International Journal of Current Research in Multidisciplinary (IJCRM)
ISSN: 2456-0979
www.ijcrm.com Volume 1 Issue 1 ǁ May. 2016 ǁ PP 27-29

App. Size Reducer: Reduce the Size of Smartphone Application Updates

Mahesh Chincholikar[1], Shaneshwar Bhagat[2], Ajit Laware[3], Gaurav Chinke[4]
[1,2,3,4,]*Department of Information Tech. NMIET, Talegaon, Pune. India*

Abstract:
This technique of creating and deploying update patches improves on Google smart Application Update by initial unpacking the android Application Package so compressing its elements on an individual basis. The smart phone user will then transfer a smaller patch. Experiments show that performance yields 49 % additional reduction relative to Google's resolution, increasing the savings in cellular network bandwidth use and leading to lighter application server loads. This reduction in android application-update traffic may translate to a 1.7 % decrease in annual U.S.A. cellular traffic. Similar methods applied to iPhone application updates may yield even larger savings.

Keywords: *Updates, APK, Server, Network traffic, Compression.*

I. Introduction

Mobile computing is that the discipline for making an data management platform, that is free from abstraction and temporal constraints. the liberty from these constraints permits its users to access and method desired data from anyplace within the space. The state of the user, static or mobile, does not have an effect on the data management capability of the mobile platform. A user will still access and manipulate desired data whereas traveling on plane, in car, on ship, etc. Thus, the discipline creates an illusion that the required data and sufficient process power are obtainable on the spot, wherever as essentially they will be set far-off. Otherwise Mobile computing may be a generic term accustomed refer to a variety of devices that enable people to access data and information from wherever ever they are.

II. Existing System:

Many researchers have conducted experiments to scale back network traffics wherever one in every of them was RT patch. it's a software system that extracted solely the distinction between the recent and new versions that reduced the version size by ninety nine however they were compatible solely with the windows OS [12]. The versatile kind for automaton was changing the applying to automaton Application packages (APK) which could be a nothing archived kind that contains all components of Associate in Nursing automaton application, as well as program computer memory unit code, resources, assets, certificates, and also the manifest file. They contain six main components META-INF directory, Classes.dex, lib directory, Resources.arsc, res directory, AndroidManifest.xml [4]. These files kind one APK. more Google has used "Delta Encoding" as ,"Google Smartapp Update" that finds the distinction between the recent APK and also the new APK victimization the bsdiff rule and also the distinction is taken as a "patch" that is update to the end-user. This methodology of delta coding has reduced the traffic up to forty nine p.c [4].

III. Proposed System:

In our paper, we offer another flip of plan to scale back the network traffic because of updates alittle any. Studies show that image files consume extra space than text files. so compression over text files that contains the java code, .xml files and different such files area unit compressed victimization ‚proguard‟ that depends within the ADT (Android Developer Tool). The proguard.cfg file is directly created in the root directory on making an android app using ADT and it removes all unused and debug classes from the code and shrinks it. We invoke this within the APK and shrink the java file. The second compression we tend to propose is that the compression of pictures [13]. It is experimentally proven that pictures occupy additional memory than text [11] and so compress image files using compression tools like Jstrip, PUNYpng, etc., Thisprovides losslesscompression of the image up to sixteen.1 percent less than the initial size [11].

This compressed file is integrated into the APK(if image files area unit necessary for the app) then uploaded to the server for updating. By these strategies we tend to scale back the network traffic to regarding twenty p.c together with Delta encryption, so creating lighter app hundreds.

IV. Methodology Description:

Delta++ could be a new technique for reducing the scale of android application updates [9]. It provides substantial reduction in network traffic created by application updates. It will therefore by computing the distinction between the older and newer versions of the application files. This distinction is employed to create the patch that contains the relevant files for change. a smart phone application is updated by downloading solely this distinction of previous version and also the new one and applying the delta patch in smart phones. This method includes decompression of android APK package and compression is performed on every modules of the APK [9].

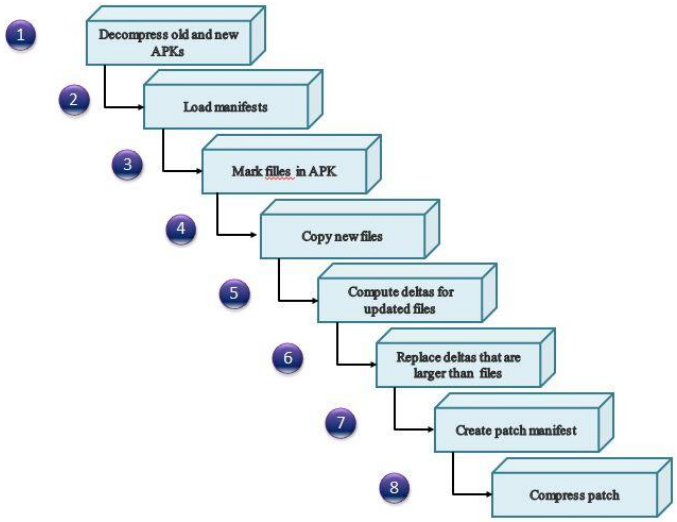

Fig 1. Steps in App Size Reducer patch construction.

