Fachhochschule
Bonn-Rhein-Sieg

Log Analyse von Web Servern

Ausarbeitung im Rahmen der Veranstaltung

Verteilte und parallele Systeme II

an der Fachhochschule Bonn-Rhein-Sieg
Fachbereich Informatik

Wintersemester 04/05

Autoren: Tanja Hohl und Ralf Debusmann

Inhaltsverzeichnis

1 Einführung

Analysen und Messungen des Webtraffics spielen oft eine wichtige Rolle im Design von Webseiten, in der Verwaltung von Webservern und Proxies und im Betrieb von Netzwerken. Außerdem sind sie oftmals der Anlass für die Entwicklungen neuer Technologien zur Performance-Steigerung.

Die Analyse von Webtraffic wird in 3 Schritten vollzogen:

1. Webtransfer an bestimmter Stelle beobachten
2. Logfiles in bestimmten Format generieren
3. Logfiles bearbeiten um Analyse vorzubereiten

Neben Webserver- und Proxy-Logging, kann man auch im Webclient dem Browser oder direkt auf Paketebene die gesendeten IP-Pakete loggen. Die zuletzt genannten Alternativen zur Messung von Webtraffic halten sich leider nicht an bestimmte Log-Formate.

Obwohl Formate für Logfiles nicht standardisiert sind, halten sich die meisten Implementierungen von Webservern und Proxies an inoffizielle Standards für Logfile-Formate. Zum einen gibt es das CLF (Common Log Format), welches in Kapitel 4.1 näher beschrieben wird und zum anderen existiert das ECLF (Extended Common Log Format), über das wir in Kapitel 4.2 mehr erfahren.

Da die nackten Informationen in Logfiles noch nicht sehr hilfreich sind, müssen die Daten erst bearbeitet werden um eine Analyse zu ermöglichen, aus der dann aussagekräftige Informationen gewonnen werden können.

In Kapitel 6.1 stellen wir Ihnen beispielhaft ein Analyse-Tool namens WebSuxess vor anhand dessen wir die Analyse und die Auswertung der Daten aufzeigen.

2 Motivation

In diesem Kapitel diskutieren wir über die Motivation der Webdatenanalyse aus verschiedenen Sichtweisen. Wir betrachten die Sichtweisen der Webseiten-Entwickler, der Webserver-Betreiber, sowie die Sichtweise der Netzwerk-Betreiber und heben besondere Interessen dieser Gruppen hervor.

2.1 Motivation der Webseiten-Entwickler und Inhaber

Für Webseiten-Entwickler ist es natürlich interessant zu messen wie oft und welche Seiten von Kunden angeklickt bzw. besucht werden. Diese Daten sind nicht nur wichtig um dem Chef Aussagen über den Erfolg einer Marketingaktion zu liefern (ROI - Return-On-Invest), sondern auch um die Webseiten für Kunden benutzerfreundlicher und übersichtlicher zu gestalten. Die interpretierten Statistiken ermöglichen es, den Aufbau und die Struktur der Internetseite zu optimieren.

Eine weiterer wichtiger Aspekt ist die Erhebung aussagekräftiger Statistiken, die zur Festlegung angemessener Preise beispielsweise für Werbebanner, die auf einer Webseite

eingebunden sind, dienen. Die Analyse der Logfiles liefert hier interessante Aussagen über die Anzahl an Internet-Usern, die diese Werbebanner täglich sehen.

Außerdem können den Kunden Verhaltensmuster zugeordnet werden, die es ermöglichen die Navigation auf der Webseite dem Verhalten der Kunden anzupassen und öfter angeklickte Links beispielsweise in der Navigationsleiste höher einzuordnen.

Misst man z.B. kurze Verweildauern auf Internetseiten, dann ist dies ein Zeichen dafür, dass die Besucher nicht den richtigen Inhalt gefunden haben oder dass die Seite nicht gefällt. Diese Aussagen würden Änderungen entweder am Seiteninhalt oder am Aufbau der Seite veranlassen.

Wird eine zu hohe Ladezeit der Seite gemessen, muss entweder die Datenmenge auf der Seite reduziert werden, z.B. durch kleinere Bilder oder falls die Verzögerung serverbedingt ist, sollte darüber nachgedacht werden den Webhoster zu wechseln.

Auch interessant für Webseiten-Entwickler ist es zu messen, wie die Besucher von der Seite erfahren. Misst man z.B. dass 25% der Anfragen über ein Werbebanner auf einer anderen Seite kommen, dann wäre es sinnvoll dieses Banner auf der Seite zu belassen.

2.2 Motivation der Webserver-Betreiber

Webanalysen und Messungen spielen auch beim Verwalten eines Webservers eine große Rolle.

Wird z.B. gemessen, dass Besucher auf eine Webseite nur nachts zugreifen, und auf einer anderen Seite wird hauptsächlich tagsüber gesurft, dann wäre es sinnvoll diese beiden Seiten auf einen Server zu spielen um den Webtraffic insgesamt ausgeglichen zu halten.

Außerdem sind Webseiten-Analysen wichtig um Performance-Informationen zu gewinnen.

Misst man häufige Verzögerungen beim Download einer Webseite, wäre es sinnvoll über Investitionen in High-Speed-Verbindungen ans Netzwerk nachzudenken. Weiterhin helfen diese Informationen, die an Webservern verstellbaren Parameter optimal zu konfigurieren.

2.3 Motivation der Netzwerk-Betreiber

Auch für Netzwerk-Betreiber liefern Webanalysen wertvolle Informationen.

Beispielsweise könnte ein Betreiber eines LAN (Local Area Network) nach Auswertung der Logfiles Informationen über den Bedarf weiterer Caching-Proxies erhalten. Messungen von Webtraffic liefern geschätzte Prozentzahlen von Aufrufen, die durch einen weiteren Caching-Proxy bearbeitet und somit schneller beantwortet werden können.

Ebenso könnte ein ISP (Internet Service Provider) Messungen durchführen wieviel Bandbreite durch weitere Caching-Proxies gespart werden würde und somit den Netzzugang beschleunigen.

Generell kann man sagen, dass Webanalysen für Netzwerk-Betreiber Engpässe an Netzwerk-Equipment aufzeigen kann, die dann durch geeignete Erweiterung oder Konfiguration behoben werden können.

3 Log-Techniken

Messungen an Internetseiten können auf verschiedenste Art und Weise durchgeführt werden. Wobei jede Alternative Vor- und Nachteile aufzeigt und sogar verschiedene Arten von Informationen gewonnen werden können.

Im folgenden diskutieren wir über die Unterschiede zwischen Server-Logging, Proxy-Logging, Client-Logging und Packet-Monitoring.

3.1 Server-Logging

Jeder Server generiert defaultmäßig eine Logdatei für Zugriffe auf eine Webseite. Jeder Eintrag in die Logdatei steht für einen HTTP-Aufruf an den Server und loggt Informationen über den aufrufenden Client, dessen IP-Adresse, den Zeitpunkt und die angeforderte Datei.

Ein Logfile-Eintrag kann demnach ungefähr so aussehen:

```
62.104.191.241 - - [01/Dec/2004:11:39:00+0100] GET /h.png HTTP/1.1 200 302
```

Mehr zu Formaten von Logfiles in Kapitel 4

Ein großer Nachteil von Server-Logs ist, dass HTTP-Aufrufe, die durch Proxy- oder Browser-Caches beantwortet werden, aus Server-Logs nicht ersichtlich sind. Somit geht ein Großteil an Informationen über Webseitenaufrufe verloren.

Dies kann zwar durch Verbot des Caching bei Proxies und Browsern anhand von Restriktionen im Header behoben werden, allerdings würden diese Verbote mehr Traffic verursachen und Caching uneffektiv machen.

