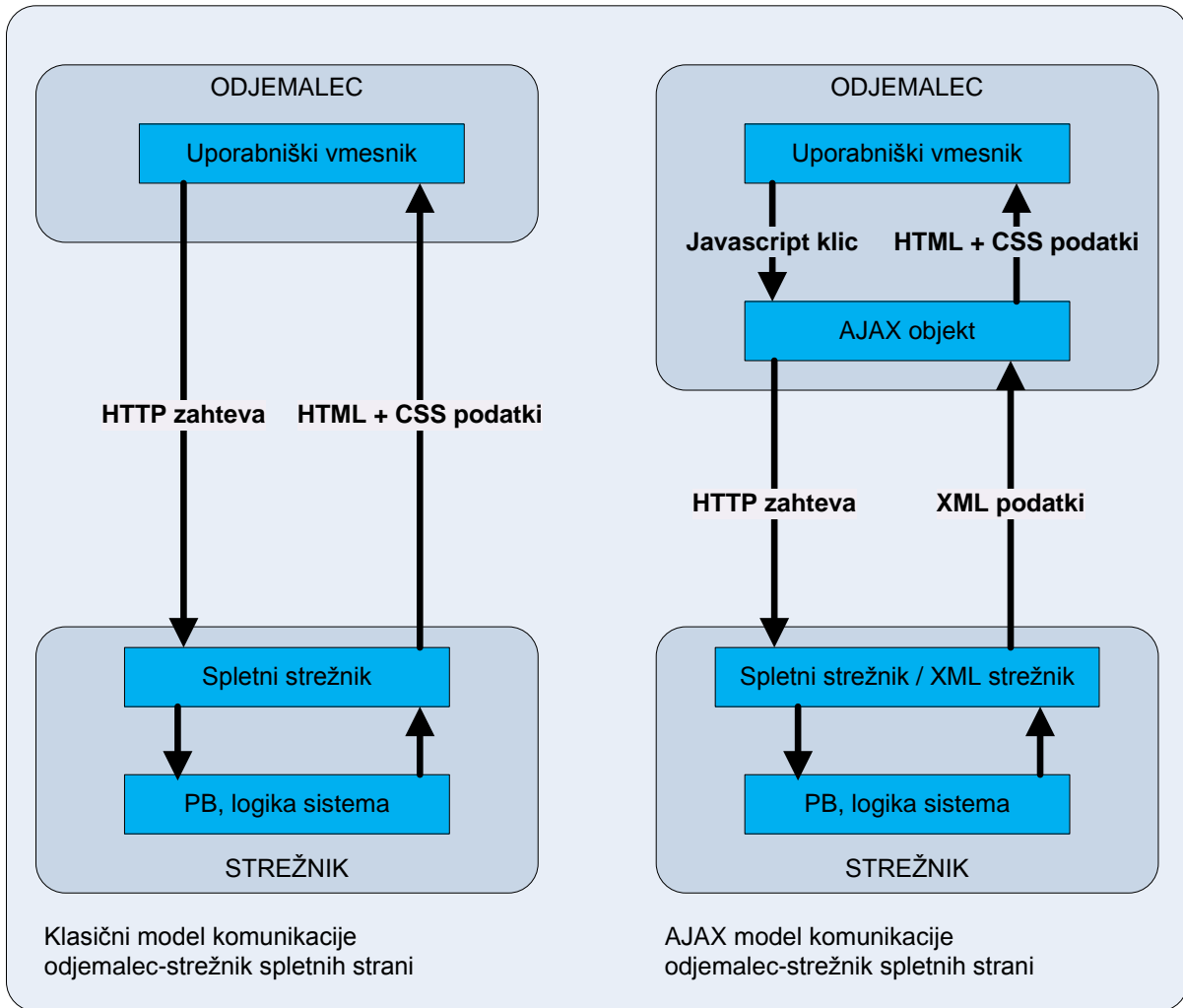
Posledica tega je uporabniški vmesnik, ki se veliko hitreje odziva na vnose uporabnika. Razlog za vpeljavo te tehnologije je tudi dejstvo, da se med odjemalcem (brskalnikom) in strežnikom prenese veliko manj podatkov, ter da poteka nalaganje podatkov asinhrono. Poleg tega se zmanjša obremenitev spletnega strežnika, ker se veliko obdelave lahko naredi na strani odjemalca.

Vodilno pri uporabi tehnologije Ajax je podjetje Google. Primeri uporabe v njihovih izdelkih:

- Google Mail
- Google Suggest
- Google Docs

Večina popularnih spletnih strani iz generacije Web 2.0 uporablja Ajax:

- Flickr
- Del.icio.us
- Facebook



Slika 16: Primerjava spletne komunikacije

4.5.4 CSS

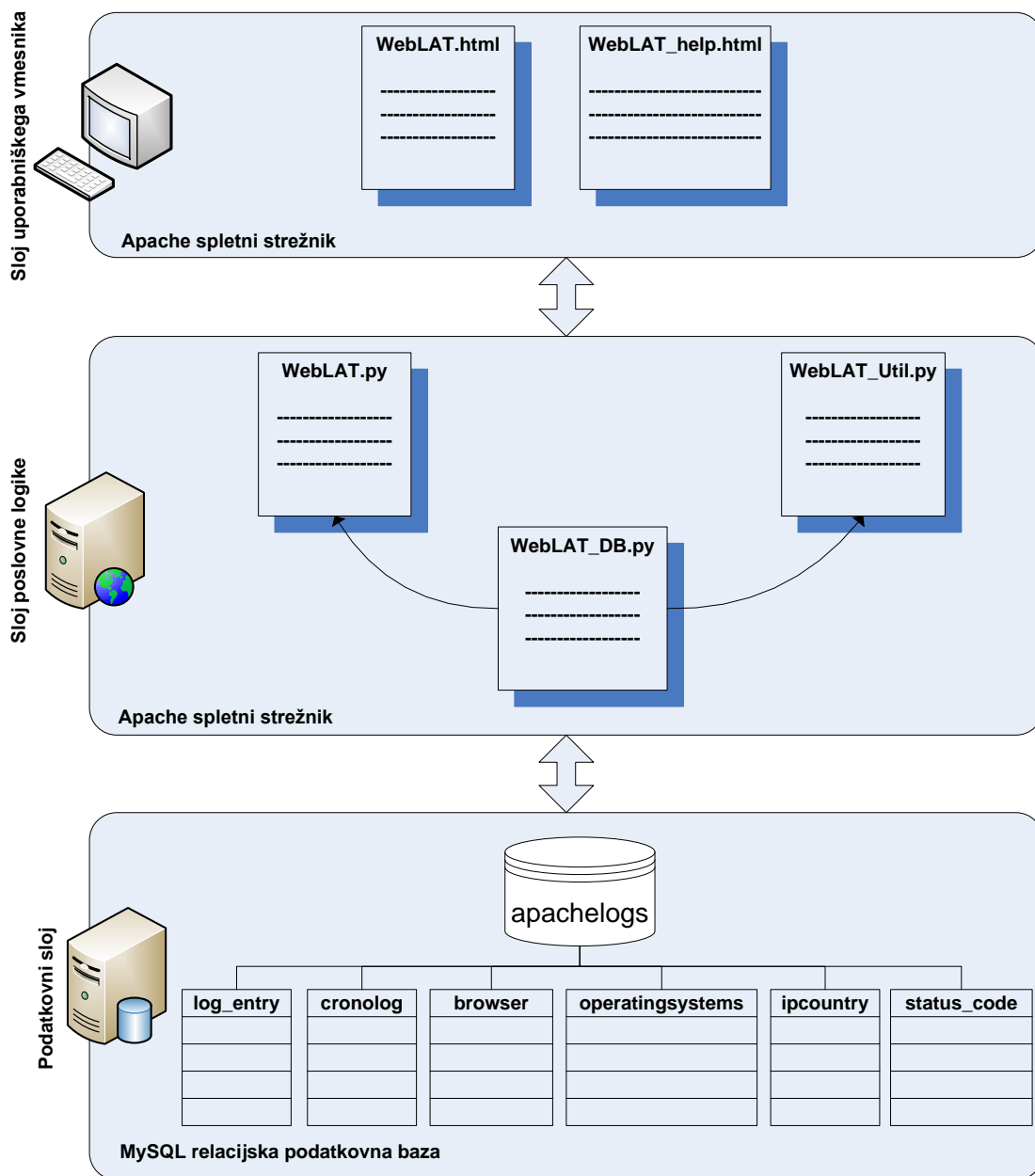
[8] Cascading Style Sheets (kratica CSS) so predloge, ki določajo izgled spletnih strani. Z njimi določamo pisavo, velikosti črk ter vizualno predstavitev spletne strani. HTML naj bi predstavljal semantično strukturo in smiselno hierarhijo dokumenta, CSS pa predstavitevno vlogo.

CSS podpirajo novejši spletni brskalniki (Internet Explorer 4.0 in novejši, Netscape Navigator 4.72, Mozilla, Firefox, Opera). Spletni brskalniki, ki standarda CSS ne podpirajo, bodo prikazali običajen HTML-dokument brez oblikovanja.

Kot pri Javascriptu, je tudi CSS možno vključevati na dva načina:

- Posebna datoteka s CSS
- Znotraj dokumenta HTML v označbi `<style type='text/css'>`

5 ARHITEKTURA SISTEMA



Slika 17: Trinivojska arhitektura

Arhitektura sistema, razvitega v okviru diplomske naloge, je večnivojska (Slika 17: Trinivojska arhitektura). Pri večnivojski arhitekturi kodo, ki dostopa do podatkov, ločimo od poslovne logike in kode uporabniškega vmesnika. S tem dosežemo boljše vzdržljivost aplikacije in lažjo implementacijo bodočih zahtev ter popravkov.

Na najnižjem nivoju oz. sloju je **podatkovni sloj**, kjer se nahajajo podatki v obliki relacijske podatkovne baze (MySQL). Nad podatkovnim slojem se nahaja **sloj poslovne logike**. Ta sloj skrbi za povezovanje uporabnika in podatkovne baze ter generiranje grafov in matrik na podlagi zahtev uporabnika. Najvišji nivo je **sloj uporabniškega vmesnika**, ki je edini sloj, do katerega neposredno dostopa uporabnik.

5.1 Podatkovni sloj

Podatkovni sloj je realiziran z relacijsko podatkovno bazo MySQL. Podatki so **shranjeni** v bazi *apache logs*. **Zbiranje** podatkov pa poteka v načinu spletnih dnevnikov.