V. Arithmetic Diagram:

APK Creation: Here we have a tendency to act because the administrator and make our own application beside its updated feature. A cloud acts as a server wherever we have a tendency to upload the newer versions for updating the application. Here we have a tendency to create use of associate android info named SQLite. The standard File/Open operation calls sqlite3_open () to connect to the database file [9]. Updates happen mechanically as application content is revised that the File/Save menu choice becomes superfluous. The File/Save As menu choice are often enforced victimization the backup API. Before the files are bundled in to application package, they are on individual basis compressed victimization numerous tools which is able to be mentioned below.

VI. Compression of an apk:

In order to reduce the dimensions of AN apk we tend to commit to compress the files present within the application package[14]. This compression is finished at the server by the content supplier. This method uses the advanced delta encryption technique referred to as delta++.This uses the bsdiff algorithm[3] that calculates the distinction between 2 files by unpacking the apk's.

VII. Proguard:

The ProGuard tool shrinks, optimizes, and obfuscates your code by removing unused code and renaming classes, fields, and strategies with semantically obscure names. The result's a smaller sized .apk file that's a lot of difficult to reverse engineer. ProGuard is integrated into the android build system, thus you are doing not need to invoke it manually [13]. As presently as an apk file is created, proguard.cfg file is mechanically generated within the root directory of the project. This file defines however ProGuard optimizes and obfuscates your code, thus it's important to know a way to customise it for your needs[15]. To modify ProGuard so it runs as a part of an Eclipse build, set the proguard.config property within the <project_root>/project. Properties file. the path are often an absolute path or a path relative to the project's root [13]. android compiles Java source files to Java byte code. The Java byte code is next regenerate to Dalvik byte codes by the "dx" tool. In between these two steps we are able to apply Proguard to shrink the Java byte codes. The result's impressive: a reduction of regarding half-hour or a lot of in .apk size.

This leads to the application downloads quicker, taking on less area on the restricted flash storage on the phone and initiating quicker. The only disadvantage, it makes analyzing an application crash harder, because the stack trace of an exception has only the new short names and therefore is pretty obscure[16].

VIII. Image compression:

The images area unit compressed before constructing an apk file. Here we tend to use lossless compression algorithms. The image remains the same in its quality before and once compression. The Lempel-Ziv-Welch algorithm[14] provides an encoding and decoding of pictures without any loss.

A. Encoding:

A high level read of the encoding algorithmic rule is shown here:

1. Initialize the dictionary to contain all strings of length one.
2. Realize the longest string W within the dictionary that matches the present input.
3. Emit the dictionary index for W to output and take away W from the input

B. Decoding:

1. Scan the value from the encoded input encoded input and output the corresponding String from the initialized dictionary.
2. The decoder yield to following input value.
3. Concatenates this string with the primary character of following input when coding it.
4. The method is recurrent till there's no a lot of input.

C. Deployment:

After the user had with success registered to enter in to the application an update founder is provided. User will check all the applications put in into their Smartphone. they will additionally check whether or not an update for the several application is offered within the server. If the updates are found then the size of the application is checked. If the user desires to update then he will download the newer version and install it in to their Smartphone.

IX. Future enhancement:

The compression will be still improved by unpacking the apk's and comparison the size of recent and new versions of an application. As the behavior of AN apk is clearly understood we can additionally involve horizontal matching .An update which may be compatible to two different versions of AN app might even be created.

X. Conclusion:

Due to compression applied to the apk's the complete size of an application is small even with an updated feature. therefore it eliminates the delay in downloading during a serious traffic network. It conjointly saves the ability as installation is quick and supported the files Compressed, memory is additionally with efficiency used. Downloading the app a smaller application package travels through the network that reduces the amount of bandwidth needed per MB. This reduces the traffic within the network. Thanks to this saving in memory the speed and potency of the mobile remains an equivalent even if it is updated with such a big amount of applications. this could contribute a lot of benefits to iphone.

References

[1] N. Samteladze and K. Christensen, "DELTA: Delta Encoding for Less Traf_c for Apps," Proc. IEEE Conf. Local
[2] "K. De Vere, "Android Reaches 25 Billion App Downloads, 675,000 Total AppsAvailable,
[3] "comScore Reports July 2012 U.S. Mobile Subscriber Market Share," comScore, September
[4] T. Simonite, The Great Bandwidth Brawl, MIT Technology Review, May 30,2012. URL:
[5] "Cisco Visual Networking Index: Global Mobile Data Traf_c Forecast Update, 2012–2017," Cisco, 6 Feb. 2013;
[6] "State of the Media: Mobile Media Report Q3 2011," Nielsen, 15 Dec. 2011; www.nielsen.com/us/en/reports/2011/stateof- the-media--mobile-media-report-q3-2011.html.
[7] S. Musil, ―Google Play Enables Smart App Updates, Conserving Batteries,‖ CNET News, 16 Aug. 2012; http://news.cnet.com/8301-1023_3-57495096-93/google playenables- smart-app-updates-conserving- batteries/.
[8] C. Percival, ―Naive Differences of Executable Code,‖ draft, 2003; www.daemonology.net/bsdiff.
[9] Nikolai Samteladze, ―Delta Encoding Based Methods to Reduce the Sizeof Smartphone Application Updates―,January 2013. URL: University of South Florida, nikolay.samteladze@gmail.comFollow this and additional works at: http://scholarcommons.usf.edu/etd.