

# Extending App Installable Memory in Android Smartphones

Sanjay Singh  
SVNIT, Surat  
Dept. of Computer Engg.  
Ichchhanath, Surat, India  
san.singhsanjay@gmail.com

Ranjan Kumar  
C-DAC  
Gulmohar Cross Road No. 9,  
Juhu, Mumbai India  
ranjank@cdac.in

Ashwin Nivangune  
C-DAC  
Gulmohar Cross Road No. 9,  
Juhu, Mumbai India  
ashwin@cdac.in

Padmaja Joshi  
C-DAC  
Gulmohar Cross Road No. 9,  
Juhu, Mumbai India  
padmaja@cdac.in

Sathish Kumar  
C-DAC  
Gulmohar Cross Road No. 9,  
Juhu, Mumbai India  
asathish@cdac.in

Dhiren Patel  
SVNIT, Surat  
Dept. of Computer Engg.  
Ichchhanath, Surat, India  
dhiren29p@gmail.com

## ABSTRACT

The increasing need of app installations on mobile devices demands a lot of internal memory i.e., *app installable memory*. 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. The research work in this paper focuses on providing a cloud based solution to the *limited app installable memory* to allow the users to have more number of applications on their smart phone. The proposed solution uses a cloud to extend the app installable memory of user's mobile. The unused or less used apps are moved over to cloud storage until they are required by the user thereby making internal memory available for new installations. The moved apps maintains user data and avoid permanent deletion of apps.

## Keywords

Mobile App, Extending Internal Memory, MCC, Mobile Internal Memory

## 1. INTRODUCTION

Nowadays, mobile devices are providing functionality similar to that of computer systems with added feature such as GPS navigation system, motion sensors, cellular communication, enabling cyber physical integration. These make the mobile devices much more popular. As per the IDC report, market of smartphones grew worldwide by 13% in 2015 Q2 [1]. Note that the words *mobile devices* and *smart phones* are used interchangeably in this work and do mean the same w.r.t. the work in this paper.

There are various mobile operating systems for smart phones like Apple iOS, BlackBerry, Windows Phone OS, etc. Among

all these mobile operating systems, Android is the most widely used with a market share of 82.8% [1]. Android is an open-source mobile operating system currently developed by Google and using Linux kernel at its core [2]. This huge adoption of Android OS makes us focus it for our research work in this domain.

Today, mobile devices have become the most necessary and useful devices which can provide multiple functions. In order to access online services as well as to perform offline tasks using these devices, one needs to install apps on these smart phones. As these devices are handheld and mobile, they are resource critical with respect to their size, memory and power. One such crucial resource is app installable memory, known as internal memory. This limited app installable memory prevents the users from installing too many apps.

Smart phone memory is divided into three types: RAM, internal storage and external storage. (1) RAM provides fast access for files to the processor. It's a unit which temporarily stores files only for processing purpose. This is a volatile memory same as present in desktops and laptops. (2) The internal storage is available for installing applications. It contains sensitive information such as SMS, contact list, email settings, etc, as it is a non-volatile storage. Thus, files saved by an app to this storage are private to itself and hence, not accessible by other apps and/or users [3]. In this paper, this storage class of smart phone memory organization is referred as *app installable memory*. (3) An external storage unit can be a removable storage unit like an SD card or it can also be a non-removable storage unit. It is a shared storage which a user can use to save his/her data files. Files stored over this storage unit are readable to all [3].

The paper is organized as follows. The next section discusses the background information which is necessary to understand the proposed solution. Section 3 talks about the current research efforts for extending app installation memory. Section 4 describes our approach to help resolve the point in question. Section 5 elaborates the challenges faced while implementing the proposed approach along with the results of the migration. Finally the paper is concluded with Section 6 covering further intended work on the said topic.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'15 April 13-17, 2015, Salamanca, Spain.

Copyright 2015 ACM 978-1-4503-3196-8/15/04 ...\$15.00.

## 2. BACKGROUND

Majority of smart phone users are suffering from limited internal memory space. Because of which they are unable to install many Android apps or need to remove the apps before installing the new required ones.

The need of having more and more apps on our smart phone is increasing daily. A few e-commerce companies like Flipkart, Myntra, OLA cabs etc, have shut down their website and are operating using mobile apps as the only way of their online business. It may become a path followed by many. Today, even most of the banks are providing their apps. Every online service provider has an app and for using their services requires the app to be installed in one's smart phone. But limited internal memory of the smart phone prevents users from installing many apps in their smart phones.

