## #84 Exploring the Apache access_log

If you're running Apache or a similar web server that uses the Common Log Format, there's quite a bit of quick statistical analysis that can be done with a shell script. The standard configuration for a server has an `access_log` and `error_log` written for the site; even ISPs make these raw data files available to customers, but if you've got your own server, you should definitely have and be archiving this valuable information.

Table 10-1 lists the columns in an `access_log`.

| Column | Value |
|---|---|
| 1 | IP of host accessing the server |
| 2–3 | Security information for https/SSL connections |
| 4 | Date and time zone offset of the specific request |
| 5 | Method invoked |
| 6 | URL requested |
| 7 | Protocol used |
| 8 | Result code |
| 9 | Number of bytes transferred |
| 10 | Referrer |
| 11 | Browser identification string |

**Table 10-1:** Field values in the `access_log` file

A typical line in an `access_log` looks like the following:

```
63.203.109.38 - - [02/Sep/2003:09:51:09 -0700] "GET /custer HTTP/1.1"
301 248 "http://search.msn.com/results.asp?RS=CHECKED&FORM=MSNH&
v=1&q=%22little+big+Horn%22" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"
```

The result code (field 8) of 301 indicates success. The referrer (field 10) indicates the URL of the page that the surfer was visiting immediately prior to the page request on this site: You can see that the user was at `search.msn.com` (MSN) and searched for "little big Horn." The results of that search included a link to the `/custer` URL on this server.

The number of hits to the site can be quickly ascertained by doing a word count on the log file, and the date range of entries in the file can be ascertained by comparing the first and last lines therein:

```
$ wc -l access_log
   10991 access_log
$ head -1 access_log ; tail -1 access_log
64.12.96.106 - - [13/Sep/2003:18:02:54 -0600] ...
216.93.167.154 - - [15/Sep/2003:16:30:29 -0600] ...
```

With these points in mind, here's a script that produces a number of useful statistics, given an Apache-format `access_log` log file.

## The Script

```
#!/bin/sh

# webaccess - Analyzes an Apache-format access_log file, extracting
#    useful and interesting statistics.

bytes_in_gb=1048576
# You might need to adjust the following two to ensure that they point
# to these scripts on your system (or just ensure they're in your PATH)
scriptbc="$HOME/bin/scriptbc"    # from Script #9
nicenumber="$HOME/bin/nicenumber"    # from Script #4
# You will also want to change the following to match your own host name
# to help weed out internally referred hits in the referrer analysis.
host="intuitive.com"

if [ $# -eq 0 ] ; then
  echo "Usage: $(basename $0) logfile" >&2
  exit 1
fi

if [ ! -r "$1" ] ; then
  echo "Error: log file $1 not found." >&2
  exit 1
fi

firstdate="$(head -1 "$1" | awk '{print $4}' | sed 's/\[//')"
lastdate="$(tail -1 "$1" | awk '{print $4}' | sed 's/\[//')"

echo "Results of analyzing log file $1"
echo ""
echo "  Start date: $(echo $firstdate|sed 's/:/ at /')"
echo "    End date: $(echo $lastdate|sed 's/:/ at /')"

hits="$(wc -l < "$1" | sed 's/[^[:digit:]]//g')"

echo "        Hits: $($nicenumber $hits) (total accesses)"

pages="$(grep -ivE '(.txt|.gif|.jpg|.png)' "$1" | wc -l | sed 's/[^[:digit:]]//g')"

echo "   Pageviews: $($nicenumber $pages) (hits minus graphics)"

totalbytes="$(awk '{sum+=$10} END {print sum}' "$1")"

echo -n " Transferred: $($nicenumber $totalbytes) bytes "

if [ $totalbytes -gt $bytes_in_gb ] ; then
  echo "($($scriptbc $totalbytes / $bytes_in_gb) GB)"
elif [ $totalbytes -gt 1024 ] ; then
```

```
  echo "($($scriptbc $totalbytes / 1024) MB)"
else
  echo ""
fi

# Now let's scrape the log file for some useful data:

echo ""
echo "The ten most popular pages were:"

awk '{print $7}' "$1" | grep -ivE '(.gif|.jpg|.png)' | \
  sed 's/\/$//g' | sort | \
  uniq -c | sort -rn | head -10

echo ""

echo "The ten most common referrer URLs were:"

awk '{print $11}' "$1" | \
  grep -vE "(^\"-\"$|/www.$host|/$host)" | \
  sort | uniq -c | sort -rn | head -10

echo ""
exit 0
```

