NAME

http-analyze – a real fast log analyzer for web servers

SYNOPSIS

http-analyze [**–{d|m|h}**] [**–nrstuvxz**] [**–c cfgfile**] [**–i logfile**] [**–o outdir**]
[**–p privdir**] [**–N #s|u**] [**–H homepage**] [**–S srvname**] [**–T title**] [**file**]

DESCRIPTION

http-analyze analyzes logfiles of web servers and creates detailed statistics of the servers's access load in graphical and tabular form. **http-analyze** expects logfiles entries in *common logfile format*, which is used by web servers such as Netscape's, NCSA's, and CERN's httpd. If your server uses another format, **http-analyze** can't read the logfile.

http-analyze has been highly optimized to process large logfiles at the maximum possible speed. This is achieved by using a history mechanism to skip logfile entries which have been processed already in a previous run of the program, and by using two modes of operation (named after their maximum useful update interval) with a different detail level in the analysis of the logfile entries:

daily mode (option **–d**):
: **http-analyze** generates a short summary showing the hits per day only. By using a history to skip entries processed already and by avoiding detailed analysis of each log entry, **http-analyze** requires only a fraction of the time needed for a full report.

monthly mode (option **–m**):
: In this mode, a full report with much more details is generated. The history is used to produce a summary for the last 12 month.

If your logfiles are rather large, you can use an update-interval in the range of one to 24 hours to generate a short statistics more frequently and an update-interval from one to 30 days to generate a full report. Since **http-analyze** maintains a history of the results from previous runs, you may rotate the logfile on a daily base when generating short (daily) reports. However, to generate a full (monthly) report you have to feed all logfiles of the appropriate summary period to **http-analyze** at once, because the program needs to do further analysis on all logfile entries. After generating a detailed report for a month, you can save the corresponding logfile(s) on tape and remove them from your system.

HTML OUTPUT FILES

In daily mode, **http-analyze** writes the short summary into the output file **stats.html** and updates the daily values in the history file. The short summary includes the following informations by day (see the following section for an explanation of this numbers):

- the total number of hits
- the total number of 304's (Not Modified responses)
- the total number of files transferred
- the total number of unique sites

- the amount of data sent by the server

In monthly mode, **http-analyze** updates the short summary in **stats.html** and the monthly values in the history file. Additionally, it creates the following files:

statsMMYY.html
: contains the detailed summary for the period determined by analyzing the logfile. *MM* and *YY* are replaced by the month and the year respectively.

filesMMYY.html
: lists the URLs of all documents sent by your server. This file is created by default, but you can suppress its creation with an option if you want to exclude them from the statistics.

sitesMMYY.html
: lists the hostnames of all sites accessing your server if the server could successfully resolve the IP address. Again, this file is created by default unless you explicitely suppress its creation.

statsYYYY.html or *index.html*
: contains a summary of the last 12 month. Which name is choosen depends on the date of the last logfile entry processed: If the last entry indicates that **http-analyze** is analyzing the current month's log, the name *index.html* is used for easy reference of the statistics pages. In all other cases the name *statsYYYY.html* is used. This naming convention allows you to create reports for previous summary periods (e.g. for last year) without affecting the results for the current period.

gr-icon.gif
: a small icon for your link to the statistics page (59x41 pixels).

All files are created in the current directory unless you explicitely specify an output directory for the HTML files. Furthermore, the files containing the detailed lists of sites and URLs may be created in a private directory to protect them by authorization.

The full summary (*statsYYMM.html*) contains the following informations:

- the total number of hits/304's/files/KB for this month
- the amount of data requested/transferred/saved by cache
- the total number of unique URLs/sites for this month
- the numbers of response codes other than 200 (OK) or 304 (NoMod)
- the maximum/average hits per day/hour
- the total number of hits/files/304's/sites/KB by day
- the top 5 seconds, 5 minutes, and 24 hours of the summary period
- the top 10 sites accessing your server most often
- the top 30 most commonly accessed URLs
- the last 10 frequently accessed URLs
- the hits/304's/KB sent by Country

The following section describes the meaning of all entries in the summary report, which are not self-explaining:

Hits
: (color key: green) The total number of hits processed by the server including requests which did generate an invalid response.

