# Homework 2: User-agent Analysis

COSC 6377 Computer Networks

Mai Trinh

April 16, 2021

## 1 Analysis of provided data

Given a file containing headers information that were logged in class, we were to extract and analyze the user-agent strings information based on the paper *Who's left behind? Measuring Adoption of Application Updates at Scale*. The file contains 28 headers. I manually extracted application and operating systems information from HTTP User-Agent strings (results shown in the first 5 columns of Figure 1). To capture the behindness, I calculated an application version's days behind as the number of days which have passed since a newer version of the application has been released. The release data of application versions is publicly available on the Internet. I also calculate the age of OS and application which is the number of days between the current date and the release date of that version. These information are shown in the remaining columns of Figure 1.

| Application | Application Version | Operating System | OS Version | Device | OS Version Release date | OS version age | Application version release date | Application version age | Latest version of application | Latest version application release date | Days behind |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Chrome | 89.0.4389.11 | Mac OS X | 11.2.3 | Macintosh | 3/8/21 | 39 | 3/31/21 | 16 | 90.0.4430.72 | 4/15/21 | 15 |
| Chrome | 89.0.4389.11 | Mac OS X | 11.2.3 | Macintosh | 3/8/21 | 39 | 3/31/21 | 16 | | | 15 |
| Chrome | 89.0.4389.11 | Mac OS X | 11.2.2 | Macintosh | 2/25/21 | 50 | 3/31/21 | 16 | | | 15 |
| Chrome | 89.0.4389.11 | Mac OS X | 11.2.2 | Macintosh | 2/25/21 | 50 | 3/31/21 | 16 | | | 15 |
| Chrome | 89.0.4389.11 | Mac OS X | 11.2.2 | Macintosh | 2/25/21 | 50 | 3/31/21 | 16 | | | 15 |
| Chrome | 89.0.4389.11 | Mac OS X | 11.2.2 | Macintosh | 2/25/21 | 50 | 3/31/21 | 16 | | | 15 |
| Chrome | 89.0.4389.11 | Mac OS X | 11.2.1 | Macintosh | 2/9/21 | 66 | 3/31/21 | 16 | | | 15 |
| Chrome | 89.0.4389.90 | Mac OS X | 10.15.7 | Macintosh | 9/23/20 | 205 | 3/12/21 | 35 | | | 34 |
| Chrome | 89.0.4389.90 | Mac OS X | 10.15.3 | Macintosh | 1/28/20 | 444 | 3/12/21 | 35 | | | 34 |
| Safari | 14.0.3 | Mac OS X | 10.15.6 | Macintosh | 7/15/20 | 275 | 2/1/21 | 74 | 14.0.3 | 2/1/21 | 0 |
| Safari | 14.0.3 | Mac OS X | 10.15.6 | Macintosh | 7/15/20 | 275 | 2/1/21 | 74 | | | 0 |
| Safari | 14.0.3 | Mac OS X | 10.15.6 | Macintosh | 7/15/20 | 275 | 2/1/21 | 74 | | | 0 |
| | | | | | | | | | | | |
| Chrome | 89.0.4389.11 | Windows | Windows 10 | Windows NT | 7/29/15 | 2088 | 3/31/21 | 16 | 90.0.4430.72 | 4/15/21 | 15 |
| Chrome | 89.0.4389.11 | Windows | Windows 10 | Windows NT | 7/29/15 | 2088 | 3/31/21 | 16 | | | 15 |
| Chrome | 89.0.4389.11 | Windows | Windows 10 | Windows NT | 7/29/15 | 2088 | 3/31/21 | 16 | | | 15 |
| Chrome | 89.0.4389.11 | Windows | Windows 10 | Windows NT | 7/29/15 | 2088 | 3/31/21 | 16 | | | 15 |
| | | | | | | | | | | | |
| CriOS | 87.0.4280.77 | iOS | 14.4 | iPhone | 1/26/21 | 80 | 11/23/20 | 144 | 87.0.4280.16 | 4/6/21 | 134 |
| CriOS | 87.0.4280.77 | iOS | 14.4 | iPhone | 1/26/21 | 80 | 11/23/20 | 144 | | | 134 |
| Safari | 14.0.3 | iOS | 14.4.2 | iPhone | 3/26/21 | 21 | 9/16/20 | 212 | 14.0. | 9/16/20 | 0 |
| Safari | 14.0.3 | iOS | 14.4.2 | iPhone | 3/26/21 | 21 | 9/16/20 | 212 | | | 0 |
| Safari | 14.0.3 | iOS | 14.4.2 | iPhone | 3/26/21 | 21 | 9/16/20 | 212 | | | 0 |
| Safari | 14.0.3 | iOS | 14.4.2 | iPhone | 3/26/21 | 21 | 9/16/20 | 212 | | | 0 |
| Safari | 14.0.3 | iOS | 14.4.2 | iPhone | 3/26/21 | 21 | 9/16/20 | 212 | | | 0 |
| | | | | | | | | | | | |
| Chrome | 87.0.4280.88 | Linux | N/A | Linux Deskto | N/A | N/A | 12/4/20 | 133 | 90.0.4430.72 | 4/15/21 | 132 |
| Chrome | 87.0.4280.88 | Linux | N/A | Linux Deskto | N/A | N/A | 12/4/20 | 133 | | | 132 |
| Chrome | 87.0.4280.88 | Linux | N/A | Linux Deskto | N/A | N/A | 12/4/20 | 133 | | | 132 |
| | | | | | | | | | | | |
| Chrome | 89.0.4389.10 | Android | 8.0.0 | Samsung Ga | 8/21/17 | 1334 | 3/22/21 | 25 | 90.0.4430.66 | 4/14/21 | 23 |
| Chrome | 89.0.4389.10 | Android | 8.0.0 | Samsung Ga | 8/21/17 | 1334 | 3/22/21 | 25 | | | 23 |