Sollen Logfiles zur Ermittlung von Verhaltensmustern dienen, ergeben sich weitere Nachteile des Serverlogs. Da HTTP ein zustandsloses Protokoll ist, ist es fast unmöglich die einzelenen HTTP-Anfragen bestimmten Clients zuzuordnen. Theoretisch könnte man die Clients anhand ihrer geloggten IP-Adressen unterscheiden, jedoch gibt es hier in den meisten Fällen Probleme.

Zum einen tauchen Clients, die über einen Proxy zu dem Server gelangen nur mit der IP-Adresse des Proxys im Logfile auf. Zum anderen werden IP-Adressen von ISPs dynamisch vergeben und können somit auch keinem bestimmten User zugewiesen werden. Eine Möglichkeit User eindeutig zu identifizieren, besteht darin die Benutzung von Cookies zu erzwingen.

Obwohl es noch keinen offiziellen Standard für Logfile-Formate gibt, halten sich die meisten Webserver an einen inoffiziellen Standard. Diese Formate CLF und ECLF werden in Kapitel 4.1 und 4.2 näher behandelt.

Die Benutzung dieser inoffiziellen Standards ermöglicht die Erstellung von Analyse-Tools.

3.2 Proxy-Logging

Auch Proxies loggen genau wie Server HTTP-Aufrufe mit. Im Gegensatz zum Server loggt ein Proxy allerdings Aufrufe zu vielen verschiedenen Webseiten im Internet, vor allem, wenn der Proxy nah am Client stationiert ist.

Zum Beispiel kann ein Proxy eines Unternehmens alle Aufrufe ins World Wide Web von Mitarbeitern der Firma loggen.

Ein Proxy hat oft detailliertere Informationen über den aufrufenden Client als der Server, auf dem die aufgerufene Webseite liegt. Der erste Proxy, der in einer Kette von Proxies am nächsten am Client liegt, kann die verschiedenen User anhand ihrer IP-Adressen unterscheiden. Wenn der HTTP-Aufruf diesen Proxy durchläuft, wird nur noch die IP-Adresse des Proxys angezeigt, welche keine einfache Unterscheidung der einzelnen Personen einer Firma mehr ermöglicht (Ausnahme: Cookies). Somit können im Proxy-Logfile im Gegensatz zu Server-Logfiles aussagekräftigere Informationen über Verhaltensmustern von Usern gewonnen werden.

Ein weiterer Vorteil des Proxy-Loggings ist, dass der Proxy alle Aufrufe loggt, die durch den Proxy-Cache erfüllt werden. Diese wichtigen Informationen gelangen aber nicht zum Server. Da nun aber die beliebtesten Webseiten meist schon im Cache des Proxys zwischengespeichert sind, werden die Analysedaten über Webtraffic bestimmter Webseiten, die erst auf dem Server geloggt werden, verzerrt.

Ein großer Nachteil des Proxy-Logging ist, dass hier nur ein kleiner Teil an Usern beobachtet werden kann, der nicht repräsentativ für alle Nutzer des World Wide Webs sein muss. Es ist schwer aus diesen gesammelten Daten aussagekräftige Werte für bestimmte Webseiten zu gewinnen, da nicht alle Anfragen an einen bestimmten Server diesen Proxy durchlaufen.

ähnlich wie beim Server-Log besteht beim Proxy das Problem, dass Aufrufe an Webseiten, die im Browser-Cache zwischengespeichert sind, nicht geloggt werden können.

Ferner werden nicht alle LogFiles auf Proxies öffentlich zugreifbar gemacht.

3.3 Client-Logging

Die Erstellung von LogFiles am Client bietet sehr detaillierte Informationen über den aufrufenden User und dessen Verhaltensmuster. Es können sogar vorzeitig gestoppte oder vom Browser-Cache beantwortete Anfragen geloggt werden, die den Proxy oder Zielserver nie erreichen.

Das Client-Logfile speichert Informationen über nur einen einzelnen Client, welche nicht repräsentativ für alle User des Internets sind. Diese Informationen sind jedoch sehr viel detailreicher als die eines Servers, da durch Logging aller Header-Informationen eines HTTP-Requests nicht so ein enormer Overhead erzeugt wird, wie dies auf dem Server der Fall wäre.

Im Gegensatz zu Proxy- und Server-Logging basieren Browser-Logfiles nicht auf einem inoffiziellen oder offiziellen Standard. Die bekannteren Browser schreiben defaultmäßig keine Logfiles. Hier bedarf es einer Modifizierung des Source-Codes um das Generieren von Logfiles zu ermöglichen.

Zur Analyse von Verhaltensmustern der User braucht man allerdings eine große Anzahl an Daten verschiedener User, deshalb ist ohne weiteres nicht möglich aus Browser-Logfiles Statistiken über Verhaltensmuster zu generieren, die allgemeingültig sind.

3.4 Packet-Monitoring

Die letzte Alternative zur Generierung von Logfiles, die wir hier vorstellen, ist das Packet-Monitoring. Bei dieser Methode werden Informationen direkt auf der Netzwerkschicht abgerufen. Das hat den Vorteil, dass kein zusätzlicher Overhead durch Daten, die auf höheren Schichten verschickt werden müssen, entsteht.

Beim Packet-Monitoring werden IP-Pakete, die durch Netzwerk-Geräte wie z.B. Router gesendet werden, kopiert und ausgewertet. Hierbei können die zeitlichen Verläufe von Übertragungen durch das Netz detailliert erfasst werden. Im Gegensatz zum Server oder Proxy, die nur einen Zeitpunkt pro HTTP-Anfrage speichern können, können beim Packet-Monitoring die Zeitpunkte jedes einzelnen IP-Pakets festgehalten werden. Dies ist vor allem wichtig zur Analyse von Verzögerungen, Durchsatz und Fehlerhäufigkeit bei übertragungen durch das Netz.

Weiterhin können beim Packet-Monitoring Informationen der Transportschicht geloggt werden. Zum Beispiel wann eine TCP-Verbindung gestartet oder beendet wird. Somit kann auch festgestellt werden, ob und wann HTTP-Verbindungen frühzeitig unterbrochen wurden. Diese Informationen sind von Servern und Proxies meist nicht zugreifbar.

Ebenso wie Server- und Proxy-Logs kann aber auch ein Packet-Monitor keine HTTP-Anfragen loggen, die vom Browser-Cache beantwortet oder durch SSL verschlüsselt worden sind.

Ein Nachteil dieser Methode ist, dass ein Packet-Monitor für Informationen auf HTTP-Ebene die einzelnen IP-Pakete zu einer HTTP-Anfrage zusammenfügen muss, was durch die Vielzahl von IP-Paketen, die durch das Netz geschickt werden erschwert wird. Außerdem verfolgen nicht alle zu einer HTTP-Anfrage gehörenden IP-Pakete dieselbe Route durch Internet, was die Rekonstruktion einer Anfrage fast unmöglich macht. Bei dieser Alternative müssen jedoch die Regelungen zur Erfassung personenbezogener Daten im Bundesdatenschutzgesetz beachtet werden.

4 Log-Formate

Die meisten Webserver und Proxies generieren ihre Logfiles nach einem inoffiziellen Standard. Zum einen gibt es das CLF (Common Log Format) und zum anderen das ECLF (Extended Common Log Format).

In den nächsten 2 Kapiteln werden wir die zwei weitgenutzten Log-Formate vorstellen.

Apache, der wohl meist verbreitetste Webserver, kann in beiden Formaten loggen. Defaultmäßig loggt er im CLF, dies kann jedoch sehr leicht umgestellt werden. In der Konfigurationsdatei httpd.conf, die im Installationsverzeichnis zu finden ist, findet man die folgende Zeile:

```
CustomLog logs/access.log common
```

Common steht hier für Common Log Format. Modifiziert man diese Zeile nun zu

```
CustomLog logs/access.log combined
```

loggt der Apache im ECLF. Auch der angegebene Pfad für die Logdatei kann modifiziert werden. Dieser ist relativ zum Installationsverzeichnis.