5.1.1 Dnevniška datoteka

Dnevniška datoteka (angl. log file) se zapisuje na spletnem strežniku Apache ob vsaki zahtevi HTTP. Glede na možnosti zbiranja najrazličnejših podatkov je treba strežniku definirati obliko zapisa vanjo. Oblika, ki jo uporabljamo v našem sistemu, je naslednja:

```
LogFormat "%{UNIQUE_ID}e\t %h\t %Y-%m-%d %T\t %s\t %>s\t %m\t %U\t %q\t %H\t %{User-Agent}i\t %{Referer}i\t %{Cookie}i\t %X\t %T\t %D\t %B\t" WebLAT
```

Tabela 3: Razčlenitev definicije dnevniške datoteke

Naziv	Koda	Tip	Opis
uniqueid	%{UNIQUE_ID}e	Niz	<i>Ni v uporabi</i>
remotehost	%h	Niz	IP številka
datetime	%Y-%m-%d %T	Niz	Datum in čas zapisa
statusbeforededir	%s	Cifra	Status pred preusmerjanjem
statusafterredir	%>s	Cifra	Status po preusmerjanju
requestmethod	%m	Diskretna spr.	Tip zahteve (GET,HEAD,POST,TRACE)
URLpathrequested	%U	Niz	Zahtevana pot
querystring	%q	Niz	Povpraševalni niz v naslovu zahteve
protocol	%H	Diskretna spr.	Tip protokola
useragent	{User-Agent}i	Niz	Brskalnik
referer	{Referer}i	Niz	Stran, s katere je prišla zahteva
cookie	{Cookie}i	Niz	Piškotek uporabnika
connectionstatus	%X	Diskretna spr.	Status, ko je zahteva opravljena (X,+,-)
seconds	%T	Cifra	Čas v sekundah
microseconds	%D	Cifra	Čas v microsekundah
bytescontent	%B	Cifra	Količina prenesenih podatkov v bytih

5.1.1.1 Primer zapisa vrstice v log datoteki

```
-      189.130.119.228      2009-02-17 00:00:27  200  200  GET  /orange/main.asp
      HTTP/1.1      Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.0.6)
Gecko/2009011913 Firefox/3.0.6
http://www.google.fr/search?num=100&hl=fr&q=ORANGE+data+mining&btnG=Rechercher&meta=
      X      0      328125  12909
```

5.1.2 Shranjevanje podatkov

Za razumevanje shranjevanja podatkov je potrebno vedeti naslednje:

- Program Cronolog skrbi za pisanje dnevniških datotek vsak dan v novo datoteko (npr. *access2009-03-13.log*, *access2009-03-13.log*).
- Naloga (angl. task) v sklopu operacijskega sistema dnevno kliče skripto Python (*WebLAT_Scheduler.py*), ki poskrbi, da se podatki zgoraj omenjene datoteka vnesejo v podatkovno bazo.

5.1.3 Tabele podatkov relacijske baze

Entitetni model (Slika 18) sestavlja šest tabel, od katerih so štiri tabele šifrantov (*ipcountry*, *status_code*, *operatingsystems* ter *browser*), tabela v kateri se hranijo vsi podatki o zahtevah na strežniku (*log_entry*) ter tabela, ki hrani podatke o avtomatskem vstavljanju dnevniških datotek v podatkovno bazo (*cronolog*).

Tabela *ipcountry*

Tabela šifrantov, ki omogoča pretvarjanje IP naslova v državo ter kontinent, od koder je prišla zahteva na strežnik. Vse pretvorbe potekajo v času vpisa dnevniške datoteke v relacijsko bazo, v tabelo *log_entry*, kjer se država vpiše kot atribut *country*, kontinent pa kot atribut *continent*.

Podatki o šifrantu so bili pridobljeni s spletne strani <http://ip-to-country.webhosting.info/> dne 3.12.2008.

Tabela *operatingsystems*

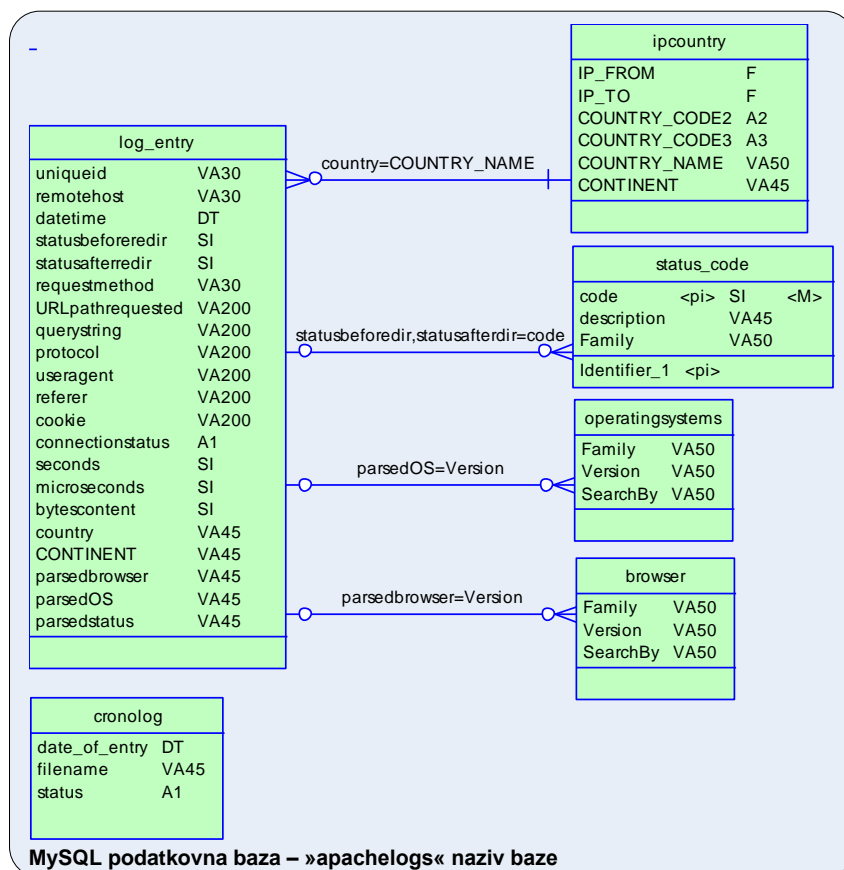
Šifrant operacijskih sistemov je namenjen pretvorbi atributa *useragent* tabele *log_entry* v atribut *parsedOS* (*parsedOS* = *Version*). V šifrantu so navedeni vsi do sedaj poznani operacijski sistemi ter najpogostejši »search bot«-i (*Family*=«Search bot«).

Podatki o šifrantu so bili zbrani na različnih spletnih straneh v decembru 2008.

Tabela *browser*

Šifrant spletnih brskalnikov je namenjen pretvorbi atributa *useragent* tabele *log_entry* v atribut *parsedbrowser* (*parsedbrowser* = *Version*). V šifrantu so navedeni vsi do sedaj poznani spletni brskalniki.

Podatki o šifrantu so bili zbrani po različnih spletnih straneh v decembru 2008.



Slika 18: Entitetni model

Tabela *cronolog*

Informativna tabela, kjer so zapisane vse pretvorbe dnevniških datotek v podatkovno bazo.

date_of_entry: Datum zapisa v podatkovno bazo

filename: Ime dnevniške datoteke.

status: Status zapisa ([1] – uspešen, [0] – neuspešen)

Tabela *log_entry*

Tabela, v kateri so zbrani vsi dostopi do spletnega strežnika. Je osrednja tabela aplikacije in se uporablja ob vsakem izrisu grafa ali matrike.

Tabela *status_code*

Šifrant, ki dodaja nazive statusnim kodam protokola, zabeleženim v dnevniški datoteki. Vse pretvorbe potekajo v času vpisa dnevniške datoteke v relacijsko bazo, v tabelo *log_entry*, kjer se status vpiše kot atribut *parsedstatus* v obliki [XXX-Naziv] ([code-description]).

Podatki o šifrantu so bili pridobljeni s spletne strani:

<http://www.w3.org/Protocols/rfc2616/rfc2616.html> dne 3.12.2008

Tabela 4: Seznam vseh kod statusa

1xx – Informational 100 – Continue 101 – Switching Protocols	4xx – Client Error 400 – Bad Request 401 – Unauthorized 402 – Payment Required 403 – Forbidden 404 – Not Found 405 – Method Not Allowed 406 – Not Acceptable 407 – Proxy Authentication Required 408 – Request Timeout 409 – Conflict 410 – Gone 411 – Length Required 412 – Precondition Failed 413 – Request Entity Too Large 414 – Request-URI Too Long 415 – Unsupported Media Type 416 – Requested Range Not Satisfiable 417 – Expectation Failed
2xx – Successful 200 – OK 201 – Created 202 – Accepted 203 – Non-Authoritative Information 204 – No Content 205 – Reset Content 206 – Partial Content	
3xx – Redirection 300 – Multiple Choices 301 – Moved Permanently 302 – Found 303 – See Other 304 – Not Modified 305 – Use Proxy 306 – (Unused) 307 – Temporary Redirect	5xx – Server Error 500 – Internal Server Error 501 – Not Implemented 502 – Bad Gateway 503 – Service Unavailable 504 – Gateway Timeout 505 – http Version Not Supported

5.2 Sloj poslovne logike

Poslovna logika je realizirana v obliki skript v programskem jeziku Python, ki tečejo na spletnem strežniku Apache. Povezavo spletnega strežnika ter programskega jezika Python programskega jezika omogoča modul `mod_python`.

Poslovna logika vsebuje tri skripte, ki so razdeljene glede na namenskost. *WebLAT_DB.py* je skripta z metodami, ki povezujejo poslovno logiko s podatkovnim slojem. *WebLAT.py* je skripta z metodami, ki povezujejo poslovno logiko s slojem uporabniškega vmesnika ter metodami, ki izrisujejo grafe in matrike. Skripta *WebLAT_Util.py* pa vsebuje podporne metode za drugi dve skripti.

V nadaljevanju so opisane vse metode zgoraj navedenih skript.

5.2.1 WebLAT_DB.py

LogFile2DB(fileName)

Vrne: /

Ključne metode: LogFileAddParsedData

Opis delovanja: Metoda kot argument sprejme ime dnevniške datoteke in zapise iz datoteke zapiše v podatkovno bazo.

