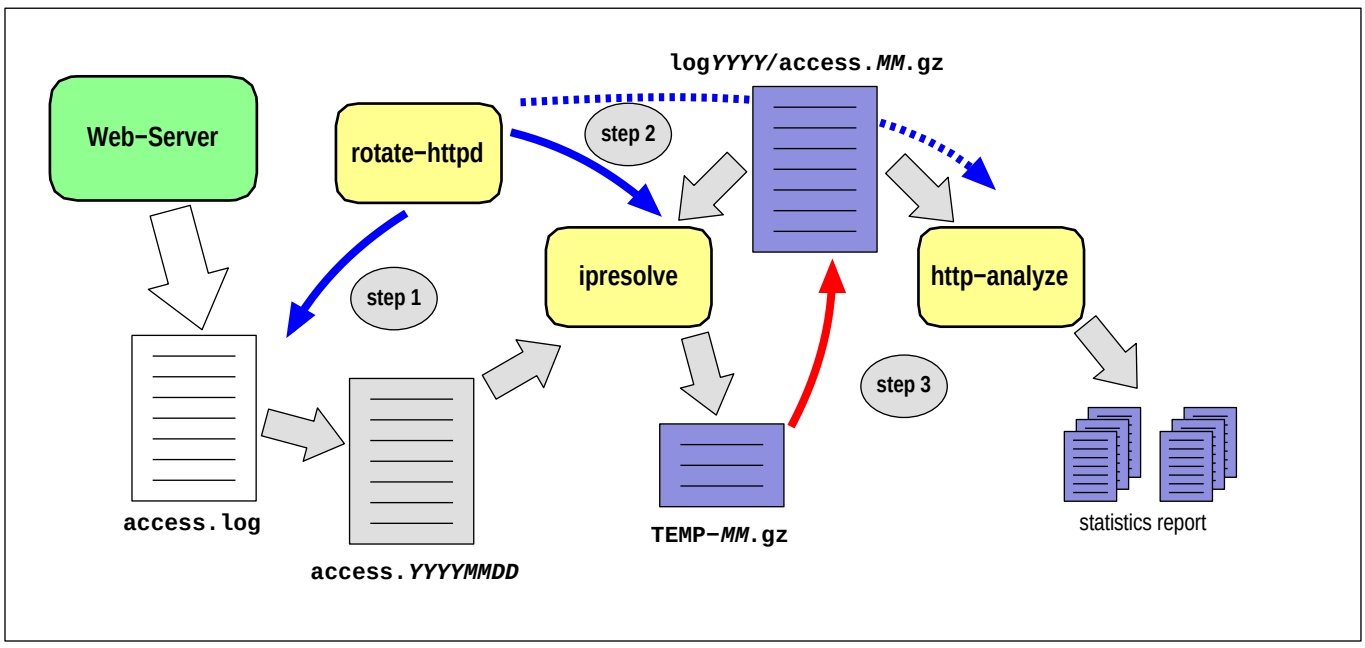
Technical Report 03-2003-09-08

Daily rotation and compression of logfiles with http-analyze 2.5 and ipresolve 2.0

Starting with version 2.5 of **http-analyze** and version 2.0 of **ipresolve**, both programs can read - and **ipresolve** also write - *gzip*'ed logfile data. This means that with a clever rotation scheme the logfiles of a web server can be rotated once per day and can be saved in a space-conserving manner. This technical report explains how to set up such a scheme with a shell script **rotate-httpd** on a system running the Netscape Fasttrack server, the Apache server or any other web server. **rotate-httpd** uses **ipresolve** to resolve IP numbers into hostnames and **http-analyze** to create a statistics report for the server.

The shortest time-period needed by **http-analyze** to create a full statistics report is one month. To rotate logfiles on a daily base, you need to save all old logfiles from the 1st of the current month until this month has been »finalized«. To save disk space, old logfile data can be saved in *gzip*'ed format.
The script **rotate-httpd** is responsible for saving the logfile of a web server and create a statistics report. It is started automatically by *cron(8)* at 00:00. First, **rotate-httpd** renames the web server's logfile **access** to **access.***YYYYMMDD*, where *YYYYMMDD* is the date of the previous day. This temporary file is still in raw (uncompressed) format (step 1 in the figure below).