4.1 CLF (Common Log Format)

Jeder Eintrag in Form eines CLF-Logs besteht aus folgenden 7 Feldern:

FELD	BEDEUTUNG
Remote Host	Hostname oder IP-Adresse des aufrufenden Clients
Remote Identity	Entsprechender Account zur Applikation auf Clientseite
Authenticated user	Bei Authentifizierung angegebener Username
Time	Datum/Uhrzeit des Aufrufs
Request	Aufruf-Methode, Request-URI und Protokollversion
Response code	http-Antwort (3 Digits)
Content length	Anzahl der Bytes in der Antwort

Tabelle 1: CLF (Common Log Format) Felder

Im Folgenden werden wir jedes Feld des CLF-Logs ausführlich erläutern.

4.1.1 Remote Host

Der Remote Host identifiziert die IP-Adresse oder den Hostnamen des Clients. Die IP-Adresse kann der Server direkt aus dem Socket des HTTP-Aufrufes entnehmen. Um den Hostnamen zu erhalten muss der Server eine Reverse DNS-Abfrage (Domain Name Server) starten. Ist die DNS-Abfrage erfolglos, speichert der Server die IP-Adresse des Clients im Log. überlastete Webserver können den Reverse DNS-Aufruf abschalten.

4.1.2 Remote Identity

Die Remote Identity identifiziert den Besitzer der Applikation, die auf der Client-Seite der TCP-Verbindung, die Anfrage startet. Normalerweise ist der zugehörige Besitzername einer Applikation am anderen Ende der TCP-Verbindung nicht abrufbar. Es gibt jedoch ein Identification Protocol RFC 1413, welches einen entfernten Host nach diesen Informationen befragt. Da diese Abfrage allerdings sehr zeitaufwändig ist, wird sie im Normalfall nicht gestartet und das Feld Remote Identity im CLF-Log bleibt leer.

4.1.3 Authenticated user

Dieses Feld loggt den Namen, der im HTTP-Header unter Authorization angegeben ist. Dieses Feld ist nur gefüllt, wenn auf passwort-geschützte Informationen zugegriffen wird. Es ist also der korrespondierende Login-Name zu dem Passwort.

4.1.4 Time

Ein CLF-Log speichert den Zeitpunkt der HTTP-Anfrage mit einer Genauigkeit von 1 Sekunde. Da dieses Feld des CLF-Logs nicht genau definiert ist, variiert der geloggte Zeitpunkt abhängig von dem Zeitpunkt in der Verarbeitung der HTTP-Anfrage, an dem der Server das System nach der genauen Zeit fragt. Zum Beispiel kann der Server den System-Call zur Abfrage der Uhrzeit starten, wenn die HTTP-Anfrage gerade ankommt oder wenn er die Antwort in den Socket Buffer schreibt. Diese Zeitpunkte können bei großen Datenmengen stark variieren.

4.1.5 Request

Das Feld Request loggt die Art des HTTP-Requests wie z.B. GET oder POST, die Request-URI, z.B. /img/fhlogo.jpg und die Versionsnummer des verwendeten Protokolls.

4.1.6 Response Code

Der Response Code ist ein 3-Digit-Code, der indiziert, ob eine Anfrage ok war oder evtl. ein Fehler aufgetreten ist. Z.B.: 404 steht für Seite nicht gefunden.

4.1.7 Content Length

Das Feld Content Length beinhaltet die Länge der HTTP-Antwort in Bytes. Dieses Feld ist wie das Feld Time nicht genau definiert. Manche Server loggen die Länge der gesamten Nachricht, andere wiederum loggen nur die Länge des Entity Bodies.

Nun noch einmal zu dem Beispiel aus Kapitel 3.1.:

```
62.104.191.241 - - [01/Dec/2004:11:39:00+0100] GET /h.png HTTP/1.1 200 30
```

Dieser CLF-Log sagt aus, dass der Client mit der IP-Adresse 62.104.191.241 am 1.12.04 um 11:39:00 Uhr die Datei h.png angefragt hat. Dazu benutzter er die HTTP-Version

1.1. Die Antwort-Code war 200 für ok und die Antwort war 302 Bytes lang. Es wurden keine Remote Identity und kein Authenticated user geloggt.

4.2 ECLF (Extended Common Log Format)

Das Extended Common Log Format basiert auf dem CLF, jedoch werden noch zusätzliche Felder geloggt. Das ECLF-Format gibt nicht genau vor welche Felder zusätzlich erfasst werden. Im folgenden beschreiben wir alle möglichen zusätzlichen Felder für ein Logfile im ECLF-Format.

FELD	BEDEUTUNG
User agent	Informationen über user agent Software
Referer	URI, von der die Request-URI gesendet wurde
Request processing time	Benötigte Zeit zur Bearbeitung dieser Anfrage
Request header size	Größe des HTTP-Headers in Bytes
Request body size	Größe des HTTP-Bodys in Bytes
Remote response code	Antwort-Code des Servers
Remote content length	Größe der Server-Antwort
Remote response header size	Größe des HTTP-Headers der Server-Antwort
Proxy request header size	Größe des vom Proxy gesendeten HTTP-Headers
Proxy response header size	Größe des vom Proxy zu Client gesendeten Antwort-Header

Tabelle 2: ECLF (Extended Common Log Format) Felder

Im Folgenden werden wir jedes zusätzliche Feld des ECLF-Logs ausführlich erläutern.

4.2.1 User agent

Dieses Feld beinhaltet den Namen und die Version der Software, die auf Client-Seite den HTTP-Request sendet. In der Praxis kann dieses Feld auch weiterführende Informationen enthalten, wie z.B. das Betriebssystem, das auf dem Client läuft.

4.2.2 Referer

Das Feld Referer enthält die URI der Webseite, von der der Client den Request sendet. Klickt der User beispielweise auf einen Link in einer Suchmaschine, wird die URI dieser Suchmaschine als Referer mitgeschickt.

4.2.3 Request processing time

Die Request processing time ist die Zeit, die der Server braucht um die HTTP-Antwort zu generieren. Von besonderem Interesse ist dieses Feld, wenn der Server eine relativ lange Zeit benötigt um beispielsweise ein Script zu starten.

4.2.4 Request header size

Hier wird die Anzahl der Bytes im HTTP-Header des Requests geloggt.

4.2.5 Request body size

Dieses Feld speichert die Größe des HTTP-Bodys des Requests in Bytes. Ein HTTP-Request besitzt nur einen Body, wenn die Methoden PUT oder POST benutzt werden.

Die fehlenden Felder eines ECLF-Logs beziehen sich auf Proxies. Ein Proxy dient gleichzeitig als Server und als Client beim Durchreichen einer Nachricht. Daher kann ein Proxy andere Daten loggen als beispielsweise ein Zielserver.

Diese Felder können von Nutzen sein, um die Performanz eines Proxy zu analysieren.

4.2.6 Remote response code

Dieses Feld indiziert den Status der Antwort anhand des 3-Digit-Codes, der vom Zielserver an den Proxy gesendet wird. Er kann sich unterscheiden von dem Code, der vom Proxy an den Client gesendet wird.

4.2.7 Remote content length

Hier wird die Länge der Antwortnachricht an den Proxy in Bytes gespeichert. Dieser Wert kann sich unterscheiden zur Content length, die von Proxy an Client gesendet wird.

4.2.8 Proxy request header size

Dieses Feld enthält die Länge des HTTP-Headers, der von Proxy zu Server geschickt wird. Dieses Feld kann wiederum einen anderen Wert enthalten, als die Länge des Headers, der von Client an Proxy gesendet wird.

4.2.9 Proxy response header size

Die Proxy response header size beinhaltet die Anzahl an Bytes des HTTP-Headers von Proxy an Client.