LogFileAddParsedData(fileName)

Vrne: /

Ključne metode: /

Opis delovanja: Metoda kot argument sprejme ime dnevniške datoteke in kreira novo datoteko, kjer so dodana polja razčlenjenih atributov (država, kontinent, operacijski sistem, brskalnik, status koda).

totalDirInsert()

Vrne: /

Ključne metode: LogFile2DB

Opis delovanja: Metoda preišče celoten direktorij in vstavi vse dnevniške datoteke (*.log) v podatkovno bazo. Uporabna je za administrativne naloge.

5.2.2 WebLAT_Util.py

Get_DBconn()

Vrne: String, String, String, String

Ključne metode: /

Opis delovanja: Metoda iz konfiguracijske datoteke *Dbconn* prebere vrednosti (host, user, passwd, db), potrebne za povezavo na podatkovno bazo.

Get_ticks(par, listX)

Vrne: list[]

Ključne metode: /

Opis delovanja: Metoda numerične vrednosti parametrov dnevov, ur in mesecev spremeni v uporabniku prijaznejše nize (npr. dan 0 postane »Sun«).

Get_label()

Vrne: String

Ključne metode: /

Opis delovanja: Metoda vrne parameter za prikaz na grafikonu v uporabniku prijaznejšem formatu.

Get_percentageList(listY2D, type)

Vrne: list[[]], list[], list[], int

Ključne metode: /

Opis delovanja: Metoda preračunava odstotkovni vrednosti podane tabele glede na podan tip preračunavanja (po X osi, po Y osi, preko celotne tabele).

parseSimpleHTML(html)

Vrne: list[[]], list[[]]

Ključne metode: /

Opis delovanja: Metoda podani niz v HTML obliki razčleni v dve tabeli. Prva je tabela argumentov uporabljenega filtra, druga pa je tabela prikazanih statističnih podatkov.

parseAdvancedHTML(html)

Vrne: list[[]], String, list[[]]

Ključne metode: /

Opis delovanja: Metoda podani niz v HTML obliki razčleni v tabelo argumentov uporabljenega filtra, niz naslova do shranjenega grafa ter v primeru, da je prikazana matrika, v tabelo podatkov matrike.

parseComplexHTML(html)

Vrne: list[[]]

Ključne metode: /

Opis delovanja: Metoda podani niz v HTML obliki razčleni v dvodimenzionalno tabelo.

createSelect(ChartType, par1, par2, par3, whereCIs)

Vrne: String

Ključne metode: createColumnForSelect

Opis delovanja: Metoda sestavi stavek »SELECT« za poizvedbo, ki vrne podatke za izris grafikona.

createColumnForSelect(par, colName)

Vrne: String

Ključne metode: /

Opis delovanja: Podporna metoda metodi *createSelect*, ki vrača del stavka SELECT.

iff(strFind, strInput, str)

Vrne: String

Ključne metode: /

Opis delovanja: Metoda, ki išče podniz v nizu. Če ga najde, vrne prvi parameter, v nasprotnem primeru drugega.

calculatePercentage(listY2D)

Vrne: list[[]]

Ključne metode: /

Opis delovanja: Metoda iz podane tabele preračuna odstotkovne vrednosti polj in jih vpiše v novo tabelo, ki jo nato vrne.

trimListMap(dataList, map)*Vrne:* list[][]*Ključne metode:* /*Opis delovanja:* Metoda izbere podatkovno tabelo glede na podano vrsto zemljevida.unitConverter(unit, value)*Vrne:* String*Ključne metode:* /*Opis delovanja:* Metoda na podlagi podane vrednosti in enote pretvori enoto v lažje čitljivo vrednost.**5.2.3 WebLAT.py**PDFFooter(req, filename, html)*Vrne:* /*Ključne metode:* /*Opis delovanja:* Metoda doda »nogo« vsaki strani poročila v formatu PDF.Call PDF Simple(html, req)*Vrne:* String [v »Ajax Request«]*Ključne metode:* parseSimpleHTML*Opis delovanja:* Metoda razčleni podani niz v obliki HTML in ustvari datoteko PDF ter vrne njeno ime.Call PDF Advanced(html, req)*Vrne:* String [v »Ajax Request«]*Ključne metode:* parseAdvancedHTML*Opis delovanja:* Metoda razčleni podani niz v obliki HTML in ustvari datoteko PDF ter vrne njeno ime.Call PDF Complex(html, req)*Vrne:* String [v »Ajax Request«]*Ključne metode:* parseComplexHTML*Opis delovanja:* Metoda razčleni podani niz v obliki HTML in ustvari datoteko PDF ter vrne njeno ime.

Call_Simple(req req, date from, date to, ignoresearchbots, ignoreunknownuseragent, succrequests, failrequests, showGeneral, showSucc, showFail, rankCountry, rankContinent, rankBrowser, rankOS, rankProtocol, rankReqMethod, rankConnStat, rankStatus)

Vrne: String [v »Ajax Request«]

Ključne metode: GeneralSummary

Opis delovanja: Metoda iz parametrov, podanih preko spletne strani, generira niz v obliki HTML za prikaz na spletni strani.

Call_Advanced(req, date from, date to, size from, size to, request from, request to, URL, Referer, IP, ignoresearchbots, ignoreunknownuseragent, CountryList, ContinentList, BrowserList, OSList, ProtocolList, RequestMethodList, ConnectionStatusList, StatusCodeList, chartType, par1, par2, par3, par4, Grid, Legend, SizeX, SizeY)

Vrne: String [v »Ajax Request«]

Ključne metode: iff, createSelect, C1_Line_SQL, C2_Area_SQL, C3_Pie_SQL, C4_Bubble_SQL, C5_Column_SQL, C6_ColumnStacked_SQL, C7_Map_SQL, C8_Table_SQL

Opis delovanja: Metoda iz parametrov, podanih preko spletne strani, generira niz v obliki HTML za prikaz na spletni strani. Znotraj HTML-ja je podana tudi pot do slike grafikona, ki je bil ustvarjen za poizvedbo.

Call_Complex(req, select)

Vrne: String [v »Ajax Request«]

Ključne metode: /

Opis delovanja: Metoda iz parametrov, podanih preko spletne strani, generira niz v obliki HTML za prikaz na spletni strani.

Call_Test()

Vrne: /

Ključne metode: createSelect

Opis delovanja: Metoda namenjena testiranju.

GeneralSummary(dateFrom, dateTo, ignoresearchbots, ignoreunknownuseragent)

Vrne: String

Ključne metode: iff, unitConverter

Opis delovanja: Metoda na podlagi podanih parametrov ustvari in izvede poizvedbo, nato pa izračuna vse statistične podatke rezultata (angl. result set) poizvedbe.

fillFilterList(req, keyword)

Vrne: String [v »Ajax Request«]

Ključne metode: /

Opis delovanja: Metoda, ki omogoča polnjenje filtrirnih podatkov seznamov na spletni strani.

C1 Line SQL(dataTable, counter, axisX, groupBy, sizeX, sizeY, grid, lgnd)

Vrne: String

Ključne metode: Get_ticks

Opis delovanja: Metoda na podlagi podane tabele podatkov (dataTable) in argumentov ustvari sliko grafikona (črte), jo shrani v datoteko in vrne njegovo ime.

C2 Area SQL(dataTable, counter, axisX, groupBy, sizeX, sizeY, grid, lgnd)

Vrne: String

Ključne metode: calculatePercentage, Get_ticks

Opis delovanja: Metoda na podlagi podane tabele podatkov (dataTable) in argumentov ustvari sliko grafikona (črte po odstotkovni vrednosti), jo shrani v datoteko in vrne njeno ime.

C3 Pie SQL(dataTable, counter, groupBy, ignoreUnder, sizeX, sizeY, lgnd)

Vrne: String

Ključne metode: Get_ticks

Opis delovanja: Metoda na podlagi podane tabele podatkov (dataTable) in argumentov ustvari sliko grafikona (pita), jo shrani v datoteko in vrne njeno ime.

C4 Bubble SQL(dataTable, counter, axisX, axisY, dotSize, sizeX, sizeY, grid)

Vrne: String

Ključne metode: Get_ticks

Opis delovanja: Metoda na podlagi podane tabele podatkov (dataTable) in argumentov ustvari sliko grafikona (mehurčki), jo shrani v datoteko in vrne njeno ime.

C5 Column SQL(dataTable, counter, axisX, groupBy, sizeX, sizeY, grid, lgnd, lbls = 'true')

Vrne: String

Ključne metode: Get_ticks

Opis delovanja: Metoda na podlagi podane tabele podatkov (dataTable) in argumentov ustvari sliko grafikona (stolpci), jo shrani v datoteko in vrne njeno ime.

C6 ColumnStacked SQL(dataTable, counter, axisX, groupBy, sizeX, sizeY, grid, lgnd)

Vrne: String

Ključne metode: Get_ticks

Opis delovanja: Metoda na podlagi podane tabele podatkov (dataTable) in argumentov ustvari sliko grafikona (stolpci po procentualnih vrednostih), jo shrani v datoteko in vrne njeno ime.

C7 Map SQL(dataTable, counter, map, sizeX, sizeY)

Vrne: String

Ključne metode: trimListMap

Opis delovanja: Metoda na podlagi podane tabele podatkov (dataTable) in argumentov ustvari sliko grafikona (zemljevid), jo shrani v datoteko in vrne njeno ime.

C8 Table SQL(dataTable, axisX, axisY, counter, percentage)

Vrne: String [v HTML obliki]

Ključne metode: Get_percentageList

Opis delovanja: Metoda na podlagi podanih parametrov ustvari niz v obliki HTML, ki vsebuje tabelo za prikaz na spletni strani.