Figure 1: User-strings extracted information

Figure 2 plots the distribution of days behind for all applications from the viewpoint of April 7, 2021 (the captured date). On this date, we find that only 30% of all applications are running the latest software versions - that is 0 days behind. 80% of the population 34 days or less behind their latest software version. The remaining 20% of population is over 130 days behind the latest application version.

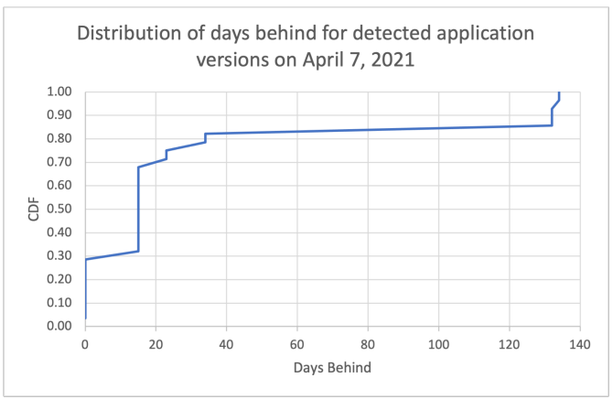


Figure 2: Days behind for all applications

Figure 3 shows, for 5 different operating systems, the distribution of days behind for detected application versions (from all applications running on the given OS). Due to the limited data set, we cannot completely analyze the trend of behindness for each platform. The most popular OS with enough data for analysing is Mac OS X and iOS. Between these platform, iOS have the largest fraction of the population with the application that is up-to-date, with 70% of all applications are 0 day behind, while Mac OS X has 25% of its populations that are 0 day behind. However, overall, all applications running in OSX are less than 35 days behind, while 20% of applications running in iOS are more than 130 days behind. All applications running in Windows and Android are less than 30 days behind, while all applications running in Linux are more than 130 days behind.

