Technical Report 02−2003−09−08

A shell script to rotate the logfiles of web servers

In order to save disk space and processing time, logfiles of servers should be rotated (backed up) on a regular base. If you use **http−analyze** to create a statistics report, logfiles must be kept intact for the whole current month until it can be »finalized« (processed a last time) in the following month. The time by which a statistics report can be finally created without having changes is at 00:00 at the 1st day of the following month. This report explains **rotate−httpd** in its basic version. **rotate−httpd** is a shell script, which saves the logfile of all web servers and runs **http−analyze** to finalize the statistics of the previous month. **rotate−httpd** is called through the *cron(8)* service on a regular base.

The first version of **run−ha** is a **ksh**−script. The server root is **/var/netscape/fasttrack/httpd−***sitename*. The logfiles are in subdirectory **logs** and are called **access** and **errors**. The saved logfiles will go to a subdirectory **log***YYYY* and will be called **access.***MM***.gz**. The script first gets the date and computes the last month. It then changes into **SERVERROOT** and saves the logfiles for each server in the appropriate subdirectory. The script then restarts the web server and delays 20 minutes to not disturb other jobs running at this time before it starts resolving IP numbers into hostnames.

```
#!/bin/ksh
#
# Rotates server log files and archives the old logfiles.
# Does not affect users who may be connected to the server.
#

# The script to analyze the logfile
RUNHA_CMD=/usr/local/bin/run-ha

# The script to resolve IP numbers
IPRES_CMD=/usr/local/bin/ipresolve
IPRES_OPTS="-d /var/tmp/DNS-data"

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

# Rotate these logfiles
LOGS='access errors'

integer MON YEAR
typeset -Z2 MON
typeset -Z4 YEAR

eval $(date "+MABB='%B' MON='%m' YEAR='%Y'")

((MON=$MON-1)); [ $MON -eq 0 ] && { MON="12"; ((YEAR=$YEAR-1)); }
LOGDIR="log$YEAR"

cd $SERVERROOT || { echo "panic: can't cd into $SERVER_ROOT" 1>&2; exit 1; }

# Rotate all logfiles, inform the server of the chance
for server in httpd-*; do
    (cd $server/logs
        [ ! -d "$LOGDIR" ] && mkdir $LOGDIR
        for file in $LOGS; do
            [ -f $file ] && { \
              mv $file $LOGDIR/$file.$MON; \
              echo "$server/logs/$file saved in $LOGDIR/$file.$MON" 1>&2; }
        done)
    $server/restart
done

# 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.
sleep 1200
```

Continued on page 2

The script **rotate-httpd** will then run **ipresolve** to resolve IP numbers into hostnames and produce a *gzip*'ed summary (note that this requires **ipresolve** version 2.0 or above). After resolving IPs the script runs **http-analyze** on the old logfile. Upon exit, the DBM database of ipresolve is cleaned.

```
#
# Now we resolve the logfile and produce a gzip’ed summary.
# This usually takes a long time and just in case we need
# the logfiles more than once, we save the resulting data
# with all resolved names.
for server in httpd-*; do
        (cd $server/logs/$LOGDIR
            $IPRES_CMD $IPRES_OPTS -o access.$MON.gz access.$MON && \
                rm -f access.$MON)
done

# Now run http-analyze through run-ha
$RUNHA_CMD $MON $YEAR

# Clean the DNS database from entries older than 90 days.
# Since we save resolved data, the database helps only to
# speed up queries for IP numbers found in the actual
# (current) logfile. However, it helps to speed up the
# resolving process for the current month, so we keep
# last 31 days of data.

$IPRES_CMD $IPRES_OPTS -c 32days
exit 0
```