Files
: (color key: blue) The total number of files kind sent by the server (*OK* responses). Here "file" means any kind of file, thus including not only documents, but also images, CGI scripts, audio and video clips, etc.

304's
: (color key: yellow) A code 304 (*Not Modified*) response is sent by the server if a document hasn't been updated since the last time it was requested. This field therefore contains the total number of requests which didn't cause the transmission of a file because of various caching mechanisms used by proxies and browsers.

Other responses
: The total number of all answers from the server which are not *OK* (200) or *Not Modified* (304) responses. The full summary includes a list of all those other responses.

Unique URLs
: This field contains the total number of unique URLs (not counting erroneous requests).

Unique sites
: (color key: red) In the *Totals* section, this is the total number of unique sites per month, while in the *Hits by day* section it reflects the number of unique sites per day. Therefore, the sum of all sites shown in the "Hits by day" section is not equal to the total number of unique sites.

KBytes requested
: The amount of data requested by the users of your server. **http-analyze** computes this number by adding the values of the next two fields (see below).

KBytes transferred
: (color key: orange) The amount of data sent as reported by the server.

KBytes saved by cache
: The amount of data saved by various caching mechanisms. It's value is computed by multiplying the number of Not Modified requests per page with the size of the document (if known). Note: Because **http-analyze** can determine the size of a page only if the page has been requested successfully at least once in the same summary period, the values for "KB saved by cache" and "KB requested" are just approximations of the real values.

OPTIONS

–h print a short help list explaining the usage of the options.

–d *(daily mode)* generate short statistics for the current month only. If a history file exists, the values for previous days are read from this file and the corresponding logfile entries are skipped. If the history file does not exist, the whole logfile will be processed and a history will be created. (This option is set by default.)

–m *(monthly mode)* generate full statistics for a whole month. Although the values from the history file are usually used to create a summary for the last 12 month, the actual logfile entries always have preceedence over any records in the history file. This means that you should rotate your logfile at least on a monthly base. The option **–m** includes **–d**.

–n *(no update)* don't create or update the history file. Useful if you want to generate statistics for previous summary periods (before the last month) without overwriting the current state of the history.

–r don't create a list of all URLs for hidden items (if any) in the full statistics.

–s *(no sitelist)* don't create a list of all sites in the full statistics.

–t *(no TOP lists)* don't create the top seconds/minutes/hours lists. Also suppresses the "Hits by hours" bar chart.

–u *(no URL list)* don't create a list of all requested URLs in the full statistics.

–v *(verbose)* comment ongoing processing.

–x Don't comprise images by default. Normally, **http-analyze** sums up the values of all images (**.gif, *.jpg, *.ief, *.pcd, *.rgb, *.xbm, *.xpm, *.xwd, *.tif*) and hides them under the item "All images" to avoid getting the top lists filled up with lots of image URLs. If **–x** is given, images are accounted for as single items.

–z don't create graphical representations of the results.

–c *cfgfile*
Use *cfgfile* as the configuration file. By using a config file, **http-analyze** allows you to define some options and to tailor the basic HTML page layout somewhat. See "CONFIGURATION FILE" below for a description of the config file format.

–i *logfile*
Use *logfile* as the server's logfile. If '-' is given, *stdin* is processed. See also the **HTTPLogFile** entry in the config file.

–o *outdir*
This is the name of the directory where the HTML output files should be created. If no directory is given, the files are created in the current directory. See also the **HTMLDir** entry in the config file.

–p *privdir*
Use this directory for the list of all URLs/sites *(filesMMYY.html and sitesMMYY.html)* . This is useful if you want to grant public access to your web server's statistics while permitting access to the detailed lists to the staff only by using server authentication. See also the **PrivateDir** entry in the config file.