5.2.4 WebLAT_Scheduler.py

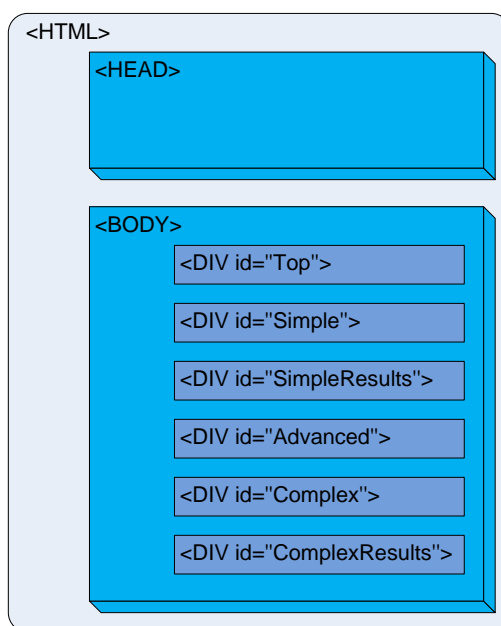
Skripta, dolga tri vrstice, ki zgolj kliče metodo *WebLAT_DB.py/LogFile2DB(datoteka)*, kjer je datoteka generirana glede na datum obliki *accessYYYY-MM-DD.log*

5.3 Sloj uporabniškega vmesnika

Sloj uporabniškega vmesnika je realiziran kot spletna aplikacija v jeziku HTML v obliki dveh spletnih strani. Na strani *WebLAT.html* teče aplikacija, ki je povezana z nekaj datotekami Javascript ter CSS. Na strani *WebLAT_help.html* pa je uporabniški priročnik, podoben poglavju »7 – DELOVANJE APLIKACIJE«.

5.3.1 WebLAT.html

WebLAT.html je glavna datoteka z aplikacijo. Struktura strani je razdeljena hierarhično v skladu z ustreznimi spletnimi standardi. Stran je označena kot `<HTML>`, pod katerim sta `<HEAD>` (glava) ter `<BODY>` (telo) strani. Telo je nadalje razdeljeno na elemente `<DIV>` (sekcija) (Slika 19: Struktura WebLAT.html). Različne sekcije so bile uporabljene zaradi hitrejšega delovanja strani. Če bi za vsak način delovanja aplikacije (»Simple«, »Advanced«, »Complex«) uporabljali ločeno stran, bi bilo vsakič potrebno sprotno nalaganje strani, tako pa se tega problema rešimo, saj lahko vsi načini tečejo na eni strani, uporabniku pa se prikazujejo le izbrani. To dosežemo s skrivanjem/prikazovanjem posameznih sekcij.



Slika 19: Struktura WebLAT.html

Sekcija »Top«

Sekcija »Top« je vedno prikazana (Slika 20), na njej pa lahko uporabnik izbira med načini delovanja aplikacije. Sekcija vsebuje tudi logo aplikacije in gumbe za pomoč, kontakt, izvoz podatkov ter tiskanje.



Slika 20: Sekcija »Top«

Sekcija »Simple«

Sekcija »Simple« (Slika 21) vsebuje preprost filter ter različne možnosti prikaza statističnih podatkov za način delovanja »Simple«. Filter vsebuje najpomembnejše parametre, kot je izbira obdobja zajema podatkov, selekcioniranje na podlagi uspešnosti/neuspešnosti zahtev ter na podlagi agenta, ki je dostopal na strežnik. Izbira prikaza statističnih podatkov obsega splošne statistične podatke ter rangiranje najpogostejših zahtev po različnih parametrih.

Slika 21: Sekcija »Simple«

Sekcija »SimpleResults«

Sekcija se prikaže (Slika 22), ko aplikacija pridobi zahtevane podatke v načinu delovanja »Simple«. Sekcija vsebuje izbrane statistične podatke, glede na izbran filter.

Filter used for currently shown statistics: Date between 2009-03-10 00:00:00 and 2009-03-17 23:59:59 Ignore search bots: true Ignore unknown useragents: true Successful requests: true Failed requests: true			
Days with visits:	6		
Unique visitors (IP):	3014		
# of requests:	103313		
# of requests per day:	17218.83		
Data transfered:	10.304 GB		
Data transfered per request:	99.739 KB		
Data transfered per day:	1.717 GB		
Total time for requests:	1925day(s) 12hr		
Time per request:	26min 50sec 316msec		
Time per day:	320day(s) 22hr		
# of successful requests:	94518		
# of successful requests per day:	15753.00		
Most req. from country:	1: SLOVENIA (61.33%)	2: UNITED STATES (9.66%)	3: UNITED KINGDOM (2.83%)
Most req. from browser:	1: Firefox 3 (73.94%)	2: Internet Explorer 6 (13.85%)	3: Internet Explorer 7 (7.02%)

Slika 22: Primer prikaza sekcije "SimpleResults"

Sekcija »Advanced«

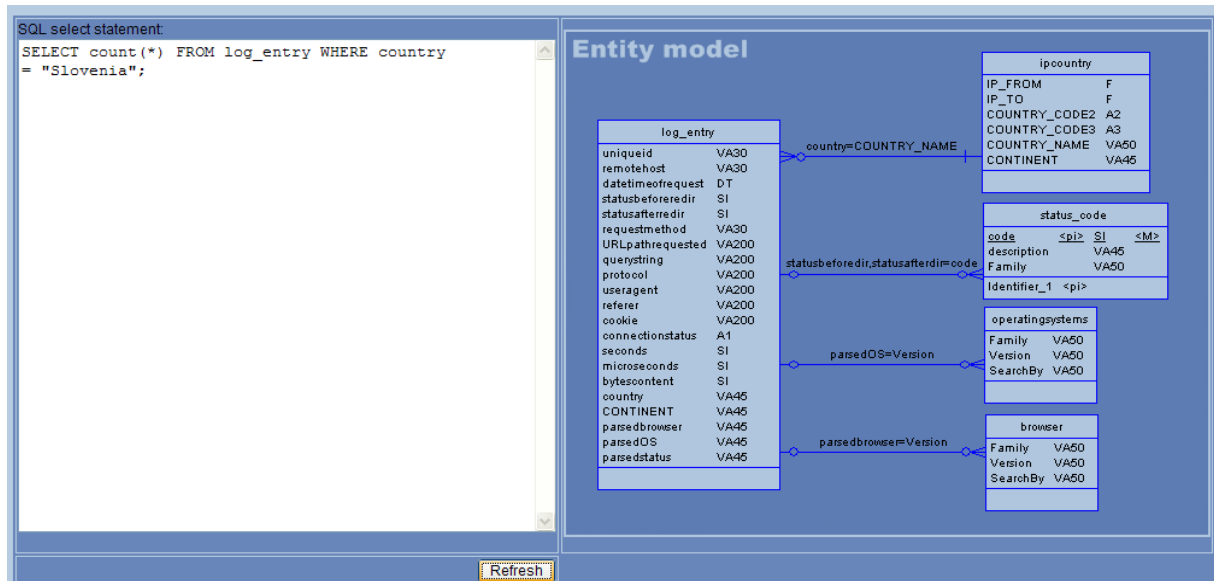
Sekcija v načinu delovanja »Advanced«. Sekcija je sestavljena iz treh logičnih delov (Slika 23: Skica "Advanced" sekcije). Zgornji del omogoča izbiro grafikona, levi del je sestavljen iz polj in menijev, v katerih določimo obliko in parametre grafikona ter filter podatkov, v osrednjem delu sekcije pa se izrisuje izbrani grafikon ali matrika.



Slika 23: Skica "Advanced" sekcije

Sekcija »Complex«

Sekcija v načinu delovanja »Complex« (Slika 24: Prikaz sekcije "Complex") vsebuje zelo preprost uporabniški vmesnik, ki omogoča vpisovanje SQL poizvedb. Za lažjo uporabo je prikazan entitetni model podatkovne baze.



Slika 24: Prikaz sekcije "Complex"

Sekcija »ComplexResults«

Sekcija se prikaže, ko aplikacija na podatkovni bazi izvrši zahtevano poizvedbo in prikaže podatke v tabeli.

5.3.2 WebLAT_js_funkcije.js

WebLAT_js_funkcije.js vsebuje logično podporo spletnemu delu aplikacije. Funkcije so napisane v jeziku Javascript.

getXmlHttpRequestObject()

Vrne: XMLHttpRequest

Opis: Metoda vrne objekt XMLHttpRequest, ki je potreben za delo v načinu Ajax.

ComplexResults()

Vrne: /

Opis: Metoda preko AJAX-a (objekt *complexReq*) kliče Pythonovsko skripto na strežniku (*WebLAT.py/Call_Complex*) s podanim nizom stavka poizvedbe.

handleComplexResults()

Vrne: /

Opis: Metoda preverja, ali je XMLHttpRequest objekt (*complexReq*) z rezultati že pripravljen. Če je pripravljen, ga zapiše na spletno stran z ukazom:

```
document.getElementById('complexresults').innerHTML = complexReq.responseText;
```

SimpleResults()

Vrne: /

Opis: Metoda preko AJAX-a (objekt *simpleReq*) kliče Pythonovsko skripto na strežniku (*WebLAT.py/Call_Simple*) z vsemi podanimi parametri filtriranja (*date_from* – datum od, *date_to* – datum do, *ignoresearchbots* – ignoriraj dostope iskalnikov, *ignoreunknownuseragent* – ignoriraj nepoznane dostope, *successful requests* – uspele zahteve, *failed requests* – neuspele zahteve) v načinu »Simple« delovanja.

handleSimpleResults()

Vrne: /

Opis: Metoda preverja, ali je objekt XMLHttpRequest (*simpleReq*) z rezultati že pripravljen. Če je pripravljen, ga zapiše na spletno stran z ukazom:

```
document.getElementById('simpleresults').innerHTML = simpleReq.responseText;
```

AdvancedResults()

Vrne: /

Opis: Metoda preko AJAX-a (objekt *advancedReq*) kliče Pythonovsko skripto na strežniku (*WebLAT.py/Call_Advanced*) z vsemi podanimi parametri načina delovanja »Advanced«.