5 Instrumentarium

5.1 Kenngrößen

Um eine aussagekräftige Analyse zu kochen werden einige Basiszutaten benötigt, diese lassen sich direkt aus den Logfiles gewinnen. Wenn man diese Kenngrößen miteinander kombiniert, gewinnt man sehr aussagekräftige Kennzahlen, dabei muss man sich aber immer im klaren sein, dass die Zuverlässigkeit immer von jeder einzelnen Zutat abhängt. Im folgenden werden die einzelnen Bestandteile einer Loganalyse genauer beleuchtet und welche Kombinationen es gibt.

5.1.1 Hits

Unter einem Hit versteht man die Anforderung einer einzelnen Datei vom Webserver, dabei ist es nicht relevant, um was für eine Datei es sich handelt. Für die Logfile Analyse hat der Hit an sich fast keine praktische Bedeutung. Jedoch alle weiteren Kenngrößen leiten sich aus dem Hit ab, diese sind von größerem Interesse, um eine Aussage über den Erfolg einer Website machen zu können. Auch erfolglose Anfragen werden als Hit gezählt zum Beispiel wenn die Datei nicht vorhanden ist. Manche Analyseprogramme ignorieren fehlerhafte Zugriffe. Falls zwischen erfolgreichen und erfolglosen Hits unterschieden wird, kann das Verhältnis eine Bedeutung für die Optimierung von Serverprozessen haben. Das Logfile auf dem Webserver wächst proportional mit der Anzahl der Hits. Andere Begriffe für Hit sind zum Beispiel Treffer, Abgerufene Files, total files sent und total hits.

5.1.2 PageView

Die Anfrage nach einem HTML-Dokument wird als PageView (Seitenabruf) bezeichnet. Zu einem Seitenabruf gehören auch alle Dateien, die in das HTML-Dokument eingebettet sind, zum Beispiel Bilder, CSS-Dateien, eingebundene Programme und weitere multimediale Elemente. Um die Anzahl der Seitenabrufe zu ermitteln, werden alle Anfragen nach Dateien mit der Endung .htm oder .html gezählt. Für statische Webseiten werden dadurch zuverlässige Statistiken erstellt. Jedoch bei dynamischen Webseiten, die im Moment der Anforderung erzeugt werden, ist diese Methode nicht anwendbar, dies liegt an den unterschiedlichen Dateiendungen zum Beispiel .php, .jsp, .js und .asp.

Ein weiteres Problem bei der Erfassung von Seitenabrufen gibt es, bei der Verwendung von Frames. Dabei besteht eine angeforderte Webseite aus mehreren HTML-Dokumenten. Der Anwender hat den Eindruck (Impression), dass nur eine Webseite aufgerufen wird, jedoch werden mehere HTML-Seiten jeweils in die einzelen Frames geladen. Dadurch vervielfachen sich auch die Einträge in den Logfiles. Deshalb wird bei Frameseiten zusätzlich zwischen Seitenabrufen und Page-Impression unterschieden. Page-Impression bedeutet, durch eine Aktion des Nutzer, wird ein neuer Inhalt in ein Framefenster geladen.

Andere Begriffe für Seitenaufruf sind: Sichtkontakt, Seitenansicht, Seiten-Impression, Page View und Abruf von Content-Pages.

5.1.3 Visits

Ein Visit ist der Besuch eines Nutzers auf einer Website. Dies hört sich recht trivial an, aber es ist in der Praxis ein schwieriges Unterfangen, einen Visit exakt zu bestimmen. Der Grund dafür liegt im HTTP-Protokoll, es ist zustandslos, daher kann keine Sitzung zwischen Browser und Webserver etabliert werden. Daher ist man auf die Analyse der Logfiles angewiesen. Es gibt zwei wichtige Parameter, die den Visit definieren, die IP-Adresse und der Time-Out. Ein Analyseprogramme geht davon aus, dass sich die IP-Adresse des Internetbenutzer nicht ändert. Mehrere PageViews von der gleichen IP-Adresse werden als Visit interpretiert. Falls nach einer bestimmten Zeit (Time-Out) keine PageView mehr erfolgt, gilt der Besuch als beendet. Jedoch ist es nicht so, dass jeder Internetbenutzer eine eigene IP hat, die nur er alleine benutzt. Falls der Benutzer z.B. über einen Proxyserver ins Internet geht, wird als IP-Adresse die Adresse des Proxyservers geloggt, andere können auch diesen Proxyserver benutzen, so dass man die Benutzer nicht voneinander unterscheiden kann. Auf die Problematik der Proxyserver wird nochmal genauer in Kapitel 7.1 eingegangen. Große Provider besitzen einen Pool von IP-Adressen, diese werden dynamisch verteilt, das heißt ein Anwender kann eine Adresse zugeteilt bekommen, die kurz vorher noch in Gebrauch war. Es gibt zwar Verfahren, damit eine Visit eindeutig zugeordnet werden kann, doch diese erfordern einen erhöhten Aufwand, z.B. die Verwendung von Cookies.

5.1.4 Abgeleitete Werte

Es gibt weitere Kennzahlen, die sich aus den Basis-Kenngrößen ableiten lassen:

- Zeit pro PageView
- PageViews pro Visit
- Zeit pro Visit
- Besucher
- Visits pro Besucher
- Einmalige Besucher
- Zeit pro Besucher
- PageViews pro Tag
- Visits pro Tag

6 Analyse

Wichtige Fähigkeiten von Analyseprogrammen sind:

- Auswertung der Besucherdaten nach verschiedenen Intervallen (Stunden, Wochentage, Tage, Wochen, Monate, Quartal und Jahre).
- Vergleichsmöglichkeiten mit den jeweiligen Vorzeiträumen.
- Darstellung wichtiger Kenngrößen.

6.1 Beispiel WebSuxess

Abbildung 1: WebSuxess - Logo

WebSuxess ist ein Web-Site-Analyseprogramm zur professionellen Analyse von statischen und dynamischen Webseiten. Aus Logfiles werden mehr als 150 grafisch aufbereitete Statistiken erzeugt. Auf Wunsch können diese auch als HTML-Dateien ausgegeben werden. Das Programm wird von der Exody E-Business Intelligence GmbH in Eschborn hergestellt und vertrieben. Es wird unabhängig vom Webserver auf einem Windows Arbeitsplatz-PC installiert. Änderungen an den Webseiten oder am Webserver sind nicht nötig. Der Anwender kann sich die Logfiles auf seinen Arbeitsplatz runterladen und in WebSuxess importieren. Das Programm kann sich die Logfiles auch direkt von einem FTP/HTTP-Server herunterladen. Die folgenden Kapitel beziehen sich auf die WebSuxess4 Enterprise Version. Es werden die einzelnen Analyse-Rubriken vorgestellt.

Grundlage der folgenden Statistiken sind die Logfiles von meiner Homepage *www.ralef.de* vom 31.10.2004 bis zum 10.12.2004.

6.1.1 Zusammenfassung

Die Zusammenfassung enthält die wichtigsten Kenngrößen: Hits, PageViews, Visits, Besucher und weitere abgeleitete Durchschnittswerte. Siehe Kapitel 5.1. Dabei ist zu beachten, dass sich einige Kenngrößen z.B. Bytes insgesamt, immer auf das gesamte Logfile beziehen, auch wenn Filter eingeschaltet sind.

