SoundSoftware Code Site - Bug #1045
Major download miscounting (or something else very weird)

2014-09-01 10:07 AM - Chris Cannam

Status:	New	**Start date:**	2014-09-01
Priority:	Normal	**Due date:**	
Assignee:	Chris Cannam	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

In the SV project, see downloads for the 2.4 betas:

http://code.soundsoftware.ac.uk/projects/sonic-visualiser/files

Currently reporting

Source tarball	7
OS/X build	12
Windows build	18963

Definitely something not right there...

History

#1 - 2014-09-01 10:46 AM - Chris Cannam

Looks like a series of 18940 download attempts between 17:31 BST on Wednesday 20th Aug and 22:43 BST on Thursday 21st Aug, neatly spaced at 5-second intervals, from the (Chinese I think) IP address 61.147.94.7.

All of these attempts cut off after a few hundred Kbytes -- as far as I can see the downloader never managed to obtain a full copy. I guess this is a download manager or CLI program re-attempting the download each time it failed. The request is through TLS, contains no referrer, and has the trivial UA string "Mozilla/5.0".

I've reset the download count to exclude these attempts (remaining downloads: 23) but this suggests various things we should look at:

- Do users in network-distant locations such as China habitually find they can't download significant attachments from this site, or was this a one-off with a poor connection at the remote end?
- Should check logs for other popular-looking files, e.g. the download counts for SV releases, easyhg etc look like they may have been inflated in the same way because they are very Windows-heavy and Windows is more widely used in China
- Can we obtain a China-hosted VPS for testing?
- Can we improve the situation, e.g. by mirroring downloads elsewhere, using a CDN (unlikely given finances!) or simply extending timeouts?
- We should not be counting partial downloads in the download count -- investigate how to suppress these

#2 - 2014-09-01 01:17 PM - Chris Cannam

Azure has no China region, only Hong Kong. A quick test from there gives

```
$ wget 'http://code.soundsoftware.ac.uk/attachments/download/1145/sonic-visualiser-2.4beta1.tar.gz'
--2014-09-01 11:45:11--  http://code.soundsoftware.ac.uk/attachments/download/1145/sonic-visualiser-2.4beta1.tar.gz
Resolving code.soundsoftware.ac.uk (code.soundsoftware.ac.uk)... 138.37.95.198
Connecting to code.soundsoftware.ac.uk (code.soundsoftware.ac.uk)|138.37.95.198|:80... connected.
```

HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: ‘sonic-visualiser-2.4beta1.tar.gz’

[<=>] 4,149,815 553KB/s in 8.2s

2014-09-01 11:45:30 (495 KB/s) - ‘sonic-visualiser-2.4beta1.tar.gz’ saved [4149815]

Which looks OK.

#3 - 2015-02-11 05:28 PM - Chris Cannam

Get download lines from all log files in chronological order:

```
$ zcat `echo code-access.log.?.gz code-access.log.??.gz|fmt -1|tac` | cat - code-access.log.1 code-access.log | grep 'download/[0-9]' > ~/downloads
```

Get stats from download lines:

```
$ cat downloads |sed 's/^[^ ]* //'|sed 's/ - .*download\//@/'|sed 's/\/.*//'|awk -F@ '{ print $2, $1 }'|sort|uniq -c|sort -rn
```

#4 - 2015-02-11 05:44 PM - Chris Cannam

And of course we also need to egrep -v '(bot|slurp|crawler|spider)\b' for the same search-bot-removal logic as in attachments_helper.

Drawn from the above, these are the total download counts for all instances where a single IP has download a given attachment more than 100 times (columns are attachment ID and total count) during the last year, excluding bots:

106 23892
107 867
224 3490
370 291
400 103
523 103
534 172
607 24071
618 2601
620 867
625 132
627 249
628 326
638 18517
690 18629
693 14154
705 867
710 2647
711 1008
712 685

760 66885
768 7459
807 188
903 6084
907 1337
908 135
918 2512
1105 106
1123 149
1129 13163
1144 18940
1186 797
1189 220733
1198 37824

And for cases where a single IP has over 1000 downloads of a given attachment:

106 23892
224 1224
607 24071
618 1080
638 18517
690 17095
693 14154
710 2647
760 66047
903 4432
1129 13163
1144 18940
1189 219118
1198 37824

These figures are highly problematic as they exceed the recorded download counts for some (maybe all, I'm not sure). So I obviously need to review again.

#5 - 2015-02-11 05:45 PM - Chris Cannam

I have moved (commit:e2c122809c5c) the download increment to after the send_file call, but I'm not sure whether that suffices -- I don't know whether send_file finishes normally (bad) or throws (good) if the send fails, or indeed whether it's entirely async. To be researched further.

update: no, it doesn't help.

#6 - 2015-02-11 06:09 PM - Chris Cannam

We can always look at the HTTP response size...

```
$ cat downloads |egrep -v '(bot|slurp|crawler|spider)\b' | grep ' 200 '|sed 's/^.*download\///'|sed 's/\/.*200 / /'|sed 's/".*//'| grep -vi '[a-z]'|sort|uniq -c > sizes
```

Then we can find the responses for while the size is at least the intended file size. For 1189 for example this number is 16878.

#7 - 2015-06-30 10:27 AM - Chris Cannam

(Alternatively, we could "fix" the problem from the other angle without resorting to the log files, by only counting e.g. one download per IP address per day.)