Tabela 5: Parametri funkcije Call_Advanced

Parameter	Tip parametra	Opis
date_from	Filter podatkov	Datum »od«
date_to	Filter podatkov	Datum »do«
size_from	Filter podatkov	Količina prenesenih podatkov [B] »od«
size_to	Filter podatkov	Količina prenesenih podatkov [B] »do«
request_from	Filter podatkov	Čas prenosa podatkov [ms] »od«
request_to	Filter podatkov	Čas prenosa podatkov [ms] »do«
URL	Filter podatkov	Naslov zahteve (datoteke na naslovu)
Referer	Filter podatkov	Naslov, s katerega je uporabnik prišel na stran
IP	Filter podatkov	IP (Internet Protocol number)
ignoresearchbots	Filter podatkov	Ignoriraj dostope brskalnikov
ignoreunknownuseragent	Filter podatkov	Ignoriraj nepoznane dostope
CountryList	Filter podatkov	Seznam držav (če je prazen se filter ignorira)
ContinentList	Filter podatkov	Seznam kontinentov
BrowserList	Filter podatkov	Seznam brskalnikov

OSList	Filter podatkov	Seznam operacijskih sistemov
ProtocolList	Filter podatkov	Seznam protokolov
RequestMethodList	Filter podatkov	Seznam metode zahteve
ConnectionStatusList	Filter podatkov	Seznam statusov povezave
StatusCodeList	Filter podatkov	Seznam kod statusov protokola
chartType	Izbran graf	Izbran graf (Line, Area, Pie, Bubble...)
par1	Oblika grafa	Parametri izbranega grafa (se razlikujejo glede na izbran graf).
par2	Oblika grafa	
par3	Oblika grafa	
par4	Oblika grafa	

handleAdvancedResults()

Vrne: /

Opis: Metoda preverja, ali je objekt XMLHttpRequest (*advancedReq*) z rezultati že pripravljen. Če je pripravljen, ga zapiše na spletno stran z ukazom:

```
document.getElementById('advancedChart').innerHTML = advancedReq.responseText;
```

fillListMethod()

Vrne: /

Opis: Metoda preko AJAX-a kliče Pythonovsko skripto na strežniku (*WebLAT.py/fillFilterList*) hkrati za vse objekte seznama. Tako se napolnijo možne vrednosti filtra v načinu »Advanced« za Country (*država*), Continent (*kontinent*), Browser (*brskalnik*), Operating System (*operacijski sistem*), Protocol (*protokol*), Request method (*tip klica*), Connection status (*status povezave*) in Protocol status code (*koda statusa protokola*).

handleFillList()

Vrne: /

Opis: Metoda preverja, ali je objekt XMLHttpRequest za katerikoli parameter filtra iz metode *fillListMethod()* v stanju 4 (*readystatus = 4*), kar pomeni, da ima pripravljene podatke. Če so podatki pripravljeni, z njimi napolni seznam pripadajočega parametra filtra.

deleteFromSelectedList(elementID)

Vrne: /

Opis: Metoda izbriše trenutno izbrane elemente v seznamu (podan kot argument metode) spletne strani.

insertToSelectedList(elementID_Selected, elementID_Available)

Vrne: /

Opis: Metoda doda iz seznama (*elementID_Available*) izbrane (selektirane) elemente v sosednji seznam (*elementID_Selected*). V objektu *elementID_Available* elementi ostanejo (se ne brišejo).

showChartShape(x)

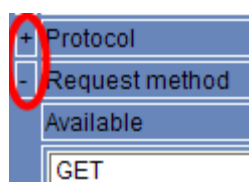
Vrne: /

Opis: Metoda glede na izbrano vrsto grafikona v načinu »Advanced«, določi vrednosti v »drop down« elementih pri izbiri oblike grafa (»Chart shape«). Izbranih je lahko osem različnih grafikonov, zato so možnosti oblikovanja pri vsakem malo drugačne.

showTBody(tbodyid,ClickIcon)

Vrne: /

Opis: Glede na element *tbodyid* in njegovo trenutno vrednost ob kliku nanj spreminja vrednosti + in – (Slika 25: Prikaz delovanja metode showTBody).



Slika 25: Prikaz delovanja metode showTBody

colorframe(x, what)

Vrne: /

Opis: Uporablja se za barvanje podlag pri izbiri načina delovanja (»Simple«, »Advanced«, »Complex«) ali pri izbiri grafa v načinu »Advanced«.

validate(field)

Vrne: /

Opis: Metoda za preverjanje vseh polj spletne strani (preverja, če so v numeričnih poljih res numerični znaki, itd.).

showDiv(listDivShow)

Vrne: /

Opis: Metoda za prikazovanje in skrivanje »Div« sekcij. Prirejena je za uporabo v brskalnikih Internet Explorer, Mozilla Firefox in Opera.

refreshPageOnEnter(e)

Vrne: /

Opis: Metoda preverja, ali je bila pritisnjena tipka »Enter«. Če je bila, začne z izrisovanjem grafa ali matrike.

convertCSVJPG()

Vrne: /

Opis: Metoda deluje glede na trenutno prikazano vrednost (matrika/grafikon). Matriko konvertira v CSV (»comma separated values«) tabelo in jo shrani. Grafikon shrani v datoteko formata PNG.

convertPDF()

Vrne: /

Opis: Metoda preko AJAX-a (objekt *PDFReq*) kliče Pythonovsko skripto na strežniku (ustrezno metodo, glede na način delovanja, v katerem kličemo izvoz podatkov v datoteko PDF) s podanim nizom stavka poizvedbe.

handlePDFResults()

Vrne: /

Opis: Metoda preverja, ali je XMLHttpRequest objekt (*PDFReq*) z rezultati že pripravljen. Če je pripravljen, se izvrši ukaz, ki odpre novo okno z odprto datoteko PDF.

ProgressStart(type)

Vrne: /

Opis: Metoda za izrisovanje »progress bar« vrstice.

ProgressStop(type)

Vrne: /

Opis: Metoda ustavi izrisovanje »progress bar«-a.

returnListString(eId)

Vrne: String

Opis: Metoda vrne niz, sestavljen iz izbranih elementov seznama (naziv katerega je podan kot argument metode), ločenih z vejico.

SetDates(type, date)

Vrne: [Nastavi datumska polja v delovanju »Simple« in »Advanced«]

Opis: Metoda glede na argumenta metode nastavi datumska polja.

daysInMonth(iMonth, iYear)

Vrne: int

Opis: Metoda vrne število dni podanega meseca in leta.

twoDig(number)

Vrne: String

Opis: Če je podan argument enomestna številka, ji doda ničlo, da postane dvomestna (npr. 4 -> 04).

CancelQuery(type)

Vrne: /

Opis: Metoda prekine izvajanje (čakanje) izrisa grafikona ali matrike.

isDate(strDate)

Vrne: Boolean

Opis: Metoda preverja, ali je podani niz v pravilni datumski obliki.

createDate(strDate)

Vrne: Date

Opis: Metoda niz pretvori v objekt tipa Date.

onlyNumbers(e)

Vrne: Boolean

Opis: Metoda vrne vrednost [true], če je podan argument cifra ali katera od tipk (enter, del, backspace, space, end, left, right), sicer vrne vrednost [false].

5.3.3 WebLAT_css.css

Datoteka določa obliko spletne strani.

5.3.4 WebLAT_help.html

Spletna stran vsebuje podrobna navodila za uporabo aplikacije.

5.3.5 Calendar.css

Datoteka opisuje obliko, ki se uporabi v poljih za izbiro datuma na strani *WebLAT.html*.

Opomba: CSS je del »*The DHTML Calendar, version 1.0 "It is happening again"*«, ki je distribuiran kot odprtokodni program (GNU Lesser General Public License - Copyright Mihai Bazon).

5.3.6 Calendar.js

Datoteka nudi logično podporo vsem poljem za izbiro datuma na strani *WebLAT.html*.

Opomba: Javascript je del »*The DHTML Calendar, version 1.0 "It is happening again"*«, ki je distribuiran kot odprtokodni program (GNU Lesser General Public License - Copyright Mihai Bazon).

6 NAMESTITEV SISTEMA

6.1 Potrebna programska oprema

Pred namestitvijo aplikacije je potrebno imeti nameščeno naslednjo programsko opremo. Vsa navedena programska oprema je brezplačna in jo je mogoče dobiti na spletu.

- Relacijska podatkovna baza MySQL
 - Navodila: <http://dev.mysql.com/doc/>
 - Verzija: 5.0 ali novejša
 - Spletna stran: <http://dev.mysql.com/downloads/>

- Strežnik Apache
 - Navodila: <http://httpd.apache.org/docs/>
 - Verzija: 2.2.10 ali novejša
 - Spletna stran: <http://httpd.apache.org/>

- Cronolog
 - Navodila: <http://cronolog.org/usage.html>
 - Verzija: 1.6.1
 - Spletna stran: <http://cronolog.org/download/index.html>

- Python
 - Navodila: <http://www.python.org/doc/>
 - Verzija: 2.5.X
 - Spletna stran: <http://www.python.org/download/>

Moduli

Module je potrebno namestiti v mapo `C:[Python mapa]\Lib\site-packages`

- Modul Numpy
 - Navodila: <http://numpy.scipy.org/>
 - Verzija: 1.2.1
 - Spletna stran: <http://numpy.scipy.org/#docs>

- Modul mod_python
 - *Navodila:* <http://www.modpython.org/live/current/doc-html/>
 - *Verzija:* 3.3.1 ali novejša
 - *Spletna stran:* <http://www.modpython.org/>

- Modul PIL – Python Imaging Library
 - *Navodila:* <http://www.pythonware.com/products/pil/support.htm>
 - *Verzija:* PIL-1.1.6.win32-py2.5
 - *Spletna stran:* <http://www.pythonware.com/products/pil/>

- Modul MySQLdb
 - *Navodila:* <http://mysql-python.sourceforge.net/MySQLdb.html>
 - *Verzija:* MySQL-python-1.2.2.win32-py2.5.exe ali novejši
 - *Spletna stran:* <http://sourceforge.net/projects/mysql-python>

- Modul Basemap
 - *Navodila:* <http://matplotlib.sourceforge.net/basemap/doc/html/>
 - *Verzija:* basemap-0.9.5.win32-py2.5.exe
 - *Spletna stran:* http://sourceforge.net/project/showfiles.php?group_id=80706

- Modul Matplotlib
 - *Navodila:* <http://matplotlib.sourceforge.net/users/installing.html>
 - *Verzija:* matplotlib-0.98.5
 - *Spletna stran:* http://sourceforge.net/project/showfiles.php?group_id=80706

- Modul ReportLab
 - *Navodila:* http://www.reportlab.org/os_documentation.html
 - *Verzija:* ReportLab 2.3
 - *Spletna stran:* <http://www.reportlab.org/downloads.html>

6.2 Konfiguracija strežnika Apache

V mapi, kjer je nameščen strežnik Apache, poiščemo datoteko *conf/httpd.conf*, jo odpremo in znotraj nje nastavimo naslednje nastavitve.