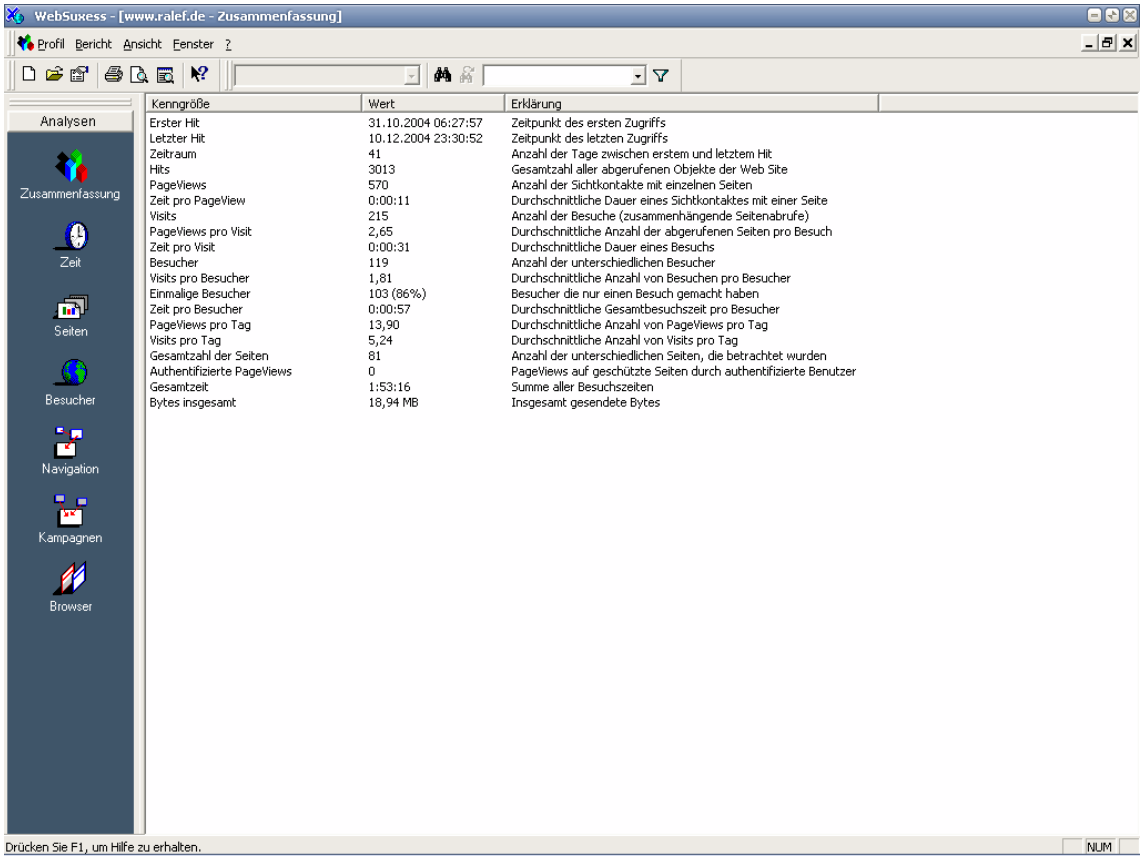
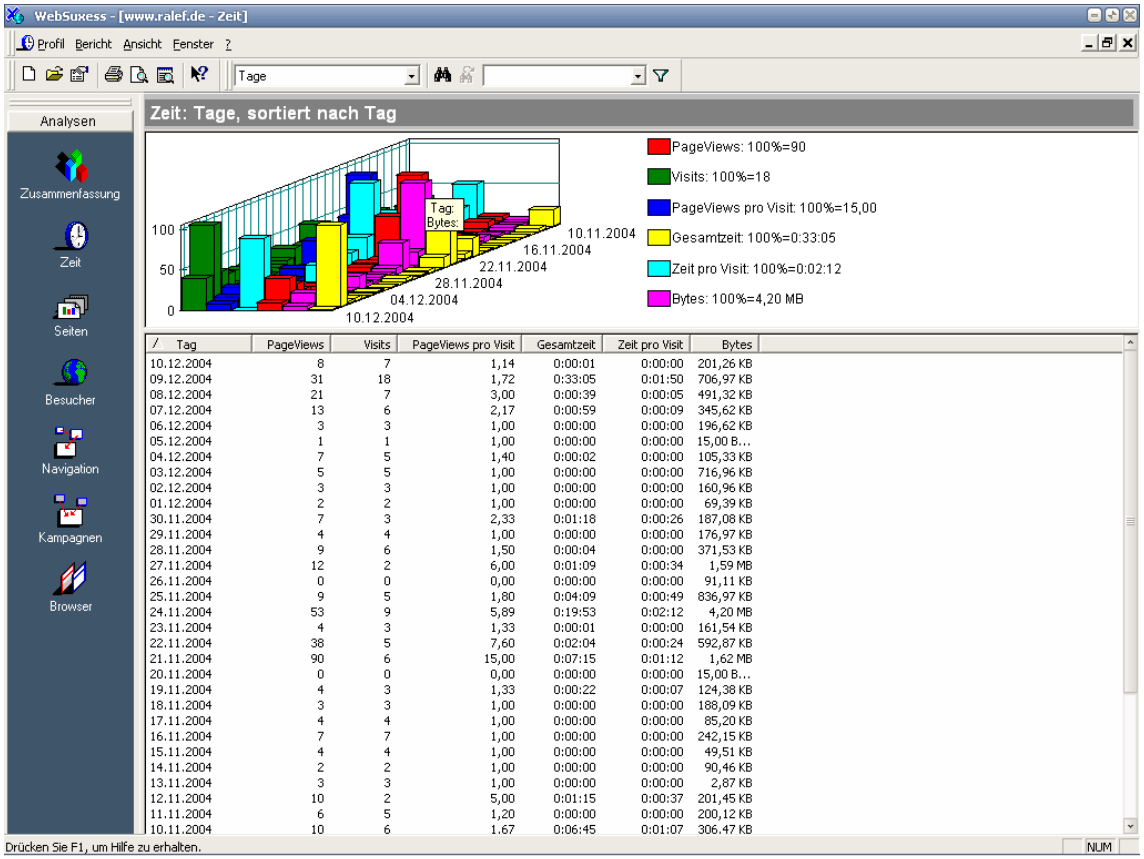


Abbildung 2: WebSuxess - Zusammenfassung

6.1.2 Zeitreihen

Die Zeitreihen können in verschiedenen Intervallen, Stunden, Wochentage, Tage, Wochen, Monate, Quartal und Jahre, dargestellt werden. In der Datentabelle können die Werte nach Zeit, PageViews, Visits, PageViews pro Visit, Gesamtzeit, Zeit pro Visit oder Bytes, auf-/absteigend sortiert werden.



Tag	PageViews	Visits	PageViews pro Visit	Gesamtzeit	Zeit pro Visit	Bytes
10.12.2004	8	7	1,14	0:00:01	0:00:00	201,26 KB
09.12.2004	31	18	1,72	0:33:05	0:01:50	706,97 KB
08.12.2004	21	7	3,00	0:00:39	0:00:05	491,32 KB
07.12.2004	13	6	2,17	0:00:59	0:00:09	345,62 KB
06.12.2004	3	3	1,00	0:00:00	0:00:00	196,62 KB
05.12.2004	1	1	1,00	0:00:00	0:00:00	15,00 B...
04.12.2004	7	5	1,40	0:00:02	0:00:00	105,33 KB
03.12.2004	5	5	1,00	0:00:00	0:00:00	716,96 KB
02.12.2004	3	3	1,00	0:00:00	0:00:00	160,96 KB
01.12.2004	2	2	1,00	0:00:00	0:00:00	69,39 KB
30.11.2004	7	3	2,33	0:01:18	0:00:26	187,08 KB
29.11.2004	4	4	1,00	0:00:00	0:00:00	176,97 KB
28.11.2004	9	6	1,50	0:00:04	0:00:00	371,53 KB
27.11.2004	12	2	6,00	0:01:09	0:00:34	1,59 MB
26.11.2004	0	0	0,00	0:00:00	0:00:00	91,11 KB
25.11.2004	9	5	1,80	0:04:09	0:00:49	836,97 KB
24.11.2004	53	9	5,89	0:19:53	0:02:12	4,20 MB
23.11.2004	4	3	1,33	0:00:01	0:00:00	161,54 KB
22.11.2004	38	5	7,60	0:02:04	0:00:24	592,87 KB
21.11.2004	90	6	15,00	0:07:15	0:01:12	1,62 MB
20.11.2004	0	0	0,00	0:00:00	0:00:00	15,00 B...
19.11.2004	4	3	1,33	0:00:22	0:00:07	124,38 KB
18.11.2004	3	3	1,00	0:00:00	0:00:00	188,09 KB
17.11.2004	4	4	1,00	0:00:00	0:00:00	85,20 KB
16.11.2004	7	7	1,00	0:00:00	0:00:00	242,15 KB
15.11.2004	4	4	1,00	0:00:00	0:00:00	49,51 KB
14.11.2004	2	2	1,00	0:00:00	0:00:00	90,46 KB
13.11.2004	3	3	1,00	0:00:00	0:00:00	2,87 KB
12.11.2004	10	2	5,00	0:01:15	0:00:37	201,45 KB
11.11.2004	6	5	1,20	0:00:00	0:00:00	200,12 KB
10.11.2004	10	6	1,67	0:06:45	0:01:07	306,47 KB