–H *homepage*

Use *homepage* as an alternate name for homepages. If your index files are named **index.html**, there is no need to define this option. However, if your server looks for more than one filename (eg. **index.html**,**Welcome.html**, and **home.html**, you must define the latter two explicitely. **http-analyze** truncates the URLs containing a homepage name so that they merge with '/' or their "base URL", respectively. (For example, the "base URL" for */dir/index.html is /dir/ .)* You can define up to three alternate names in addition to **index.html**. See also the **Homepage** entry in the config file.

–N *#{***sul***}*

This option defines the number of entries in the top site (**s** or **S**), top URL (**u** or **U**), or last URL (**l** or **L**) list. *#* is either a positive number or the value 0 to suppress the corresponding list. Note that the list of last frequently accessed URLs is generated only if the number of all unique URLs is greater than the sum of the entries in the top and last URL lists. See also the entries **TopSites**, **TopURLs**, and **LastURLs** in the config file.

–S *srvname*

Use *srvname* as the name of the server in the title of the HTML files. If undefined, **http-analyze** tries to determine the server name itself. Note: **http-analyze** uses either the *uname (2)* or the *gethostname (2)* function to determine the server name depending on what has been defined at compilation time. On most System V implementations, *uname* returns the nodename (eg. *host*), while *gethostname* often returns the full qualified domain name (FQDN, eg. *host.my.domain*). See also the **ServerName** entry in the config file.

–T *title* Use *title* as the document title and header for the HTML files. **http-analyze** appends the server name and the current summary period to this string. If left undefined, a default phrase is used. See also the **DocTitle** entry in the config file.

CONFIGURATION FILE

When specified with the option **–c**, **http-analyze** reads some defaults from the named configuration file. Parameters defined with options always take preceedence over the definitions in this configuration file. The configuration file contains one entry per line. Each entry has a name field and one or two value fields, which must be separated by one or more tabulator characters (not blanks!). All names are case-insensitive.

ServerName The name of your server (same as option **–S**).

HTTPLogFile The name of the server's logfile. Note that if you define a default name of the logfile, this file gets processed if no other file is explicitely defined at the invocation of **http-analyze**. Without this definition, **http-analyze** processes *stdin* if no file is given. To process *stdin* even if a default name has been defined, use '-' as the filename for the logfile.

DefaultMode Defines the default operation mode of **http-analyze**. The value field contains either the keyword **daily** or **monthly**. If left undefined, the default is the daily mode (**–d**).

Homepage Up to three alternate names for homepages in addition to **index.html** (same as option **–H**). All URLs containing one of the homepage names will get truncated so they merge with '/' or the base URL respectively.

HTMLDir The name of the directory where the HTML output files should be created (same as **–o**). If left undefined, files are created in the current directory.

PrivateDir The name of a private directory where the detailed site and URL lists should be created (same as option **–p**). Access to this private directory may be granted to staff only by using server authentication. Pathnames not beginning with a '/' are relative to **HTMLDir**.

TopSites, **TopURLs**, and **LastURLs**
The number of entries in the top site, top URLs, and last frequently used URLs lists (same as option **–N).** If set to zero, the corresponding list will be suppressed.

DocTitle The document title and header to use in the HTML output files (same as option **–T**). **http-analyze** appends the server's name and the current summary period to this string.

HeadPrefix The prefix string to output before the document header (after the HTML <TITLE> tag). If **HeadPrefix** is defined, it must include the HTML <BODY> tag. If left undefined, **HeadPrefix** defaults to:

```
HeadPrefix       <BODY BGCOLOR="#D6D6D6"><P><HR SIZE="8">
```

HeadSuffix The suffix string to output after the document header (after **DocTitle**). Useful if you define left- or right-aligned images in **HeadPrefix** with the headline floating around.