The second version of **rotate-httpd** is a **bash**-script, which automatically determines the list of virtual hosts to process from the Apache configuration file. The server root is **/www/vhosts/***sitename*. The script looks for a directory **www.***sitename* and *sitename*. It then looks for a **logs** subdirectory and skips the server if it finds none. Otherwise it moves the logfile **access_log** to **log***YYYY***/access_log.***MM*:

```
#!/usr/bin/bash
#
# Rotates server log files and archives the old logfiles.
#
USAGE="$(basename $0) [-v]"

APACHE_CFG=/usr/local/etc/httpd/conf/httpd.conf
APACHE_RESTART=/usr/local//bin/restart_apache
RUNHA_CMD=/usr/local/bin/run-ha
ECHO=": "

# SERVERROOT contains all configuration files
# SERVERLIST is a list of all servers to analyze
SERVERROOT=/www/vhosts
SERVERLIST=‘sed -n ’s/^ServerName[      ][      ]*\(.*\)/\1/p’ $APACHE_CFG‘

# Rotate these logfiles
LOGS=’access_log’

eval $(date "+MABB=’%B’ MON=’%m’ YEAR=’%Y’")

MON=‘expr "$MON" - 1‘
if [ "$MON" -eq 0 ]; then MON="12"; YEAR=‘expr "$YEAR" - 1‘; fi
expr "$MON" : ’..’ >/dev/null 2>&1 || MON="0$MON"
LOGDIR="log$YEAR"

while [ $# -gt 0 ]; do
   case $1 in
    -h)   echo "Usage: $USAGE"; exit 0 ;;
    -v)   ECHO=echo; RUNHA_OPTS=-e ;;
     *)   echo "Invalid parameter: $1\nUsage: $USAGE" 1>&2; exit 1 ;;
   esac
   shift
done

cd $SERVERROOT || { echo "panic: can’t cd into $SERVERROOT" 1>&2; exit 1; }

for server in $SERVERLIST; do
        if [ -d "www.$server" ]; then server="www.$server"; fi

        # absence of logs directory means skip this vhost
        if [ ! -d "$server/logs" ]; then continue; fi

        (cd $server/logs
            [ ! -d "$LOGDIR" ] && mkdir $LOGDIR
            [ -f access_log ] && { \
               mv access_log $LOGDIR/access_log.$MON; \
               $ECHO "$server/logs/access_log saved in $LOGDIR/access_log.$MON" 1>&2; }
       )
done
```

Continued on page 3

Finally, the Apache server is restarted for all web servers, a statistics report is created (using **run-ha**) for the previous month and the old logfile is compressed using the *gzip(1)* command:

```
# Now restart the web server
$APACHE_RESTART

# Then analyze the saved and new logfiles
$RUNHA_CMD $RUNHA_OPTS $MON $YEAR

# Compress the old logfiles
for server in $SERVERLIST; do
        if [ -d "www.$server" ]; then server="www.$server"; fi

        # absence of logs directory means skip this vhost
        if [ ! -d "$server/logs" ]; then continue; fi

       # otherwise compress the old logfile
        (cd $server/logs/$LOGDIR; gzip -best access_log.$MON)
done
exit 0
```

To install **rotate-httpd**, place an entry like the following in the crontab of the server user (do **NOT** use **`root`**'s crontab!) and have it executed on 00:00 on the 1st day of a new month:

```
# crontab file for http-analyze web-server statistics
#
# Format of lines:
# min hour daymo month daywk cmd

# Rotate the logfiles once per month
0  0    1    * *  /usr/local/bin/rotate-httpd

# Update the statistics once per day
17 1,13 2-31 * *  /usr/local/bin/run-ha
```

☞ Make sure that the selected user for this crontab has sufficient permission to save logfiles and to restart the web server. Modify the entry to call **run-ha** or **http-analyze** (if any) only from the 2nd to the 31th day of a month!

The script can be customized to almost all web server installations. Some things to keep in mind are:

❏ Always have **http-analyze** process all logfiles for the current month in full statistics mode, otherwise the statistics can get zeroed.

❏ If you use **http-analyze** 2.5 or **ipresolve** 2.0, they can both read *gzip*'ed logfiles directly.

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`**.