Abbildung 3: WebSuxess - Zeitreihen

6.1.3 Seiten

In der Rubrik: Seiten werden alle Dokumente angezeigt, die von den Besuchern der Webseite angefordert wurden. Diese Ansicht kann nach PageView (siehe Kapitel 5.1.2), Gesamtzeit, Zeit pro PageView, Einstiegsseite, Ausstiegsseite, Einzige Seite und Bytes sortiert werden. Die Verzeichnisse, in denen die Dokumente liegen, können in der gleichen Form dargestellt werden.

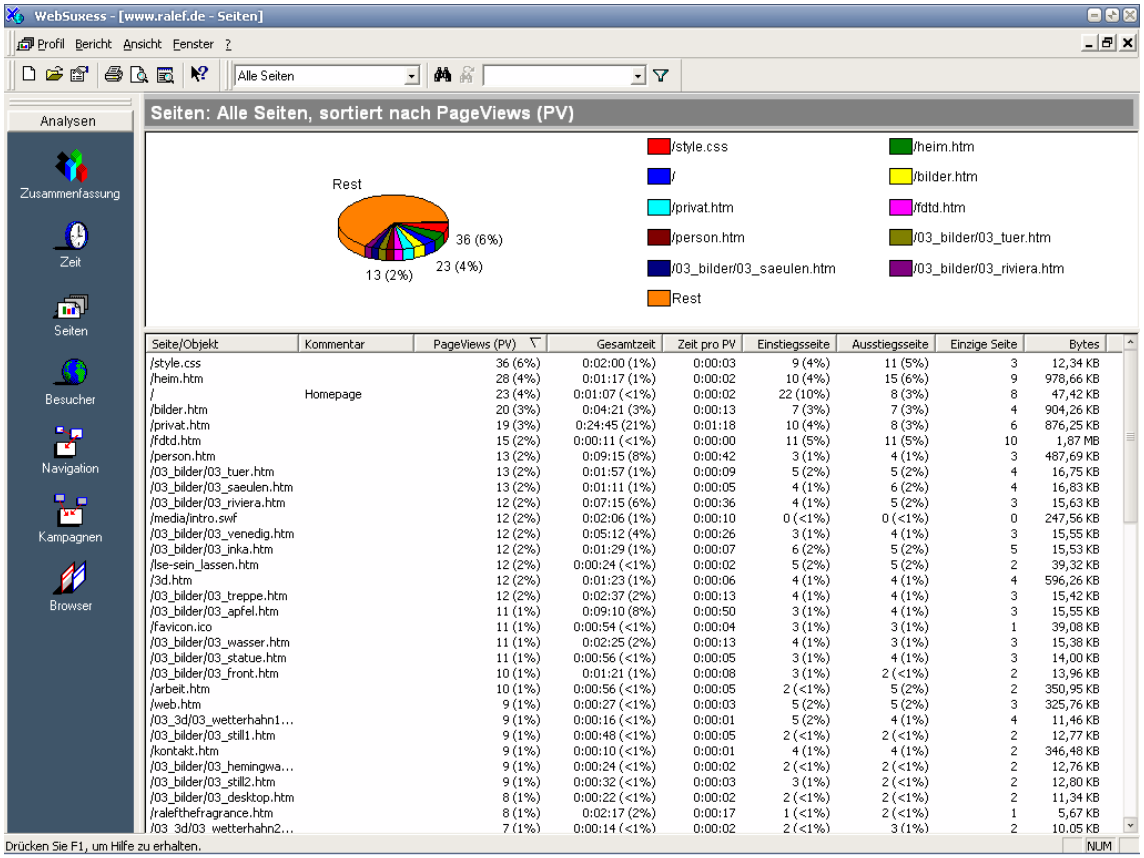


Abbildung 4: WebSuxess - Seiten

6.1.4 Besucher

Bei den Besuchern gibt es folgende Gruppierungen zur Auswahl:

- *Länder*: Sortiert nach Besuche
- *Firmen*: Sortiert nach Besucher
- *Rechner*: Sortiert nach PageViews
- *Suchmaschinen*: Sortiert nach Roboter
- *Roboter*: Sortiert nach Suchmaschine
- *Autorisierte Benutzer*: Sortiert nach PageViews

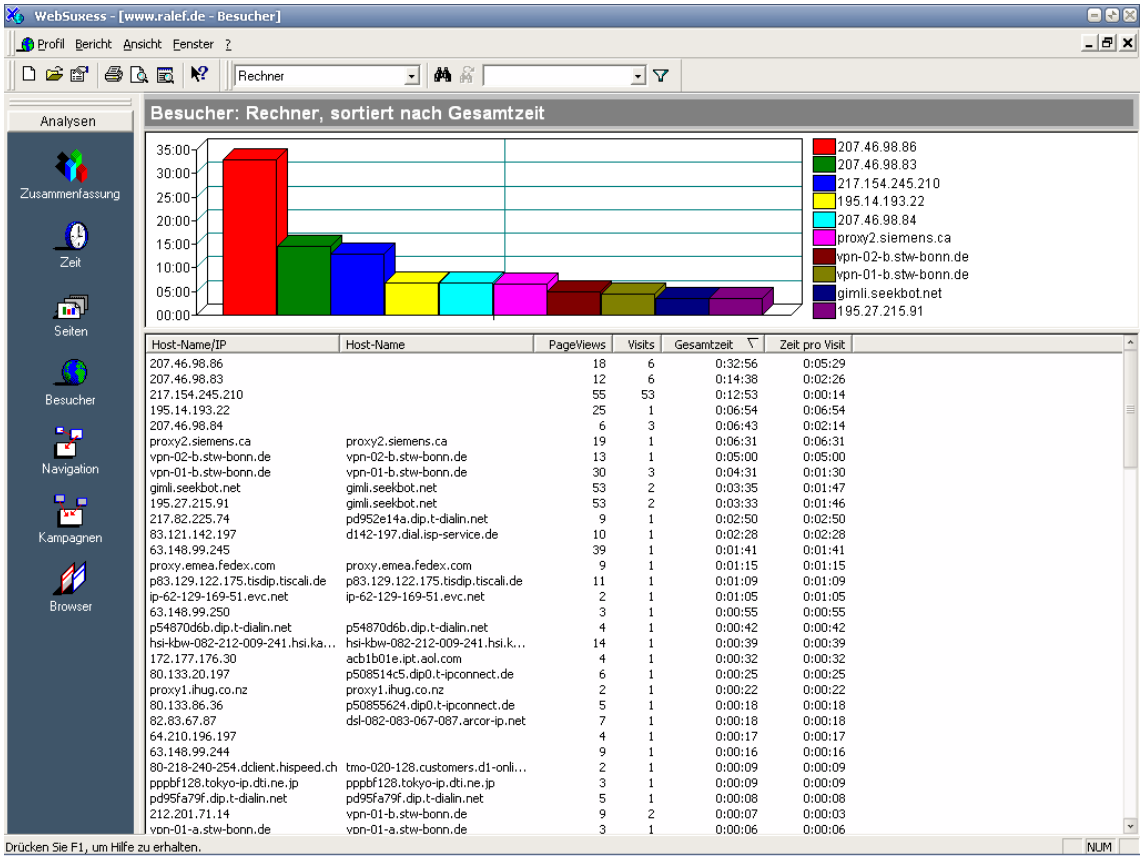


Abbildung 5: WebSuxess - Besucher

6.1.5 Navigation

In der Rubrik: Navigation wird der Pfad angezeigt, welche Dokumente ein Besucher, während einer Sitzung, angefordert hat. Daraus lässt sich nachvollziehen, wie ein Besucher sich durch die Webpräsenz navigiert.