If it is a month wrap, **rotate-httpd** also saves the file **errors**. The script then informs the web server to force creation of a new logfile.
After all logfiles have been saved, the script delays for 20 minutes to not disturb other *cron(8)* jobs, which are started at midnight. Then it starts **ipresolve**, which resolves IP numbers into hostnames by reading the saved logfile for the month (**access.***MM***.gz**) and the logfile for the previous day (**access.***YYYYMMDD*). It creates an output file **TEMP-***MM***.gz**, which is in fact the old logfile plus the logfile data from previous day. This way, only unresolved IP numbers from the previous day have to be resolved, which reduces the load from the DNS server by a significant amount (step 2):



If the old logfile and the logfile for the previous day could be combined sucessfully, **rotate-httpd** calls **http-analyze** to process this logfile (step 3).

This rotation scheme depends on consistent naming conventions if you want to fully automate it for virtual hosts or several web servers. For example, on an SGI with Netscape Fasttrack the server root is **/var/netscape/fasttrack/httpd-***sitename*. The logfiles are in a subdirectory **logs** and are named **access** and **errors**. On a FreeBSD system using the Apache server, the logfiles might be under **/usr/local/apache/vhosts/***sitename***/logs** and are named **access_log** and **errors_log**.

The important point here is that you can distinguish the servers you are going to analyze by their name, ideally in a way to pass the name to **http-analyze** using the **-S** option, which defines the name of the server shown in the statistics report. Of course you can also specify the name of an individual configuration file for this web server to tailor its look & feel of the statistics report.

The name of the server's root is also used to construct the name of the directory for the statistics report (usually under the document root of the server).

Continued on page 2

The script **rotate−httpd** is a **ksh**−script. It first defines the pathnames for the executables, the server root and the names of the logfiles. Next, the three variables **DAY**, **MON** and **YEAR** with appropriate field width are defined. The variable **MWRAP** is set on first day of a new month. **ECHO** is used to toggle the display of debug messages.

```
#!/bin/ksh
#
# Rotates the server's logfiles on a daily base, resolve IP numbers
# and archive the resulting data in a file in gzip'ed format.
#
USAGE="$(basename $0) [−hev]"

HA_CMD=/usr/local/bin/http−analyze      # The script to analyze the logfile
HA_OPTS="−3fm"                          # default options

IPRES_CMD=/usr/local/bin/ipresolve      # The command to resolve IP numbers
IPRES_OPTS="−d /var/tmp/DNS−data"       # default options

# SERVERROOT contains all configuration files
SERVERROOT="/var/netscape/fasttrack"

# LOGFILE contains the name of the logfile (rotated daily at midnight)
# ERRFILE contains the name of the error file (rotated once per month)
LOGFILE="access" ERRFILE="errors"

integer DAY MON YEAR
typeset −Z2 DAY MON
typeset −Z4 YEAR

MWRAP="" ECHO=": "
```

The script now determines the current date and computes the date of the previous day using shell arithmetic and the UNIX utility *cal(1)*:

```
# Get current date and the month's name.
# Compute the date for the previous day.
eval $(date "+MNAME='%B' DAY='%d' MON='%m' YEAR='%Y'")

((DAY=$DAY−1))                  # previous day
if [ "$DAY" −eq 0 ]; then       # month wrap
        ECHO="echo"             # be verbose
        MWRAP=true              # remember month wrap
        ((MON=$MON−1))          # previous month
                                # year wrap
        [ $MON −eq 0 ] && { MON="12"; ((YEAR=$YEAR−1)); }

        # compute day of last month at mont or year wrap
        for DAY in $(cal $MON $YEAR); do
            : nothing − upon exit DAY contains last day of old month
        done
fi
```