Definicija dnevniške datoteke

Ker deluje aplikacija na podlagi prej definirane dnevniške datoteke, je potrebno v konfiguracijo strežnika Apache dodati definicijo beleženja te datoteke. Znotraj *<IfModule log_config_module>* prepisemo spodnjo kodo, ki definira novo dnevniško datoteko in njeno vsebino:

```
LogFormat "%{UNIQUE_ID}e|t %h|t {%Y-%m-%d %T}t|t %s|t %>s|t %m|t %U|t %q|t %H|t %{User-Agent}i|t %{Referer}i|t %{Cookie}i|t %X|t %T|t %D|t %B|t" WebLAT
CustomLog "|c:/Program\ Files\Apache\ Software\ Foundation\Apache2.2\htdocs\WebLAT\cronolog\cronolog.exe c:/Program\ Files\Apache\ Software\ Foundation\Apache2.2\htdocs\WebLAT\cronolog\access%Y-%m-%d.log" WebLAT
```

Določitev »handlerja« strežnika

Da bo WebLAT uporabljal modul *mod_python*, je potrebno v konfiguracijo strežnika dodati definicijo »handlerja«.

```
<Directory "C://[Apache mapa]/htdocs/WebLAT">
  AddHandler mod_python .py
  PythonHandler mod_python.publisher
  PythonDebug On
</Directory>
```

Konfiguracija »mod_python« modula

Da bo strežnik naložil *mod_python* modul, dosežemo z naslednjo vrstico v konfiguraciji:

```
LoadModule python_module modules/mod_python.so
```

6.3 Postavitev fizične podatkovne baze

Za postavitev fizične podatkovne baze moramo skripte SQL, ki so priložene namestitvi, pognati v naslednjem vrstnem redu:

- 1) Kreiranje fizične podatkovne baze ter tabel
 - *CREATE DB TABLES.sql*
- 2) Napolnitev šifrantov
 - *INSERT sifrant statuscode.sql*

- *INSERT_sifrant_operatingsystems.sql*
- *INSERT_sifrant_ipcountry.sql*
- *INSERT_sifrant_browsers.sql*

6.4 Namestitev skript in datotek

Mapo **WebLAT**, ki je priložena, je potrebno prenesti v mapo:

C:\[Apachejeva mapa]\htdocs

Tabela 6: Seznam in opis datotek in map

Datoteka	Tip
<u><i>WebLAT.py</i></u>	Python skripta
<u><i>WebLAT_DB.py</i></u>	Python skripta
<u><i>WebLAT_Util.py</i></u>	Python skripta
<u><i>WebLAT_Scheduler.py</i></u>	Python skripta
<u><i>WebLAT.html</i></u>	HTML datoteka
<u><i>WebLAT_help.html</i></u>	HTML datoteka
<u><i>calendar.js</i></u>	Javascript datoteka
<u><i>calendar.css</i></u>	CSS datoteka
<u><i>WebLAT_js_funkcije.js</i></u>	Javascript datoteka
<u><i>WebLAT_css.css</i></u>	CSS datoteka
<u><i>Filters</i></u>	Konfiguracijska datoteka
<u><i>Dbconn</i></u>	Konfiguracijska datoteka
<u><i>world_adm0.dbf</i></u>	Datoteke potrebne za risanje zemljevidov
<u><i>world_adm0.shp</i></u>	
<u><i>world_adm0.shx</i></u>	
images	Mapa z vsemi potrebnimi slikami
images/charts	Mapa, kamor se bodo shranjevali izrisani grafi
images/help	Mapa slik, ki se prikazujejo na strani pomoči
PDFs	Mapa, kamor se shranjujejo datoteke PDF
cronolog	Mapa, kamor se shranjujejo dnevniške datoteke
cronolog/cronolog.exe	Program Cronolog
cronolog/ <u><i>Cronolog_readme</i></u>	Pomoč za Cronolog

Konfiguracija datoteke »filters«

»Filters« je konfiguracijska datoteka, v kateri nastavimo, katere zahteve želimo oz. ne želimo vpisovati v podatkovno bazo. Osnovni zapis, ki vnaša v bazo vse zahteve:

```
# This is comment (each line must start with ##)
# This file defines which URLpaths go into database or which do not
# If first line is <!I then all regular expressions below will go IN database
# If first line is <!O then all regular expressions below will be left OUT of database
# More about regular expressions at : http://www.regular-expressions.info/
<!I
```

Razlaga datoteke:

- Vsaka vrstica, ki se začne z '#' je komentar.
- Prva vrstica mora biti '<!O' ali '<!I'.
 - <!O : Podani regularni izrazi bodo izpuščeni iz baze
 - <!I: Samo podani regularni izrazi bodo vpisan v bazo.
- Prvi vrstici sledijo regularni izrazi.

Konfiguracija dostopa do podatkovne zbirke

Datoteka »Dbcoon« vsebuje podatke za dostop do podatkovne zbirke MySQL. Podati je potrebno naslednje argumente:

- host – gostitelj
- user – uporabnik
- passwd – geslo uporabnika
- db – podatkovna baza

Oblika konfiguracijske datoteke:

```
# This is comment
# DBconn file consists of four parameters (host, user, passwd, db)
host = [localhost]
user = [root]
passwd = [kekecpasteta]
db = [apache_logs]
```

Nastavitev dnevnega vpisa – »task«

Program Cronolog ustvari ločeno dnevniško datoteko za vsak dan. Vpis te datoteke v podatkovno bazo se vrši preko skripte v Pythonu, ki se zažene vsakodnevno preko upravitelja opravil (angl. task scheduler). Tega moramo ročno vnesti po naslednjem postopku:

- Start – Settings – Control Panel – Scheduled Tasks – Add Scheduled Task
- Nastavitve:
 - Run: "C:\[imenik s Pythonom]\python.exe" WebLAT_Scheduler.py
 - Start in: "C:\[imenik z Apachejem]\htdocs\WebLAT"
 - Schedule Task: *Every Day*

7 DELOVANJE APLIKACIJE

Aplikacija WebLAT omogoča analizo dostopov do spletnega strežnika. V podatkovni bazi so zabeleženi vsi dostopi do baze, podatki pa se osvežujejo enkrat dnevno, tako da imamo vedno na voljo vso zgodovino dostopov, razen tekočega dneva. Aplikacija omogoča pregledovanje različnih grafikonov, matrik in lastno vpisovanje poizvedb (SQL).

Delovanje aplikacije lahko razdelimo v tri logične sklope:

- **»Simple« (Preprosto)** Poudarek na najpomembnejših statističnih podatkih.
- **»Advanced« (Napredno)** Prikaz grafikona ali matrike z veliko možnostmi izbire in filtriranja.
- **»Complex« (Kompleksno)** Prikaz tabele, po vpisu poizvedbe SQL.

7.1 Orodna vrstica



Slika 26: Orodna vrstica

Razlaga označb (Slika 26: Orodna vrstica):

- 1) Izbira načina delovanja »Simple«
- 2) Izbira načina delovanja »Advanced«
- 3) Izbira načina delovanja »Complex«
- 4) Gumb za prikaz okna pomoči
- 5) Gumb za pošiljanje elektronske pošte oz. kontakt
- 6) Gumb za tiskanje izbranega prikaza
- 7) Gumb za izvoz podatkov v datotečnem formatu JPG (za grafikone) ali besedilnem formatu TXT, kjer so podatki ločeni s tabularojem (matrike).
- 8) Gumb za izvoz podatkov v formatu PDF.

7.2 Način »Simple«

Ta način nam omogoča prikazovanje najpomembnejših statističnih podatkov.

The screenshot shows a web analytics dashboard with the following elements:

- Filter:** Date from 2009-03-10 00:00:00 to 2009-03-17 23:59:59. Includes checkboxes for 'Ignore search bots', 'Ignore unknown useragents', 'Successful requests', and 'Failed requests'.
- Show the following statistics:** Checkboxes for 'General statistics', 'Successful requests statistics', and 'Failed requests statistics'.
- Show ranking statistics:** Checkboxes for 'Country', 'Continent', 'Browser', 'OS', 'Protocol', 'Request method', 'Connection Status', and 'Status'.
- Show statistics** button.
- Filter used for currently shown statistics:** Date between 2009-03-10 00:00:00 and 2009-03-17 23:59:59. ignore search bots: true, ignore unknown useragents: true, Successful requests: true, Failed requests: true.
- Table of statistics:**

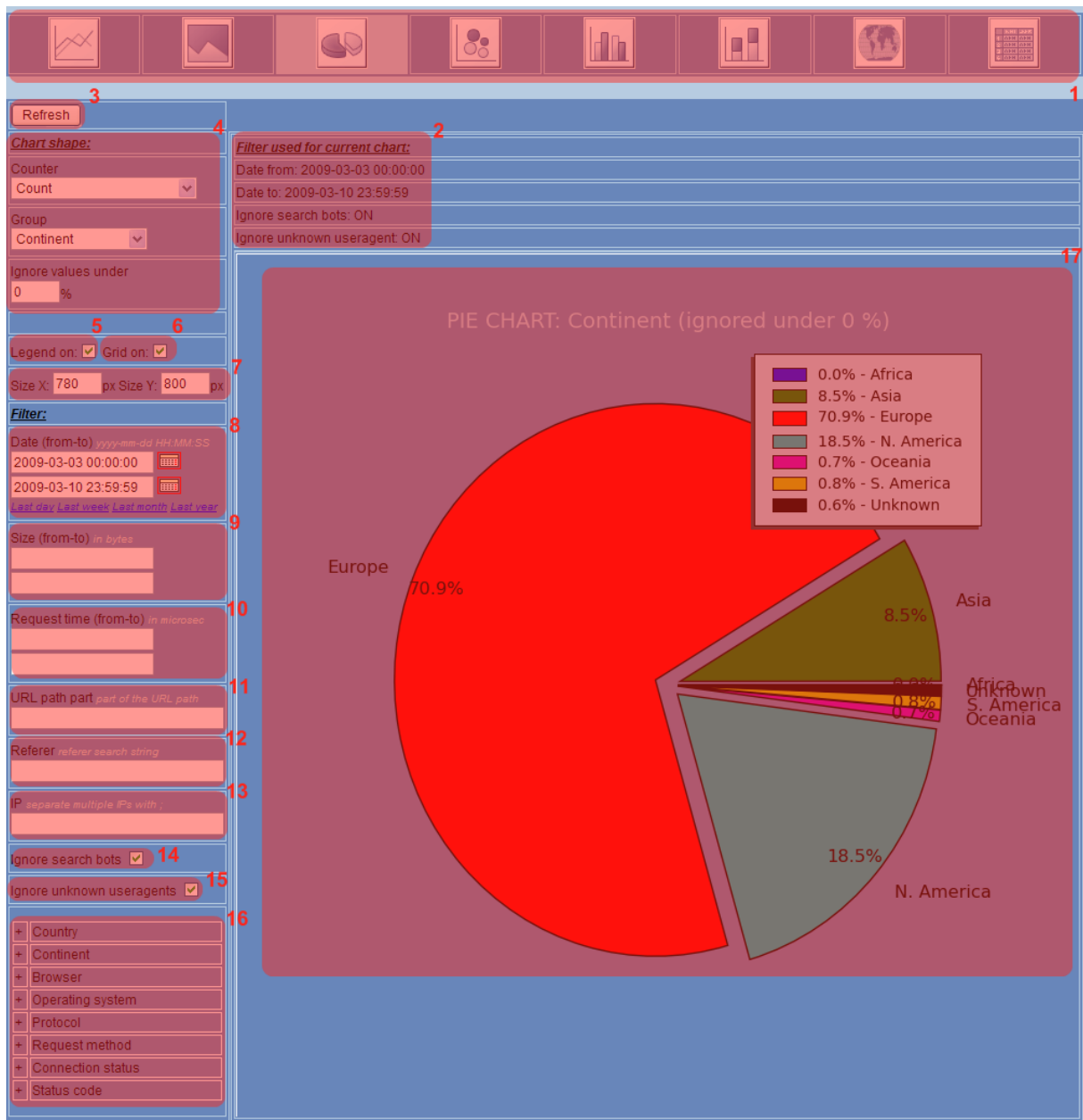
Days with visits:	6
Unique visitors (IP):	3014
# of requests:	103313
# of requests per day:	17218.83
Data transferred:	10.304 GB
Data transferred per request:	99.739 KB
Data transferred per day:	1.717 GB
Total time for requests:	1925day(s) 12hr
Time per request:	28min 50sec 316msec
Time per day:	320day(s) 22hr
# of successful requests:	94518
# of successful requests per day:	15753.00
# of failed requests:	8795
# of failed requests per day:	1465.83
Most req. from country:	1: SLOVENIA (61.33%) 2: UNITED STATES (9.66%) 3: UNITED KINGDOM (2.83%)
Most req. from browser:	1: Firefox 3 (73.94%) 2: Internet Explorer 6 (13.85%) 3: Internet Explorer 7 (7.02%)