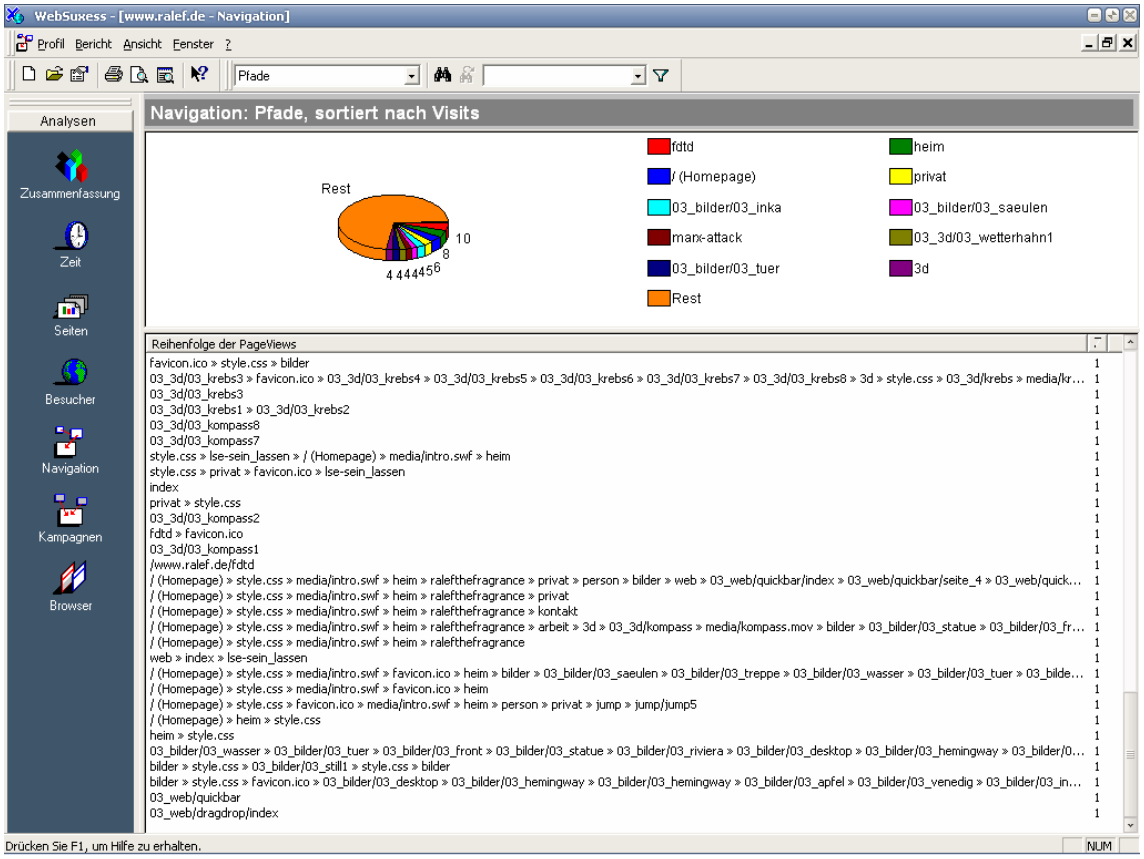


Abbildung 6: WebSuxess - Navigation

6.1.6 Kampagnen

In der Rubrik: Kampagne wird die Referer-Information, siehe Kapitel 4.2.2, im Logfile ausgewertet. Die Kampagne dient dazu, zu erkennen, über welche Kampagnenform, etwa Banner oder Suchmaschine, die Besucher zur Website gefunden haben.

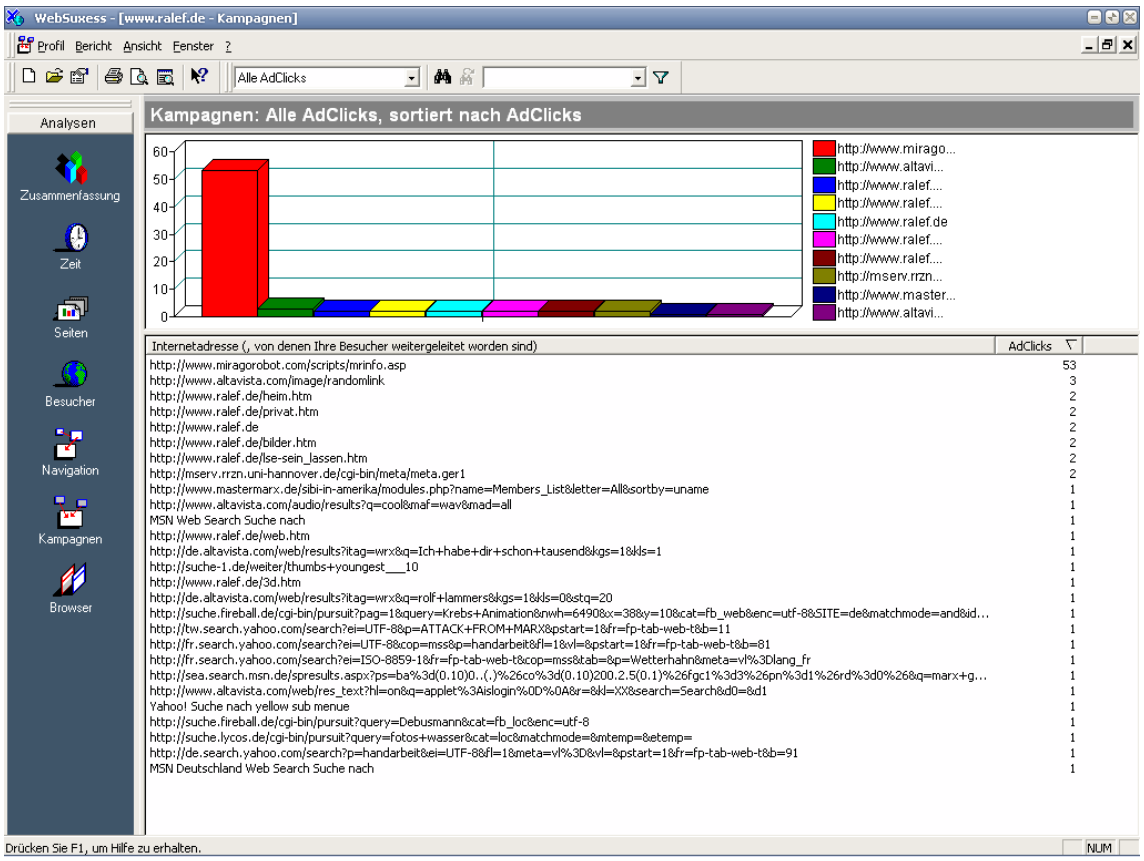


Abbildung 7: WebSuxess - Kampagnen

6.1.7 Browser

Anhand einer umfangreichen Browserdatenbank, werden in der Rubrik: Browser detaillierte Informationen über die Systemkonfiguration des Besuchers angezeigt. Dadurch lässt sich ermitteln was für eine Betriebssystem auf dem Benutzer Rechner läuft, mit welchem Browser Dokumente angefordert wurden und welche Fähigkeiten z.B. ActiveX Controls die eingesetzten Browser haben.

