Extending Internal Memory in Android Phones

Akshay A. Nawalkar[1] Akshata R. Pardeshi[2] Khushboo K. Nalbhimwar[3] Apurva V. Mulay[4]
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]Modern Education Society's College of Engineering, Pune, India

Abstract— One of the major issues of android phones is storage space running out. Storage space running out causes the phone to slow down and the user has to uninstall apps so as to download the new ones, as the internal memory becomes full. The limited size of the internal memory puts restrictions on the number of applications that one can install on a mobile device at any given instance of a time. As a solution, this paper has come up with "Expanding internal memory in android phones". To expand the internal memory, it makes use of cloud computing technology. The proposed system uses an android app to store the less frequently used or unused apps on the cloud. The idea behind this proposed system is that the whenever the user feels that this app is not required that frequently or he might need it later on, but he doesn't want to uninstall it he can store that app on the cloud, he can also retrieve that app whenever it is required. The shifting of the less used or the unused apps makes space for the new apps making internal memory available.

Key words: Mobile App, Cloud, Extending Internal Memory, Mobile Internal Memory, Security, LRU

I. INTRODUCTION

Nowadays, mobile phones provide all functions similar to computer. Hence, the mobile devices have become more popular. As the mobile applications are gaining more popularity than the web applications there is a requirement to install more and more apps on the smart phone. There are various operating systems for smart phones, Android is the most widely used. This is the reason we are working with android operating system. Mobile devices, have become the most necessary and useful devices which provide many functions. Every day, humans face the problem of less storage in their smart phones. In order to access all these functions, and perform tasks using these devices, we need to install apps in the smart phones. So the limited internal memory does not allow the users to install many apps. In this, we also try to overcome the limitation of the external memory, i.e. SD card. Focus is on providing a cloud based solution so that user can have more apps in their phones. The user has to select which app he wants to shift on the cloud, then the app will be temporarily shifted on the cloud. This will allow the users to install more apps, extending the internal memory. Whenever the user wants to run the app which he moved to the cloud, it is brought back to the internal memory [1]. The application is completely moved to the cloud. This avoids permanent deletion of the app.

II. LITERATURE REVIEW

Now a days a smart phone is an essential part of people life. They provide different apps for the user. Users need to install more and more apps to help then with their work. These app occupy most of the internal storage spaces. Due to which most of the users are facing the problem of limited internal storage space.

To solve this problem Google created a feature name android:installLocation for the users. In this feature the user can specify the location where the app should be installed. The user can provide the location of app in a file name AndroidMainfest.xml. This feature contains two modes [5].

The first mode is the preferExternal mode. In this mode the app will directly get installed in the external storage [5]. But this mode has some drawbacks. It will not necessarily install the app in the external storage. Also it only installs apk files on the SD card the remaining files like users private data, database, dex files are stored in the internal memory only. The second mode is the auto mode. In this mode his system decides which apps should be located where. If the user needs to move any app from internal memory to the external memory, then he can do so only in the auto mode [5].

Another app was produced solve the problem of limited storage space. It is Link2SD. For this app there must be two partitions on the SD card. The first part acts as regular external storage. While the second part works as an extended app installable memory partition for apps [6]. The Link2SD app moves the user apps to extended app installable memory partition and keep its link in the internal storage. But this app too has a drawback. If the extended memory on the SD card gets full then the app starts giving errors. Then the user needs to uninstall apps to solve this problem.

To solve the limited internal storage space problem Nextbit a US based phone company launched a smart phone called Robin. The problem of limited internal storage was solved in this smart phone. Unfortunately this solution was not generic i.e. it was only available for Robin users only [7]. In the proposed model the limitation of limited internal memory is solved. This model uses a cloud storage space to solve this problem. It uses the LRU strategy to select the apps to move from the internal storage space to the cloud. It also leaves a link or dummy icon behind which can be used by the user to again restore the app back from the cloud. The apps moved to cloud may contain private information of user which can be misused on the cloud. Hence this model also provides security which will encrypt the apps while moving and again decrypt it while restoring process.

III. TECHNOLOGIES

The expanding internal memory on android phones comprises of:

A. Android App: It Has Three Options

1) Register
This will link the phone to the cloud. It is mandatory to register so that the user can get the reserved cloud space.