An Android app has different parts, its executable, other required files, and the data required and/or data generated by the app. These files of an installed Android app are scattered at three different locations:

1. Data directory - All installed apps have one directory under `/data/data` for storing data [4]. This data can be the data which comes with the app such as database or it can be a data generated while using the app. It resides in the internal memory of smart phone.
2. Installation location of apps - A software vendor can set a copy protection for his/her app to On or Off before publishing. If the protection is Off, the default installation location for the app will be `/data/app` but if the protection is On, then the installation location for the app will be `/data/app-private` [4]. Both the locations, `/data/app` and `/data/app-private`, are in the internal memory.
3. Location of Dex file - Dex or Dalvik executable file is created from Java compiled classes to ensure minimal memory footprint [5]. For every installed app, at least one dex file is present in `/data/dalvik-cache` directory present inside the internal memory[4].

## 3. RELATED WORK

To solve the issue of limited app installable memory, Google started providing an optional feature, `android:installLocation` to the app developers. With this feature, developers can specify the installation location for their app. App developers can specify the value of this attribute in the file `AndroidManifest.xml`. This attribute can have values `preferExternal` or `auto` [6]. The value `preferExternal` is considered as a request to the system to install the app on external storage. But it is not guaranteed that system will install app on external storage. Only apk file (a package format to distribute Android app) will be moved to an SD card and rest of all private user data, database, dex file will remain in the internal storage [6]. When the attribute value is `auto`, the system will decide about the app's installation location [6].

A user can move an app between internal storage and external storage only when the value is `auto`. A few apps available on Google play store (like AppMgr III, Move App to SD card, AppToSD) does the job of moving user installed apps to the external storage by referring the attribute value `android:installLocation` specified above.

`Link2SD` is an app available on Google play store. This app requires two partitions on the SD card. One partition

works as a regular external storage while the second partition works as an *extended app installable memory partition* for apps. It moves the user installed apps to the *extended app installable memory partition* and keeps a link for them in the internal memory [7]. But if this extended app installable memory partition of SD card gets completely loaded, this app will start giving errors. In such a situation, the only way is to uninstall the apps. Thus, `Link2SD` is a good product to solve the limited internal memory problem of smart phones but still does not resolve the problem completely. Another limitation of `Link2SD` is those smart phones which don't support SD card and hence, `Link2SD` cannot help them.

`Nextbit` is a US based smart phone company launched in August-2015. `Robin`, a smart phone launched by Nextbit provides a solution to this problem as per the reports [8]. However, this solution is available only for `Robin` users as the solution is a part of the OS of Robin. The solution is not generic.

## 4. OUR APPROACH

To make the solution generic, which can be used by all the android users, we focus on the application layer as oppose to the kernel layer as done in `Robin`.

In our approach we also try to overcome the limitation posed by external memory and propose to use cloud instead of an external SD card. This way the proposed approach will be useful for all the android users irrespective of the smart phone they are using.

We propose to move the selected apps to a cloud setup temporarily which will allow users to download and install more number of apps extending the internal memory. Whenever, the user wants to run the migrated app, it is brought back to the internal memory. The application is completely (including data, app files and dex) moved to the cloud. This movement has posed multiple challenges which are discussed in the next section.

Selection of applications for migrating to cloud is important aspect. For example the system applications cannot be migrated to cloud. Figure 1 shows the process followed to migrate an app to the cloud setup whereas Figure 2 provides the process followed for restoring the migrated app back to the device. The approach is applicable only for the user installed apps and not for the system apps. The movement of the system apps from internal memory may hamper the functioning of the phone and/or functioning of those apps. The applications that are least recently used are used for migration to reduce the upload/download frequency from the cloud. Another aspect that can be taken into consideration is the time duration for which the app was used in foreground.

## 5. CHALLENGES AND RESULTS

This research work is focused on extending an app installable memory in Android smart phones by moving less used Android apps to the cloud and bring them back when user wants to use them again. Though it sounds very trivial, many challenges were faced in its implementation. A few of these are listed here.