Now it constructs the names of the logfiles, changes into the server root, renames all logfiles and restarts the server to force creation of a new logfile. If there is no subdirectory **logs**, this server is skipped intentionally:

```
# The names of the logfiles are constructed using the year, month and day:
#
#  LOGDIR:      logYYYY                 where YYYY is the year
#  LOGTMP:      $LOGFILE.YYYYMMDD       where YYYY is the year, MM is the
#                                       month and DD is the day
LOGDIR="log$YEAR"
LOGTMP="access.$YEAR$MON$DAY"

# First step: rotate the logfiles as fast as possible and
# inform the server of the change.
cd $SERVERROOT || { echo "panic: can't cd into $SERVERROOT" 1>&2; exit 1; }

for server in httpd−*; do
        if [ ! −d $server/logs ]; then
                $ECHO "Skipping $server − no log directory found" 1>&2
                continue
        fi
        (cd $server/logs
            [ ! −d "$LOGDIR" ] && mkdir $LOGDIR
            [ −f $LOGFILE ] && mv $LOGFILE $LOGTMP
            [ −n "$MWRAP" −a −f $ERRFILE ] && { \
                mv $ERRFILE $LOGDIR/$ERRFILE.$MON && gzip −best $LOGDIR/$ERRFILE.$MON;\
                $ECHO "$server/logs/$ERRFILE saved in $LOGDIR/$ERRFILE.$MON.gz" 1>&2; }
        )
        $server/restart
done
```

Continued on page 3

In the second step, **rotate-httpd** runs **ipresolve** to resolve IP numbers into hostnames. **ipresolve** reads the old logfile for the current month (in *gzip*'ed format) and the saved logfile for the previous day (in raw format). It creates an output file in *gzip*'ed format, which then replaces the old logfile. To reduce DNS queries, **ipresolve** uses a DBM database to store IP/hostname pairs. Note that with huge logfiles resolving IP numbers might be a very time-consuming process, especially if you don't use **ipresolve**'s DBM-based cache.

```
# Second step: wait 20 minutes to not disturbe other cron jobs
# starting at 00:00, then resolve IP numbers into hostnames and
# run the analyzer to "finalize" the statistics for the previous
# (old) month. During IP resolving we are combining the new and
# the old logfile into one final logfile. The name of this final
# logfile is $LOGFILE.MM.gz.
sleep 1200

for server in httpd-*; do
        if [ ! -d $server/logs ]; then continue; fi
        (cd $server/logs
         if [ -f $LOGDIR/$LOGFILE.$MON.gz ]; then
                ALLFILES="$LOGDIR/$LOGFILE.$MON.gz $LOGTMP"
         else   ALLFILES="$LOGTMP"
         fi
         if $IPRES_CMD $IPRES_OPTS -o TEMP-$MON.gz $ALLFILES; then
                if mv TEMP-$MON.gz $LOGDIR/$LOGFILE.$MON.gz; then
                   rm -f $LOGTMP
                   $ECHO "$server/logs/$LOGFILE saved in $LOGDIR/$LOGFILE.$MON.gz" 1>&2
                fi
         fi)
done
```

The third and last step is to analyze the old logfile and to update the statistics report. The name of the server is determined using shell pattern matching. Instead of using just the **-S** option for defining the server's name, an individual configuration file (option **-c**) could be used to tailor the output of **http-analyze** on a per-server base.

```
# Third and last step: update the statistics report
for server in httpd-*; do
        if [ ! -d $server/logs ]; then continue; fi
        (cd $server;
         SRVNAME=
         $HA_CMD $HA_OPTS -S ${server#httpd-} -o stats logs/$LOGDIR/$LOGFILE.$MON.gz)
done
```

Users of the **run-ha** script (TR-01-2003-09-08) can run this directly instead of **http-analyze** in a for loop:

```
# Third and last step: update the statistics report
/usr/local/bin/run-ha -m $MON $YEAR
```

On a month wrap, the statistics report is »finalized«, this means that the logfile is being analyzed a last time automatically. Since **http-analyze** creates a new statistics not before the 2nd day of a new month, it makes not much sense to include the current (new) logfile in this run. However, it *could* make sense to run **http-analyze** independently later at the day to get real-time figures for that day (2nd to 31th of a month).