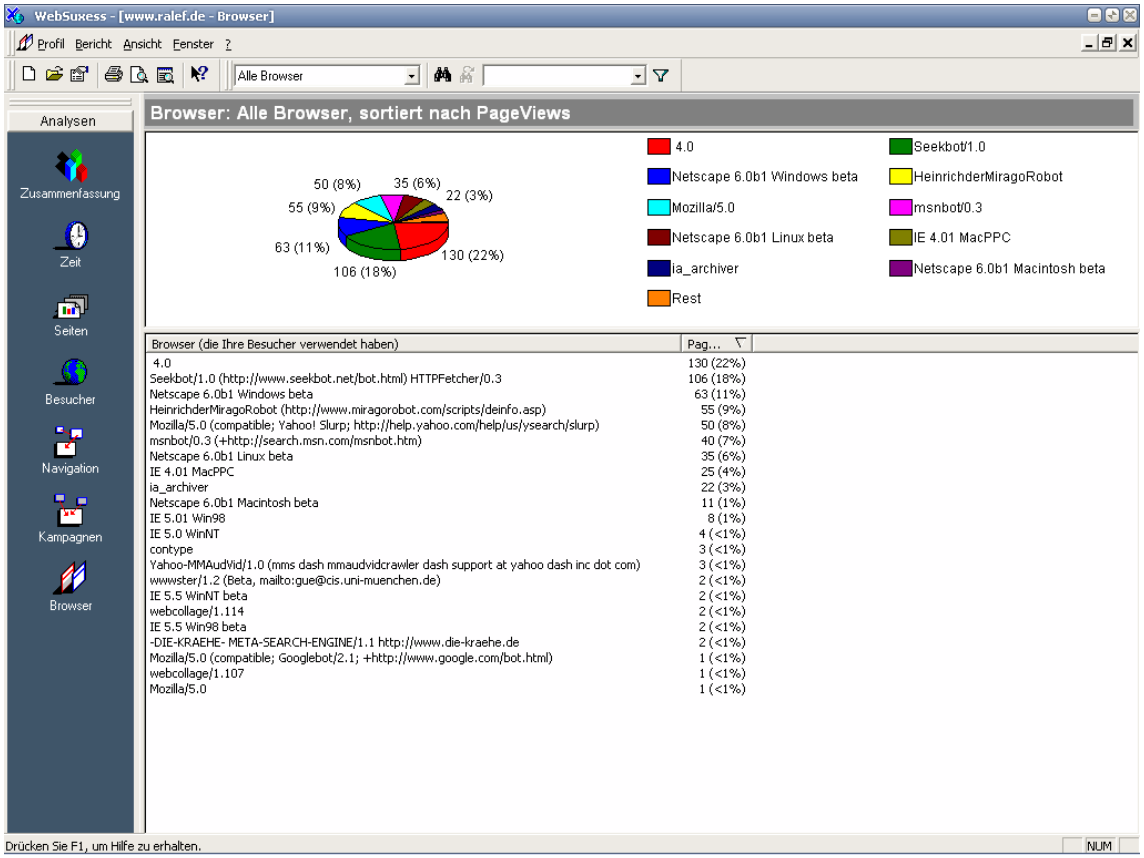


Abbildung 8: WebSuxess - Browser

6.2 Alternativen zum Logfile

Einfache Logfiles reichen oft nicht aus, um die Nutzung einer Homepage vollständig zu erfassen. Dies liegt an der Unvollständigkeit und Mehrdeutigkeit von Serverdaten. Es gibt einige alternative Ansätze zur Datenerfassung

6.2.1 Bilder

Soll die Aufzeichnung der Logfiles von einem anderen Server übernommen werden, benutzt man folgende Methode.

In das HTML-Dokument wird ein Bild eingebettet, diese Bild liegt jedoch auf einem anderen externen Webserver. Bei Abruf des HTML-Dokument, wird das Bild von dem 2. Webserver geladen und dieser externe Server loggt die Anforderung des Bildes mit. Das Bild wird entweder gut sichtbar dargestellt, als Hinweis auf den externen Statistikdienstleister, oder als 1 Pixel grosses transparentes GIF versteckt.

Dieses Verfahren ist dann besonders geeignet, wenn man keine Möglichkeit hat auf die Logfiles des Webservers zuzugreifen.

6.2.2 JavaScript

Durch JavaScript können auf der Client-Seite wesentlich mehr Informationen über dessen System gesammelt werden. Diese Daten können mit einem Aufruf an den Webserver übertragen werden.

Systemeinstellungen können erfasst werden, wie

- Bildschirmauflösung
- Farbtiefe des Bildschirms
- JavaScript-Version
- Installierte Plug-ins bei Netscape

Andere Begriffe für diese Technik: Client-Side Tagging, Client-Side Data Collection, Page Tagging, Page Beacons und Page Bugs.

7 Probleme

Bei der Logfileanalyse gibt es zwei grundsätzliche Fehlerformen:

- Bekannte Fehler, die man in Kauf nimmt
- Unerkannte Fehler, die die Statistik verfälschen

Unerkannte Fehler sind in ihrer Bedeutung nicht einschätzbar, deshalb konzentrieren wir uns auf bekannte Fehlerquellen.

7.1 Proxyserver

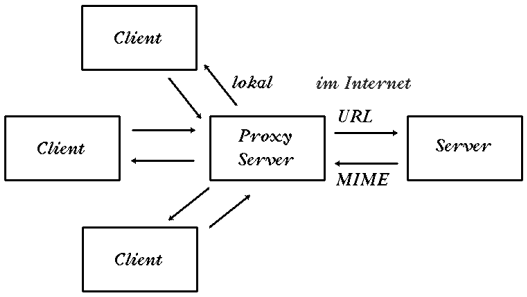


Abbildung 9: Client-Server Modell mit Proxy Server

Ein Proxy ist ein Rechner, der zwischen Client und Server geschaltet ist, siehe Abbildung 9. Beim Verbindungsaufbau vom Browser zu einem Webserver wird erst die Verbindung zum Proxy-Server aufgebaut. Diesem wird die die gewünschte Zieladresse übergeben. Der Proxy-Server baut anschliessend die Verbindung zum Webserver auf. Dabei erhält der Webserver keinerlei Informationen über den ursprünglichen Initiator der Verbindung. Gleichzeitig dient der Proxy-Server als Cache für gelesene bzw. aufgerufene Seiten.

Durch den Einsatz eines Proxyserver veringern sich insgesamt die Einträge im Logfile des Webservers, da die Dokumente auf dem Proxyserver zwischengespeichert werden. Es ist sehr schwierig abzuschätzen, wie stark ein Proxyserver die Statistik verfälscht.

8 Fazit

Hinter einem Logfile verbirgt sich eine Goldgrube an Informationen, die Kunst ist es, diesen Schatz zu finden. Es gibt mächtige Programme, die einem die Arbeit erleichtern, jedoch muss man sich intensiv mit der Thematik beschäftigen, um möglichst repräsentativ Ergebnisse zu gewinnen. Die Statistik ist ein außerordentliches mächtiges Werkzeug. Der traditionellen Log Analyse von Web Servern sind leider dank des zustandslosen HTTP-Protokolls enge Grenzen gesetzt, aber durch zusätzliche Web-Technologie lässt sich dieses Manko wett machen.

„Glaube nur der Statistik, die Du selbst gefälscht hast“(unbekannte Quelle)

Tabellenverzeichnis

Abbildungsverzeichnis

Literatur

[1] Krishnamurthy, B. ; Rexford, J.: *Web Protocols and Practice*, Addison-Wesley Professional, Boston, 2001

[2] Heindl, E.: *Logfiles richtig nutzen*, Galileo Computing, Bonn, 2003

[3] Baketaric B. ; Strübel, I.: *Auslegungssache, Webserver-Zugriffe richtig loggen und deuten*, c't 23/04, S.240