2) Migrate
App chosen by the user, is moved to the cloud, a link is present in the device to move the app.

3) Restore
User will tap on the link in the device to restore the app which was earlier moved to the cloud.

B. Cloud

It will shift the app and the app data, which the user chooses to move. It also provides access to the cloud whenever needed.

C. LRU

In this strategy, user apps are listed in order of their last used time stamp.

IV. IMPLEMENTATION

Whenever less storage situation arises user installed apps are identified. With the help of Least Recently Used Strategy (LRU) apps which are to be migrated to cloud are selected. Least Recently Used is a strategy in which the app which is not used for longer time is listed in the list and it is moved to cloud. If it does not satisfy the memory requirement of the new app then next app from the list is also moved to the cloud using same strategy. While moving the apps they are forcefully stopped to collect all types of files like dex, lib, apk in a compressed package. A link is used to move the apps to cloud. Before migrating the apps they are encrypted so that the apps and data regarding them should not be accessed by any other unauthorized user and the data will be secured. After encryption process is over the apps are moved and they are stored at user's reserved space.

Whenever user wants to retrieve any app first he need to decrypt the data and app. The user will click the icon to retrieve the app which is moved to cloud. While retrieving app's compressed package will be checked. If it is creating low internal memory situation then above process will repeat but if not then download the icon selected app package from cloud to mobile. Extract all the files from downloaded package and restore the files in internal memory so that the directory hierarchy will remain same as it was before moving the app to cloud.

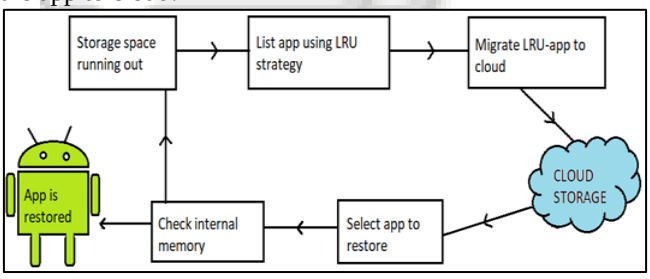

Fig. 1: Flow of System

V. CONCLUSION

Increasing Smart-phones internal memory after a certain limit is not possible. The requirement of new apps in mobile phones is increasing every-day. This paper proposes to overcome internal storage problem in android smart phones by moving the apps to cloud whenever they are not required and retrieving them back from cloud to mobile phones whenever user needs them. This approach only focus on android smart phones and not on any other smart phones. It has a big advantage that the solution is provide at Application Layer and there is no need to make any changes in the kernel. This approach is highly flexible and it is cost saving as it does not require much hardware. As it is not using hardware no physical problems and backup issues are generated. The speed of performance is very quick and fast and so it a very convenient approach.

VI. FUTURE SCOPE

As propose paper is for rooted Smartphones. We would like to extend this work for other mobile OS like Apple iOS, Windows Phone, BlackBerry OS, etc. so that every Smartphone user get benefited. Instead of permanently removing app user can utilize cloud storage to store App. For security purpose we would add Encryption-Decryption so that only authorize person can access it.

REFERENCES

[1] Sanjay Singh, Ashwin Nivangune, Sathish Kumar, Ranjan Kumar, Padmaja Joshi, Dhiren Patel. Extending App Installable memory in Android Smartphones. In IEEE International Conference on Mobile Software Engineering and Systems, 2016.
[2] Ranjan Kumar, Ashwin Nivangune, and Padmaja Joshi. Challenges in transition from web to app. In MobileDeli 2015, Proceedings of 3rd International Workshop on Mobile Development Lifecycle, pages 9-10. ACM, 2015.
[3] Smartphone os market share, 2015 q2 [online]. http://www.idc.com/prodserv/ smartphone-os-market-share.jsp
[4] Android (operating system) [online]. Https: //en.wikipedia.org/wiki/Android (operating system).
[5] App install location [online]. http://developer.android. com/guide/topics/data/install-location.html
[6] Description [online]. http://www.link2sd.info/description
[7] Nextbit phones [online]. http://gadgets.ndtv.com/mobiles/nextbit-phones