Slika 27: Način "Simple"

Razlaga označb (Slika 27: Način "Simple"):

- 1) Izbira obdobja (datumska polja)
- 2) Ignoriranje iskalnikov
- 3) Ignoriranje nepoznanih dostopov (nepoznani iskalniki ali drugi spletni dostopi)
- 4) Izbira uspešnih zahtev
- 5) Izbira neuspešnih zahtev
- 6) Prikaz splošnih statističnih podatkov
- 7) Prikaz statističnih podatkov o uspešnih zahtevah
- 8) Prikaz statističnih podatkov o neuspešnih zahtevah
- 9) Izbira prikaza statističnih podatkov o pogostosti uporabe določenega parametra
- 10) Gumb za osvežitev izbire
- 11) Zapis filtra zadnje izbire
- 12) Tabela vseh izbranih statističnih podatkov

7.3 Način »Advanced«

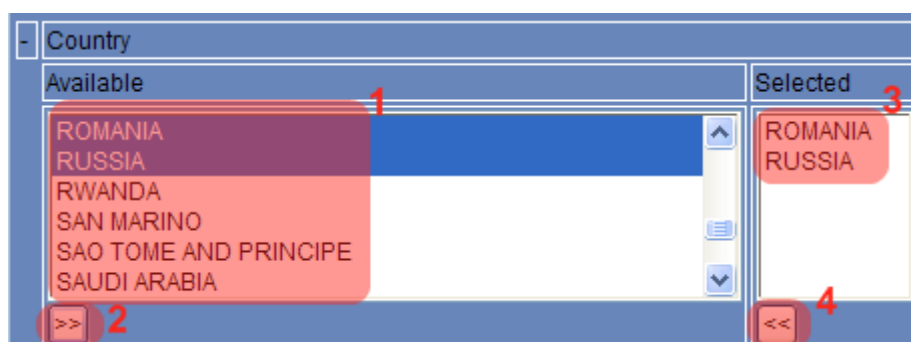


Slika 28: Način "Advanced"

Razlaga označb (Slika 28: Način "Advanced"):

- 1) Izbira vrste grafikona ali matrike
- 2) Prikaz filtra, uporabljenega pri prikazanem grafikonu ali matriki
- 3) Gumb za osveževanje grafikona ali matrike
- 4) Izbira podatkov na podatkovnih oseh (se razlikuje glede na tip grafikona!)
- 5) Prikaz legende

- 6) Prikaz mreže
- 7) Želena velikost grafa po X in Y osi
- 8) FILTER: Določitev datumskega obdobja
- 9) FILTER: Omejitev količine podatkov posamezne zahteve (v bajtih)
- 10) FILTER: Omejitev časa izvajanja zahteve (v milisekundah)
- 11) FILTER: URL pot, kamor kaže zahteva (lahko tudi le del poti)
- 12) FILTER: URL pot, od kjer je zahteva prišla
- 13) FILTER: IP naslov (lahko jih je več, ločemi z znakom ;)
- 14) FILTER: Ignoriranje iskalnikov
- 15) FILTER: Ignoriranje nepoznanih dostopov (nepoznani iskalniki ali drugi spletni dostopi)
- 16) FILTER: Filtriranje podatkovnih skupin (država, kontinent, brskalnik, operacijski sistem, protokol, metoda zahteve, status povezave, koda povezave zahteve) – za delovanje glej: Slika 29: Prikaz filtriranja podatkovne skupine
- 17) Prikaz grafikona ali matrike

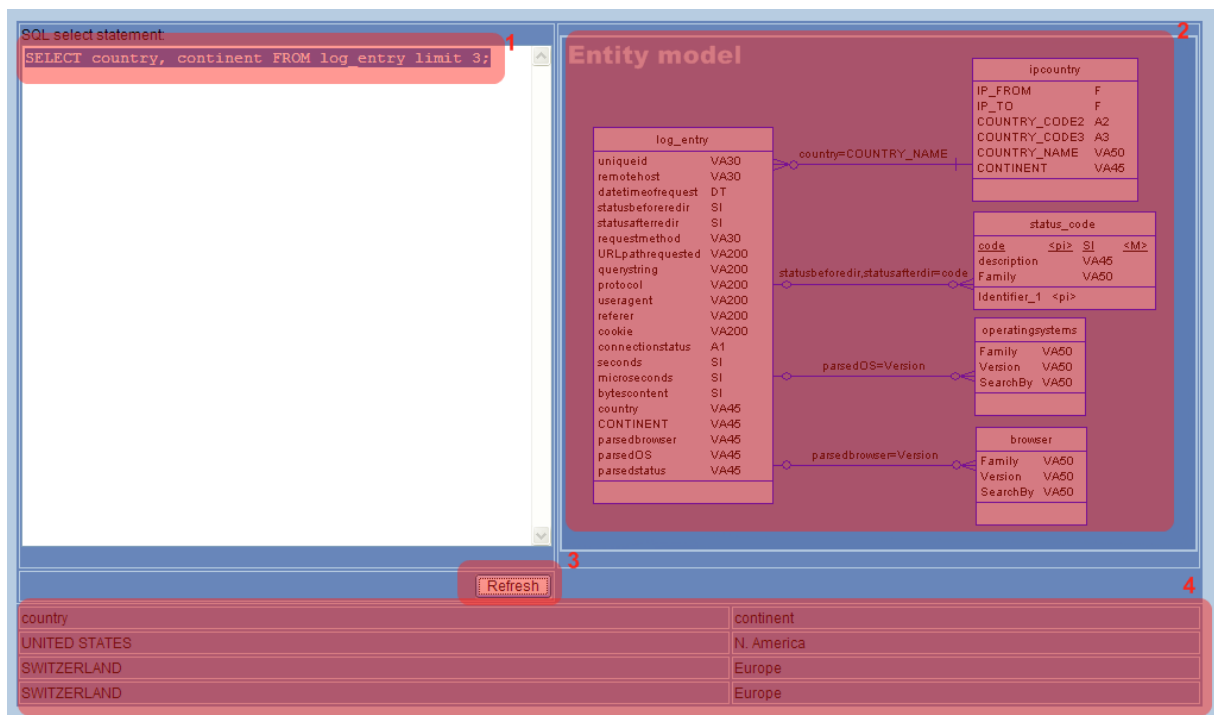


Slika 29: Prikaz filtriranja podatkovne skupine

Razlaga označb (Slika 29: Prikaz filtriranja podatkovne skupine):

- 1) Seznam možnih podatkov za izbor
- 2) Gumb za dodajanje podatkov v izbran seznam podatkov, ki jih hočemo filtrirati
- 3) Izbrani podatki (podatki v seznamu »Selected« bodo med selekcijo, če pa je seznam »Selected« prazen, se ta filter ne upošteva)
- 4) Gumb za brisanje podatkov iz seznama izbranih

7.4 Način »Complex«



Slika 30: Način "Complex"

Razlaga označb (Slika 30: Način "Complex"):

- 1) Polje za vnos poizvedbe
- 2) Entitetni model, za pomoč pri sestavljanju poizvedbe
- 3) Gumb za izvedbo poizvedbe
- 4) Prikaz poizvedbe

7.5 Izvoz podatkov

Izvoz podatkov je možen na tri načine. Tipka CSV/JPG omogoča dva načina izvoza, in sicer izvoz datoteke formata JPG, v primeru, da imamo trenutno prikazan grafikon, ali v besedilni datoteki formata TXT, v primeru, da imamo trenutno prikazano katerokoli matriko. Gumb PDF ponuja tretjo možnost izvoza podatkov, torej v datoteki PDF, ki vsebuje izbran grafikon ali matriko ter filter na podatkih, ki je bil uporabljen.

7.6 Pomoč, kontakt in tiskanje

Na orodni vrstici so povezave na pomoč (»help«), ki odpre novo okno z opisom pomoči, kontakt (»contact«), ki odpre urejevalnik spletne pošte ter tiskanje (»print«).

8 ZAKLJUČEK

Rezultata diplomske naloge sta predvsem dva.

