Apache log shell scripts

Return count of timestamps with most error logs.

A couple years ago an app error_log reported about 16000 errors within a few seconds. After fixing it, we wanted to sort by number of errors per second. This does that. The first column is the count.

```
cat /logs/httpd/ssl_error_log* | awk '{ print $1"-"$2" "$3" "$5" "$4 }' | uniq -c | sort -g
```

- uniq -c Precedes each output line with a count of the number of times the line occurred in the input.
- On Solaris, use sort -n to get a numeric sort.

Output

```
      5 [Mon-Jul 02 2012] 07:45:42      5 [Mon-Jul 02 2012] 16:35:18      5 [Tue-Jun 26 2012]
13:40:40      6 [Fri-Jun 29 2012] 13:44:16      6 [Thu-Jul 19 2012] 15:35:59      6 [Thu-Jul
19 2012] 15:36:27      6 [Thu-Jul 19 2012] 15:36:57      6 [Thu-Jul 19 2012] 15:39:43      6
[Thu-Jul 19 2012] 15:40:58      6 [Thu-Jul 19 2012] 15:41:18      6 [Wed-Jul 04 2012]
21:32:38      7 [Thu-Jul 26 2012] 15:06:29
```

Count apache log hits for a given year across a bunch of virtual host logs

```
grep -c "/2015:" /logs/*access_log | sort -t : -k2nr
```

Output

```
/logs/brh.ucla.edu-access_log:28351/logs/harrt.ucla.edu-access_log:27078/logs/jag.ucla.edu-access_log:25690/logs/epr.ucla.edu-access_log:24894
```

Explanation

- This script takes advantage of the default format of the Apache access log files that show date as `[12/May/2015:14:03:48 -0700]`
- grep -c gives a count of the number of matches for the pattern, in this case “/2015”"
- sort -t : takes the grep output in the form of `/logs/nchs.ucla.edu-access_log:40` and splits it into two key fields at the colon.
- Then the -k2nr parameters to sort tell it to sort on the second key, treat it as a number, instead of alpha sort, and then reverse the order, so we’ll get the largest counts at the top.
- This runs on Solaris, but on other *nix systems, you may need to use -g instead of -n to get sort to treat it as a numeric sort.

Look for bytes returned > 1,000,000

We were looking for a bug that was dumping way too much data, and we needed a way to find records that returned more than a million bytes. David Choi figured this out.

```
cat access_log* | grep browseinst | awk -F\" ‘{ print $1" [“$2”] [“$6”] "$3 }’ | grep browseinst | awk ‘{if ($NF > 1000000) print $0}’
```

Be careful when cutting and pasting the above command. The awk statement single-quotes are turning into left and right quotes for some reason, and they won’t work.

Explanation

1. cat access_log* – *feed contents of all access_log files into next part of script*
2. grep browseinst – *only return lines with “browseinst”*
3. awk -F\" ‘{ print $1" [“$2”] [“$6”] "$3 }’ – *split the line into fields delimited by double-quotes, and then only print 1st, 2nd, 6th, and 3rd fields*
4. grep browseinst – _watch for lines with “browseinst” again because it could have shown up in the referrer field, which we don’t want
5. awk ‘{if ($NF > 1000000) print $0}’ – _if last field is greater than 1 million, print out the last set of fields

(Note: try pulling it apart and build it back up, looking at the output at each step. That’s the only way I could make sense of it.)

Output

```
128.97.62.186 - - [14/Jul/2010:10:25:18 -0700]  [GET
/?page=browseinst&term=101&lastalpha=I&instructor=2 HTTP/1.1] [Mozilla/5.0 (Windows; U;
Windows NT 5.1; en-US; rv:1.9.2.6) Gecko/20100625 Firefox/3.6.6 ( .NET CLR 3.5.30729)]  200
1026117128.97.198.33 - - [14/Jul/2010:22:48:16 -0700]  [GET
/?page=browseinst&term=101&lastalpha=T&instructor=1207888 HTTP/1.1] [Mozilla/5.0 (Macintosh;
U; Intel Mac OS X 10.6; en-US; rv:1.9.2.6) Gecko/20100625 Firefox/3.6.6]  200 1059649...
```

Here’s an update, with partially labelled output, get those who returned more than 1,000,000 bytes, but also show the number of seconds the request took.

```
cat /logs/httpd/ssl_access_log | awk -F\" '{ print $1" [URL: "$2"] [BROWSER: "$6"] [REFERER:
"$4"] [SECONDS: "$7"] "$3 }' | awk '{if ($NF > 1000000) print $0}'75.47.248.169 - -
[27/Nov/2011:04:16:50 -0800]  [URL: GET
/file.php/7826/course_materials/Anthro33Lecture13.ppt.pdf?forcedownload=1 HTTP/1.1] [BROWSER:
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121
Safari/535.2] [REFERER: https://.../course/view/11F-ANTHRO33-1?topic=7] [SECONDS:  16]  200
129891475.47.248.169 - - [27/Nov/2011:04:17:08 -0800]  [URL: GET
/file.php/7826/course_materials/Anthro33Lecture15.ppt.pdf?forcedownload=1 HTTP/1.1] [BROWSER:
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121
Safari/535.2] [REFERER: https://.../course/view/11F-ANTHRO33-1?topic=8] [SECONDS:  19]  200
1578667
```

Revision #11
Created 2010-07-15 06:44:20 UTC by Franks, Mike
Updated 2015-03-20 19:26:50 UTC by Franks, Mike