### *How It Works*

Although this script looks complex, it's not. It's easier to see this if we consider each block as a separate little script. For example, the first few lines extract the `firstdate` and `lastdate` by simply grabbing the fourth field of the first and last lines of the file. The number of hits is calculated by counting lines in the file (using `wc`), and the number of page views is simply hits minus requests for image files or raw text files (that is, files with `.gif`, `.jpg`, `.png`, or `.txt` as their extension). Total bytes transferred is calculated by summing up the value of tenth field in each line and then invoking `nicenumber` to present it attractively.

The most popular pages can be calculated by extracting just the pages requested from the log file; screening out any image files; sorting, using `uniq -c` to calculate the number of occurrences of each unique line; and finally sorting one more time to ensure that the most commonly occurring lines are presented first. In the code, it looks like this:

```
awk '{print $7}' "$1" | grep -ivE '(.gif|.jpg|.png)' | \
  sed 's/\/$//g' | sort | \
  uniq -c | sort -rn | head -10
```

Notice that we do normalize things a little bit: The `sed` invocation strips out any trailing slashes, to ensure that `/subdir/` and `/subdir` are counted as the same request.

Similar to the section that retrieves the ten most requested pages, the following section pulls out the referrer information:

```
awk '{print $11}' "$1" | \
  grep -vE "(^\"-\"$|/www.$host|/$host)" | \
  sort | uniq -c | sort -rn | head -10
```

This extracts field 11 from the log file, screening out both entries that were referred from the current host and entries that are "-" (the value sent when the web browser is blocking referrer data), and then feeds the result to the same sequence of `sort|uniq -c|sort -rn|head -10` to get the ten most common referrers.

### *Running the Script*

To run this script, specify the name of an Apache (or other Common Log Format) log file as its only argument.

### *The Results*

The result of running this script on a typical log file is quite informative:

```
$ webaccess /web/logs/intuitive/access_log
Results of analyzing log file /web/logs/intuitive/access_log

  Start date: 13/Sep/2003 at 18:02:54
    End date: 15/Sep/2003 at 16:39:21
        Hits: 11,015 (total accesses)
   Pageviews: 4,217 (hits minus graphics)
 Transferred: 64,091,780 bytes (61.12 GB)

The ten most popular pages were:
 862 /blog/index.rdf
 327 /robots.txt
 266 /blog/index.xml
 183
 115 /custer
  96 /blog/styles-site.css
  93 /blog
  68 /cgi-local/etymologic.cgi
  66 /origins
  60 /coolweb

The ten most common referrer URLs were:
  96 "http://booktalk.intuitive.com/"
  18 "http://booktalk.intuitive.com/archives/cat_html.shtml"
  13 "http://search.msn.com/results.asp?FORM=MSNH&v=1&q=little+big+horn"
  12 "http://www.geocities.com/capecanaveral/7420/voc1.html"
  10 "http://search.msn.com/spresults.aspx?q=plains&FORM=IE4"
   9 "http://www.etymologic.com/index.cgi"
   8 "http://www.allwords.com/12wlinks.php"
  7 "http://www.sun.com/bigadmin/docs/"
  7 "http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=cool+web+pages"
  6 "http://www.google.com/search?oe=UTF-8&q=html+4+entities"
```

## *Hacking the Script*

One challenge of analyzing Apache log files is that there are situations in which two different URLs actually refer to the same page. For example, `/custer/` and `/custer/index.shtml` are the same page, so the calculation of the ten most popular pages really should take that into account. The conversion performed by the `sed` invocation already ensures that `/custer` and `/custer/` aren't treated separately, but knowing the default filename for a given directory might be a bit trickier.

The usefulness of the analysis of the ten most popular referrers can be enhanced by trimming referrer URLs to just the base domain name (e.g., `slashdot.org`). Script #85, *Understanding Search Engine Traffic,* explores additional information available from the referrer field.