Prvi del diplomske naloge ponuja strnjen pregled preteklega razvoja in uporabe spletne analize v današnjem času, vključno z medsebojno primerjavo posameznih že obstoječih sistemov glede na njihove temeljne značilnosti ter prednosti in slabosti vsakega izmed njih. Ta del morebitnemu uporabniku olajšuje izbiro ustreznega sistema glede na njegove potrebe in želje v posamezni poslovni ali drugačni situaciji.

Drugi, temeljni rezultat diplomske naloge pa je izdelan lasten sistem za spletno analizo, ki ponuja velik nabor možnosti pregledovanja podatkov (grafikoni in matrike) in analiziranja (statistični podatki) dostopov do strežnika. Samo v načinu delovanja »Advanced« je možno izbirati med sedmimi osnovnimi različicami grafikonov (črtasti, stolpični, zemljevid sveta ali vseh kontinentov, »torta« graf, itd.) ali matriko, vsak od njih pa ponuja izris vseh parametrov (časovni po dnevih, tednih, mesecih letih ter npr. operacijski sistem, brskalnik, geološki podatki, protokol, status, itd.), ki so zajeti v spletnem dnevniku. Kratek izračun prinese kar 2235 različnih grafikonov in 588 različnih matrik.

Aplikacija bo v uporabi v Laboratoriju za umetno inteligenco (LUI) na Fakulteti za računalništvo in informatiko Univerze v Ljubljani.

Med samo izdelavo diplomske naloge sem se soočal predvsem z dvema temeljnima izzivoma. Zaradi velikega števila podatkov, ki jih spletni dnevniki zajemajo, je potrebno poiskati načine, s katerimi optimiziramo oziroma pospešimo delovanje same aplikacije. Možnosti optimizacije sem iskal in izkoristil na vseh področjih, od zajema podatkov, tipa zajetih podatkov, na sloju poslovne logike uporabe najhitrejših algoritmov, do optimiziranja spletne strani. Končni rezultat je hitra in stabilna aplikacija.

Naslednji izziv je predstavljala povezljivost vseh uporabljenih tehnologij. Zato sem pred začetkom izdelave samega sistema znanje nadgradil teoretično, kar sem nato uporabil v praksi.

Večnivojska zgradba aplikacije je moderna in v današnjem času najbolj uveljavljena oblika programiranja. Kot taka omogoča enostavno nadgradnjo ali prilagajanje obstoječega sistema, saj omogoča spremembo kateregakoli sloja (uporabniški, poslovni, podatkovni) brez potrebe po spremembi ostalih slojev. Možna je menjava podatkovne baze, sprememba poslovnih pravil ali algoritmov ter sprememba izgleda uporabniškega vmesnika, brez potrebe po spreminjanju ostalih slojev.

Kot pri vseh računalniških programih, sistemih in aplikacijah, tudi pri tej aplikaciji obstoji možnost izboljšav, nadgradnje in prilagoditev. Izpostavil bi predvsem naslednje.

- Optimizacijo na področju hitrosti izrisov grafov. Kljub temu, da sem tej temi posvetil večji del pozornosti, je tu še prostor za izboljšave.
- Selektivno zbiranje podatkov bi prineslo hitrejše delovanje (manj zapisov v bazi) in hkrati boljše statistične rezultate (napake zaradi večkratnih zapisov pri enem dostopu). Selektivno zbiranje je mogoče na dva načina:
 - Odstranjevanje vseh zapisov, kjer je zahteva po določenem tipu datoteke (jpg, png, css, js). Tak način je že realiziran v aplikaciji.
 - Usklajeno zbiranje podatkov s spletnimi stranmi, ki so na strežniku. Za tak način je potrebno v programu (ali podatkovni bazi) natančno definirati vse spletne strani na strežniku – ustvariti imenike in nato na podlagi teh imenikov filtrirati zapise iz dnevnika v bazo.
- Izpopolnitev uporabniškega vmesnika.
- Morebitna poenostavitev metod in funkcij v programskih jezikih Python in Javascript.
- Dodajanje uporabniku prijaznih funkcij, ki so že del ponudbe podobnih aplikacij (npr. redno (mesečno, tedensko) pošiljanje spletne pošte, izdelava prijaznejšega vmesnika za namestitev sistema, ipd.).

9 LITERATURA

Knjige, diplomsko delo:

- [1] A. Kaushik, *Web Analytics: An hour a day*, Indianapolis: Wiley Publishing, 2007
- [2] M. Lutz, D. Ascher, *Learning Python*, USA: O'Reilly & Associates, Inc., 1999
- [3] B. Masand, M. Spiliopoulou, *Web Usage Analysis and user profiling*, Springer, 2000
- [4] P. Sostre, J. LeClaire, *Web Analytics for dummies*, Indianapolis: Wiley Publishing, 2007
- [5] B. Tittl, *Diplomsko delo: Primerjava razvoja spletne aplikacije v tehnologijah PHP in AJAX*, Fakulteta za računalništvo in informatiko, 2009
- [6] C. Ullman, L. Dykes, *Beginning Ajax*, Indianapolis: Wiley Publishing, 2007

Ostali viri:

- [7] AJAX. Dostopno na:
[http://sl.wikipedia.org/wiki/AJAX_\(programiranje\)](http://sl.wikipedia.org/wiki/AJAX_(programiranje))
- [8] CSS. Dostopno na:
<http://sl.wikipedia.org/wiki/CSS>
- [9] Definicija spletne analize. Dostopno na:
<http://www.webanalyticsassociation.org/aboutus/>
- [10] Javascript. Dostopno na:
<http://sl.wikipedia.org/wiki/Javascript>

- [11] mod_python. Dostopno na:
<http://modpython.org/>
- [12] MySQL. Dostopno na.
<http://sl.wikipedia.org/wiki/MySQL>
- [13] Python. Dostopno na:
<http://sl.wikipedia.org/wiki/Python>
- [14] Seznam aplikacij za spletno analizo. Dostopno na:
http://en.wikipedia.org/wiki/List_of_web_analytics_software
- [15] Spletni strežnik Apache. Dostopno na:
http://sl.wikipedia.org/wiki/Spletni_stre%C5%BEnik_Apache
- [16] Tržni delež aplikacij za spletno analizo. Dostopno na:
<http://blog.immeria.net/2008/01/web-analytics-vendors-market-shares.html>
- [17] Web Trends Analytics 8. Dostopno na:
http://www.intelligententerprise.com/print_article.jhtml?articleID=199700517

PRILOGE

9.1 Vsebina priložene zgoščenke

Mapa	Datoteka
/	<i>Diplomsko delo - Aleš Čadež.pdf</i>
/	<i>Povzetek diplomskega dela – Aleš Čadež.pdf</i>
WebLAT	<i>Mapa, kjer se nahajajo datoteke potrebne za delovanje programa.</i>
WebLAT	calendar.css
WebLAT	calendar.js
WebLAT	DBconn
WebLAT	filters
WebLAT	WebLAT.html
WebLAT	WebLAT.py
WebLAT	WebLAT_css.css
WebLAT	WebLAT_DB.py
WebLAT	WebLAT_DB.pyc
WebLAT	WebLAT_help.html
WebLAT	WebLAT_js_funkcije.js
WebLAT	WebLAT_Scheduler.py
WebLAT	WebLAT_Util.py
WebLAT	WebLAT_Util.pyc
WebLAT	world_adm0.dbf
WebLAT	world_adm0.shp
WebLAT	world_adm0.shx
WebLAT/cronolog	<i>Mapa, kjer se nahaja program Cronolog shranjene dnevniške datoteke.</i>
WebLAT/cronolog	cronolog.exe
WebLAT/cronolog	Cronolog_readme
WebLAT/images	<i>Mapa, kjer se nahajajo slike za prikaz na spletni strani.</i>
WebLAT/images	C1_Line.png
WebLAT/images	C2_Area.png
WebLAT/images	C3_Pie.png
WebLAT/images	C4_Bubble.png
WebLAT/images	C5_Column.png
WebLAT/images	C6_ColumnStacked.png
WebLAT/images	C7_Map.png
WebLAT/images	C8_Table.png
WebLAT/images	calendarimg.gif

WebLAT/images	Circle1.png
WebLAT/images	Circle2.png
WebLAT/images	Circle3.png
WebLAT/images	Circle4.png
WebLAT/images	Circle5.png
WebLAT/images	Circle6.png
WebLAT/images	Circle7.png
WebLAT/images	Circle8.png
WebLAT/images	CircleEmpty.png
WebLAT/images	EntityModel.png
WebLAT/images	topFramePicture.png
WebLAT/images/charts	<i>Mapa, kamor se shranjujejo slike prikazanih grafikonov.</i>
WebLAT/images/help	<i>Mapa, kjer se nahajajo slike strani pomoči.</i>
WebLAT/images/help	H1_Toolbar.png
WebLAT/images/help	H2_Simple.png
WebLAT/images/help	H3_Advanced.png
WebLAT/images/help	H4_Complex.png
WebLAT/PDFs	<i>Mapa, kamor se shranjujejo datoteke PDF, ki jih uporabniki izdelajo za izvoz</i>
Skripte SQL	<i>Mapak, kjer se nahajajo datoteke za kreiranje podatkovne baze.</i>
Skripte SQL	CREATE_DB_TABLES.sql
Skripte SQL	INSERT_sifrant_browsers.sql
Skripte SQL	INSERT_sifrant_ipcountry.sql
Skripte SQL	INSERT_sifrant_operatingsystems.sql
Skripte SQL	INSERT_sifrant_statuscode.sql
Podporni programi	<i>Namestitvene dat. za podporne programe, potrebne za delovanje sistema WebLAT.</i>
Podporni programi	apache_2.2.11-win32-x86-openssl-0.9.8i.msi
Podporni programi	basemap-0.99.3.win32-py2.5.exe
Podporni programi	cronolog-1.6.1-win32.zip
Podporni programi	matplotlib-0.98.5.2.win32-py2.5.exe
Podporni programi	mod_python-3.3.1.win32-py2.5-Apache2.2.exe
Podporni programi	mysql-5.1.32-winx64.msi
Podporni programi	mysql-gui-tools-5.0-r17-win32.msi
Podporni programi	MySQL-python-1.2.2.win32-py2.5.exe
Podporni programi	numpy-1.2.1-win32-superpack-python2.5.exe
Podporni programi	PIL-1.1.6.win32-py2.5.exe
Podporni programi	python-2.5.4.msi
Podporni programi	pywin32-213.win32-py2.5.exe
Podporni programi	reportlab-2.3.win32-py2.5.exe