- *Identifying user apps* - An Android smart phone consists of many apps. A few of them are system apps which comes with system and are essential for the

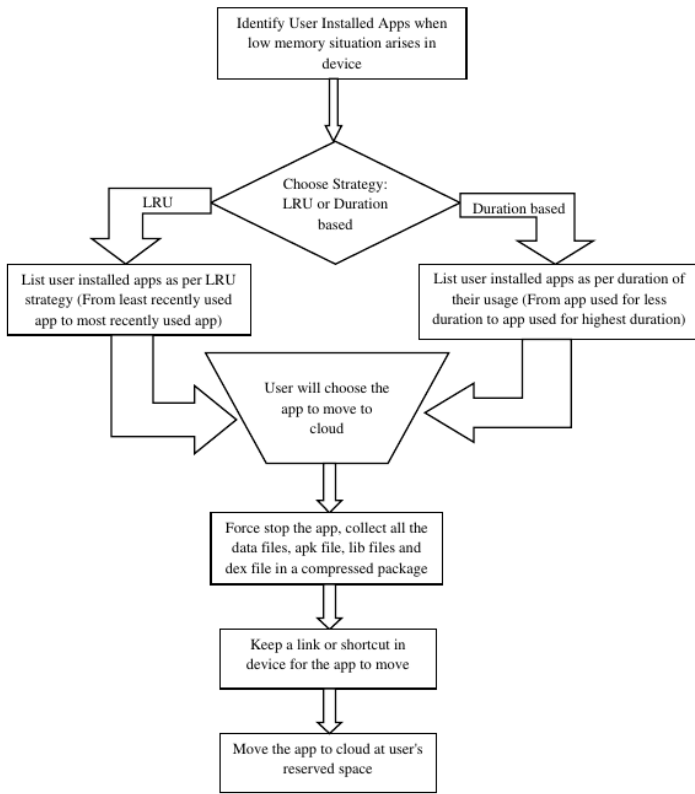


Figure 1: Process to Migrate an App to Cloud

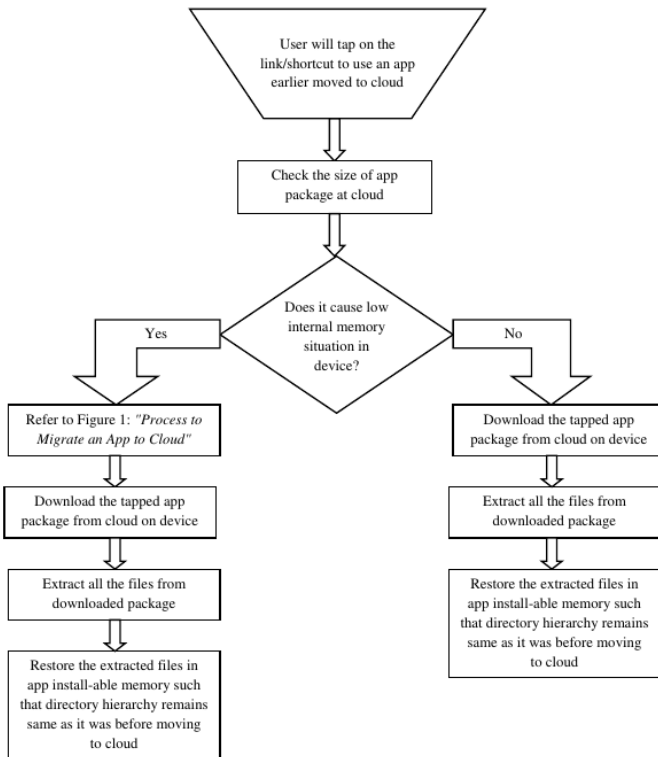


Figure 2: Process to Restore the App

proper functioning of the system. For example, Bluetooth, Auto Dialer, System UI, Android Keyboard, etc. Apps which are installed by the user are considered as user apps. Moving such apps doesn't harm the proper functioning of system. Hence, the work focuses on identifying the user apps that can be relocated to the cloud when not in use.

- *Locating apps for relocation* - After identifying user apps, it is required to find less used user apps. There can be two strategies to identify a less used user apps.

- *Least Recently Used (LRU) based strategy* - In this strategy, user apps are listed in order of their last used time stamp. An app which is not used for the longest time tops the list. Such an app may be relocated to the cloud. If relocation of one app does not fulfill the memory requirement of the new app, the next app in the list is moved to the cloud with the same strategy.

- *Usage Duration based strategy* - As per this strategy, an app which is used for the smallest period of time in the foreground is listed first and an app which is used for the longest duration of time in foreground is listed last. Thus, an app which is used for the smallest duration of time and satisfying free internal memory space demand is chosen for migration.