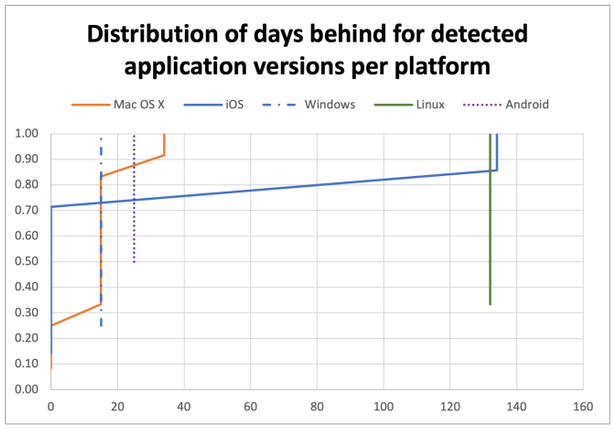


Figure 3: Days behind for all applications per platform

## 2 Hands-on work

First, use one programming language and write a server that accepts an HTTP connection and outputs the user agent string on the console or a log file. To accomplish this, I use the provided *3serv.py* file and modified it to write to a log file. The code snippet for establishing a server is shown below.

```
class GetHandler(SimpleHTTPServer.SimpleHTTPRequestHandler):
    def do_GET(self):
        # Output headers to a log file
        file = open('log.txt', 'a')
        file.write(str(self.headers))
        file.close()

        SimpleHTTPServer.SimpleHTTPRequestHandler.do_GET(self)

PORT = 8001
Handler = GetHandler
httpd = SocketServer.TCPServer(("", PORT), Handler)
httpd.serve_forever()
```

Second, use two different programming languages and write clients that send an HTTP request to an HTTP server with default and customized user agent. I chose Python and Java to complete this tasks. To create clients and send HTTP request in Python, I use *Request* library [2]. Code snippet for Python is shown below. Note that both applications only works in local network on port 8001 (i.e. *http://127.0.0.1:8001*). I combine the code for default and custom user agent in this code snippet because they have similar structure. The only difference is that to use custom user-agent, you specify the headers in *requests.get()* function (check *client.py* and *client-default.py*).

```
import requests
url = "http://127.0.0.1:8001"

#If use default user-agent
response = requests.get(url)

# If use custom user-agent
headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
                         (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36"}
response = requests.get(url, headers=headers)
```

Similarly, code snippet for Java is shown below. I utilize the code snippet from [1]. I used *java.net* library to complete this task. To customize the user-agent string, I specified the property *"User-Agent"* in the *setRequestProperty* function. Again, note that the code snippet below is combining the script for default and customized user-agent since it has a similar structure. In practice, this would be separate into 2 different files (check *client.java* and *clientDefault.java* files).

```
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.io.InputStream;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class client {
    public static void main(String[] args) throws IOException {
        HttpURLConnection connection = null;
        URL url;
        try {
            // Create connection
            url = new URL("http://127.0.0.1:8001");
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");

            // If use customized user-agent.
            connection.setRequestProperty("User-Agent", "Mozilla/5.0
                                                        (iPhone; CPU iPhone OS 14_4 like
                                                        Mac OS X) AppleWebKit/605.1.15
                                                        (KHTML, like Gecko)
                                                        CriOS/87.0.4280.77 Mobile/15E148
                                                        Safari/604.1");

            //Get Response
            InputStream is = connection.getInputStream();
            BufferedReader rd = new BufferedReader(new InputStreamReader(is));
            StringBuilder response = new StringBuilder();
            String line;
            while ((line = rd.readLine()) != null)
                response.append(line);
            rd.close();
        } catch (MalformedURLException e)
            e.printStackTrace();}}
```

I've included below the *log.txt* showing what the server logged.

```
/******** Results from client with default user-agent running Python script ********/
Host: 127.0.0.1:8001
User-Agent: python-requests/2.22.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive

/******** Results from client with customized user-agent running Python script ********/
Host: 127.0.0.1:8001
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) \
            AppleWebKit/537.36 (KHTML, like Gecko) \
            Chrome/70.0.3538.77 Safari/537.36
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
```

```
/******** Results from clients with default user-agent running Java script ********/
User-Agent: Java/1.8.0_281
Host: 127.0.0.1:8001
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive

/******** Results from clients with customized user-agent running Java script ********/
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 14_4 like Mac OS X) \
            AppleWebKit/605.1.15 (KHTML, like Gecko) CriOS/87.0.4280.77 \
            Mobile/15E148 Safari/604.1
Host: 127.0.0.1:8001
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

## References


[1] *Java code snippet.* URL: https://stackoverflow.com/questions/1359689/how-to-send-http-request-in-java.

[2] *Request: Quickstart.* URL: https://docs.python-requests.org/en/master/user/quickstart/#custom-headers.