Why is ftp directory listing so slow?

Asked 12 years, 5 months ago Modified 11 years, 11 months ago Viewed 6k times

3

The program below gives the following output:

```
Listing directory ftp://ftp.ncbi.nih.gov/pub/geo/DATA/SeriesMatrix
1.1 Mbyte
229.0 s
38 Kbit/s

Downloading file
ftp://ftp.ncbi.nih.gov/pub/geo/DATA/SeriesMatrix/GSE30/GSE30_series_matrix.txt.gz
2.3 Mbyte
3.1 s
6221 Kbit/s
```

Why is the directory listing so slow compared with the file download? Is there any way to speed up the directory listing?

Here is the code

```
from __future__ import division
import ftplib
import time

server = 'ftp.ncbi.nih.gov'
dirPath = '/pub/geo/DATA/SeriesMatrix'
filePath = '/pub/geo/DATA/SeriesMatrix/GSE30/GSE30_series_matrix.txt.gz'

ftp = ftplib.FTP(server)
ftp.login('anonymous', 'john.smith@gmail.com')

print 'Listing directory ftp://' + server + dirPath
fileNames = []
t = time.clock()
ftp.retrlines('NLST ' + dirPath, fileNames.append)
t = time.clock() - t
size = sum([len(fileName) + 1 for fileName in fileNames])
print '{0:.1f} Mbyte\n{1:.1f} s\n{2:.0f} Kbit/s'.format(size / 2**20, t, 8 *
size / 2**10 / t)

print '\nDownloading file ftp://' + server + filePath
blocks = []
t = time.clock()
ftp.retrbinary('RETR ' + filePath, blocks.append)
t = time.clock() - t
size = sum([len(block) for block in blocks])
print '{0:.1f} Mbyte\n{1:.1f} s\n{2:.0f} Kbit/s'.format(size / 2**20, t, 8 *
size / 2**10 / t)

print
raw_input("Press Return to continue")
```

python ftp

Share
Improve this question
Follow

edited Jul 11, 2012 at 18:43

asked Jul 11, 2012 at 12:35

Johan Råde
21.3k ● 22 ● 75 ● 112

3 Answers

Sorted by: Highest score (default)

8

Because the FTP server takes a long time to gather up the contents of the directory into human readable form. You can confirm this by talking to the FTP server directly by hand.

Contrariwise, the transmission of one file requires no server-side preparation, it just sends the file.

Share Improve this answer Follow

answered Jul 11, 2012 at 12:43

msw
43.4k ● 9 ● 90 ● 116

7 Contrariwise? Learned a new word today. Like. – Tim Pietzcker Jul 11, 2012 at 12:46

3 "'Contrariwise," continued Tweedledee, "if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic.' -- from 'Through the Looking-Glass' by Lewis Carroll. – Johan Råde Jul 11, 2012 at 12:57

1

I suspect there is so little data transferred that the speed results are skewed by the connection handshake and python execution times.

You are *not* just measuring throughput but the full round-trip time in your code.

Share Improve this answer Follow

answered Jul 11, 2012 at 12:37

Martijn Pieters
1.1m ● 319 ● 4.2k ● 3.4k

That does not quite explain why it takes almost 4 minutes to download the directory listing. – Johan Råde Jul 11, 2012 at 12:43

Fair enough; didn't pay that much attention to the numbers, just to your measuring method. But @msw is correct; a directory listing of over 1 MB could take some time to build. – Martijn Pieters Jul 11, 2012 at 12:50

I'm not concerned with throughput, I'm concerned with user experience. And waiting 4 minutes for a directory listing is not a good user experience. – Johan Råde Jul 11, 2012 at 12:53

Yup, but in this case, utterly outside of your control. – Martijn Pieters Jul 11, 2012 at 12:55

0

Same problem was here. One of my colleagues works at home and she was unable to get directory listing. She waited 15 minutes and still nothing. I have to wait about 15 secs to get directories, so the connection worked between my home computer and the server. The server is a FileZilla v0.9.41 beta on Windows XP, the client is Total Commander v8.0 on Win7. She has a 120Mbps connection. After reading some articles I changed the server's parameters as follows: - I increased "Internal transfer buffer size" (can be found at Edit - settings - miscellaneous) from 32768 to 262144 and also - "Socket buffer size" from 65536 to 262144. - I enabled "MODE Z" filetransfer compression (Edit - settings - Filetransfer compression). Since then it works like a charm... However I read that you have to test transfer speeds at different buffer size and after that you can choose the best buffer size value (I heard some machines with slow connections doesn't like high buffer size values). Keep tryin'!

Cheers thebaios

Share Improve this answer Follow

answered Jan 9, 2013 at 12:27

thebaios

1