Upon exit, the **rotate-httpd** script cleans the DBM database of **ipresolve** once per month:

```
# Clean the DNS database from entries older than 32 days.
# Since we save resolved data, the database helps only to
# speed up queries for IP numbers found in the logfile
# for the current month.
[ -n "$MWRAP" ] && $IPRES_CMD $IPRES_OPTS -c "32 days"
exit 0
```

To activate the script, place an entry like the following in the *crontab* of the server user (do NOT use **root**'s crontab). See the man-pages of your system for more information about the *crontab(1)* command, *crontab(5)* entries and *cron(8)* jobs.

```
# Rotate HTTPD logfiles once per month
0 0 * * *  /usr/local/bin/rotate-httpd
```

☞ Make sure that the selected user for this crontab has sufficient permission to save logfiles and to restart the web server.

Continued on page 4

Finally there are some tips and tricks for customization of **rotate-httpd**:

- To extract the list of virtual hosts from an Apache configuration file, use:

```
APACHE_CFG=/usr/local/etc/httpd/conf/httpd.conf
SERVERROOT=/www/vhosts
CUSTOMLIST=‘sed -n ’s/^ServerName[      ][      ]*\(.*\)/\1/p’ $APACHE_CFG‘
```

  (There are a tabulator and a space character inside the square brackets in the **sed** command.)

- To restart the Apache server for one or all virtual hosts, use the appropriate script or executable:

```
APACHE_RST=/usr/local/bin/restart_apache
```

  or:

```
APACHE_RST="/usr/local/bin/apachectl restart"
```

- To replace **ksh**-syntax for old-fashioned shell arithmetic, use *expr*:

```
MON=‘expr "$MON" - 1‘
if [ "$MON" -eq 0 ]; then MON="12"; YEAR=‘expr "$YEAR" - 1‘; fi
```

- To replace **ksh**-syntax **$(**...**)** for command substitution, use back-quotes:

```
for DAY in ‘cal $MON $YEAR‘; do
    : nothing - upon exit DAY contains last day of old month
done
```

- To expand single digits for the first nine days of a new month into two digits, use either:

```
expr "$MON" : ’..’ >/dev/null 2>&1 || MON="0$MON"
```

  or:

```
if [ "$MON" -lt 10 ]; then MON="0$MON"; fi
```

- To use **rotate-httpd** with **ipresolve** 2.0 and **http-analyze** 2.4, use *gzip(1)/gzcat(1)* to uncompress the logfiles:

```
gzip -dc logs/$LOGDIR/$LOGFILE.$MON.gz) | http-analyze -3fm -o stats -
```

- To have **http-analyze** create new configuration files for all servers, execute the following command once:

```
for server in http-*; do
        http-analyze -i $server/http-analyze.conf -S ${server#httpd-}
done
```

  This creates the configuration file **http-analyze.conf** in the server's root directory. The **-S** option initializes the **ServerName** directive in the new configuration file. If you want to convert old configuration files, specify them using the **-c** option. Then, add the following to **rotate-httpd**:

```
if [ -f $HA_CONFNAME ]; then
    $HA_CMD $HA_OPTS -c $HA_CONFIG -o stats logs/$LOGDIR/$LOGFILE.$MON.gz
else
    $HA_CMD $HA_OPTS -S ${server#httpd-} -o stats logs/$LOGDIR/$LOGFILE.$MON.gz
fi
```

run-ha is included in **http-analyze** since version 2.4.
rotate-httpd is included in **http-analyze** since version 2.4.
ipresolve 2.0 is available through our Customer Support site and will shortly become available to everyone.

Please send comments, enhancements, tips and tricks to: **office@rent-a-guru.de**.