DocTrailer The trailer string to output at end of page. Useful to define a link back to your homepage, as in

```
DocTrailer       <BR><FONT SIZE="-1"><A HREF="/">Back</A> to my homepage</FONT>
```

HideSys and **HideURL**
These two entries let you define names of sites or URLs which should be hidden under some arbitrary text. Hidden items are accounted for separately, but in the summary they appear comprised under the description defined here. Both entries have two value fields: the first field following the name defines a site or an URL and the second field defines the text under which this item is to be hidden. The URL/site may begin or end with a '*' as a wildcard. However, inside strings, a '*' is taken literal. If the text a item is hidden under begins with a '[' character, the item is not shown in the top sites/URLs lists, but it will be always shown in the detailed sites/URLs lists. Note that URLs are case-sensitive, while sitenames

are not. Note also, that images are hidden automatically unless the option **–x** is specified at invocation of **http-analyze**. See the **sample.conf** file for examples on how to use **HideSys** and **HideURL**.

EXAMPLES

First of all, you must know the name of your server's logfile. If, for example, the name is */usr/ns-home/httpd-80/logs/access*, you can create full statistics for the current month with the following command:

```
http-analyze -vm -S www.myserver.com /usr/ns-home/httpd-80/logs/access
```

This command will create a yearly summary in the file *index.html* (or *statsYYYY.html* for previous years) and a monthly summary in file *statsMMYY.html*, where *MM* is replaced by the month and *YY* is replaced by the year. If the period determined by analyzing the logfile is the current month, **http-analyze** creates also an up-to-date daily summary in the file *stats.html*. All files are created in the current directory.

Assuming that your old logfiles have been saved under the name *logYYYY/access.MM* in the server's log directory, use the commands

```
cd /usr/ns-home/httpd-80/logs
http-analyze -vmn -o /usr/htdocs/stats log1996/access.01
```

to create full statistics for January '96 in the directory */usr/htdocs/stats* preserving the current history (option **–n**). Note: Generating statistics for previous summary periods without the **–n** option will overwrite newer values in the history file. To reconstruct the history, you would have to run **http-analyze** for each following month until the very last one (this situation may be avoided in a following version of the program). Note also, that immediately after generating the statistics for the last month you should run **http-analyze –m** on the current logfile to create an up-to-date index file (**index.html**). Remember that this index file is created automatically only when creating a monthly summary for the **current** month.

The following command creates statistics for a whole year using a customized configuration file and reading the log entries from a pipe:

```
gzcat log1996/access.0?.gz |
http-analyze -vm -c /usr/local/bin/sample.conf -
```

REGULAR INVOCATION VIA CRON

To have statistics generated on a regular base, use the following scheme:

1) Optionally install a cron job which calls **http-analyze –d** frequently to create a daily summary. The execution interval may range from once per day up to twice per hour depending on the size of your logfile and the time needed to analyze it. On my server, I run the daily statistics once per hour.

2) Install a cron job which calls **http-analyze –m** to create a monthly summary once per week or once per day (again depending on the size of your logfile). Note that monthly summaries *(statsMMYY.html)* are created for the first time at the second day of a new month. On my server, I create a monthly

summary two times per day.

3) Create a script which rotates the server's logfile, restarts the http server, and creates the final summary for this period. Have *cron* execute this script at 00:00 on the first day of a new month. See the script **rotate-httpd** for an example on how to do this for several virtual web servers running on the same machine.

4) Because of *cron*'s scheduling overhead and delays in execution of the script which rotates the logfile, heavy used servers sometimes writes a few entries for the new month in the old logfile. **http-analyze** usually ignores such kind of "white noise" at the end of a month. However, to get correct figures, in this last step you should run **http-analyze −m** on the logfile for the current month immediately after generating the statistics for the previous month.

Note that the cron jobs must run with the uid of the owner of the directory where the HTML output files are going to be created, except for the rotate script, which usually must run with the uid of the Server. You should also take care to avoid running more than one of the cron jobs related to **http-analyze** at the same time.

Here are some sample *crontab*(1) entries for the scheme described above:

```
# Generate a full report twice per day at 01:17 and 13:17
17  1,13 * * *  /usr/local/bin/http-analyze -m -c /usr/httpd/analyze.conf

# Generate a short summary each hour except at 01:17 or 13:17
17  2-12 * * *  /usr/local/bin/http-analyze -d -c /usr/httpd/analyze.conf
17 14-23 * * *  /usr/local/bin/http-analyze -d -c /usr/httpd/analyze.conf

# Rotate the HTTPD logfiles at the first day, 00:00 of a new month
0 0 1 * *       /usr/local/bin/rotate-httpd
```

COPYRIGHT

Copyright © 1996 by Stefan Stapelberg, RENT-A-GURU®

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and in all HTML output files, that both that copyright notice and this permission notice appear in the supporting documentation, and that the hypertext link to the homepage of **http-analyze** which the program produces is left intact. This software is provided "as is" without express or implied warranty.

Credit for **http-analyze** must be given to RENT-A-GURU® in all derived works. This does not affect your ownership of the derived work itself, and the intent is to assure proper credit for RENT-A-GURU®, not to interfere with your use of this software. If you have questions, ask.

You may use this software at no cost on any installation, even at commercial sites. However, **IT IS STRICTLY FORBIDDEN** to sell or lease this software in whole or in part or to include it in whole or in part in a commercial product. If you plan to run **http-analyze** on a commercial installation **and** you need support, or if you would like to bundle the program with your products, you must sign an appropriate license agreement available from RENT-A-GURU®. Please send an email to <office@rent-a-guru.de>.

RENT-A-GURU® is a registered trademark of Martin Weitzel, Stefan Stapelberg, and Walter Mecky.

AUTHOR

Stefan Stapelberg, <stefan@rent-a-guru.de>

CREDITS

Thanks to the over 50 beta testers of **http-analyzes** for their feedback.
Special thanks to <Lars-Owe.Ivarsson@its.uu.se> for his suggestions to optimize the parser algorithm and the code he provided as an example.
Thanks also to Thomas Boutell (http://www.boutell.com) for his great GD library for fast GIF creation, without **http-analyze** couldn't produce such fancy graphics in the summary reports (gd 1.2 is copyright 1994, 1995, Quest Protein Database Center, Cold Spring Harbor Labs).

FILES

Note: output files are always created in the directory given with the **–o** option, with the **HTMLDir** entry in the config file, or in the current directory (in this order). See also HTML OUTPUT FILES above.

index.html,	summary report for last 12 month
statsYYYY.html	summary report for year *YYYY*
stats.html	short summary (daily mode)
statsMMYY.html	full summary for *MM/YY* (monthly mode)
filesMMYY.html	list of all URLs requested in *MM/YY*
sitesMMYY.html	list of all sites accessing the server in *MM/YY*
stats.hist	the history file for the last 12 month and last *N* days
avloadMMYY.gif	the *Hits by hours* bar chart image (492x190)
statsMMYY.gif	the *Hits/Files/Sites/KB* by day bar chart image (492x317)
cntryMMYY.gif	the *Total transfers by Country* pie chart image (492x320)
graphMMYY.gif	the *Hits/Files/Sites/KB* graph image (490x317)
sq_*.gif.gif	icons for creating bars in the full summary (10x8)
gr-icon.gif	an icon for making links to your statistics page (59x41)

NOTES

If you are going to analyze different logfiles in one invocation of **http-analyze**, you must sort them in ascending order of their date, otherwise the logfiles being processed after the first logfile will be silently ignored.

SEE ALSO

3Dstats(8L)	A 3D Access Statistics Generator
http://www.netstore.de/Supply/http-analyze/	The homepage of **http-analyze**

BUGS

You tell me.