- *Getting location of installed files of user apps* - In Android, an app doesn't get installed at one location as mentioned in Section 2. Its files are stored at different locations in the internal memory. Thus, finding and collecting them from different locations in the internal memory becomes a challenging task.

- *Access to the internal memory* - As the internal memory of Android smart phones is used to keep installed files and the private data of apps, so it is not accessible to the users. Even an app cannot access the files of other apps. So, without having an access to the internal memory, moving an app with all its files becomes the biggest challenge.

- *Preserving the launcher icon of the app after moving it to cloud* - After moving an app to the cloud, it is required to keep its launcher icon on the phone for later use so that when the user wants to use that app again, he/she can tap on that preserved launcher icon which starts the process to bring the app back to the device from cloud and finally app can get started.

- *Bringing app back to device* - Bringing the migrated app back to the device requires to preserve its entire directory structure and placing them at the same location in internal memory, only then, the app can run again.

## 5.1 Implementation

An Android app is developed to identify user apps, listing them on the basis of LRU (Least Recently Used) and usage duration based strategy, migrating it to cloud and restoring it back to its location in device. Note that the application is developed on the application layer. For accessing internal

**Table 1: Application Migration Results**

App Name	App Size (in MB)	Available App Installable Memory		Time in Milliseconds	
		After Moving App	After Restoring App	Time to Move	Time to restore
NDTV News App	35.85	320	284	12121	8898
Amazone Shopping App	35.15	310	275	19310	18208
Google Drive App	45.16	294	249	14017	8201
Whatsapp App	49.12	241	192	21062	10773
Twitter App	54.91	220	165	15527	9981
Paytm App	30.24	211	181	8417	5214
IRCTC App	13.59	203	190	2309	2719
Youtube App	45.48	189	143	8783	6523
Dropbox App	79.50	147	67	16844	14601
M-Indicator App	21.66	141	120	8464	8552

memory of the other apps however, currently we need to root the phone. This allows to migrate the application without accessing the kernel or recompilation of an OS. With the proposed technique, we are able to move all the identified user apps to cloud.

Restoration of apps is still challenging. After restoring the app back to its actual installed location, majority of the apps are working fine. However, a few Android apps like Facebook Messenger, Temple Run, Flipkart, Terminal Emulator are partially working. The issue we are still analyzing.

## 5.2 Testing

The developed app is tested on a Micromax A106 Android smart phone running Android 5.0. Since, Android OS doesn't provide access to app installable memory, the device was rooted. Table 1 provides the details of the testing results on ten different applications.

It can be observed from Table 1 that nearly the complete app gets migrated including the data, dex and other files, as the available memory increases nearly by the amount equal to the size of the app. This is different from many of the current available apps which do not move the complete app to the SD card.

Table 1 also provides the details about the time required for each app to migrate to cloud and restore it back. The timings given in the table are in milliseconds.

## 6. CONCLUSION AND FUTURE WORK

The handheld devices are resource limited because of their restricted size, weight and power consumption. So, increasing internal memory beyond a limit is not possible for smart phones. However, the need of installing more and more apps on mobile devices is increasing day by day. The work in this paper proposes to overcome the limitation of app installable memory by moving less used user installed apps to a cloud. The proposed approach focuses only on Android apps and also has the limitation of rooting the device. The advantage of the approach is the solution is provided on the application layer and does not require any changes in the kernel. The testing also shows that the proposed mechanism works well with majority of the user installed apps.

Currently, the developed app is lacking in two features: (1) A background service which pops-up for starting the procedure of moving app to cloud whenever low internal memory situation arises in the device. (2) An interface to launcher such that whenever the user wants to use moved

app, user can simply tap on a dummy icon which starts the procedure of restoring app.

We would also like to extend this work for Android non-rooted devices and also for those handheld devices which are running on other mobile operating systems such as Apple iOS, BlackBerry, Windows Phone OS, etc. So that every user of mobile and handheld device get benefited by using many different apps as well as he/she can utilize the cloud storage by storing apps without permanently removing them.

## 7. REFERENCES

- [1] Smartphone os market share, 2015 q2 [online]. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>, [ACCESSED: Jan-06-2016].
- [2] Android (operating system) [online]. [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)), [ACCESSED: Jan-06-2016].
- [3] Using the internal storage [online]. <http://developer.android.com/guide/topics/data/data-storage.html>, [ACCESSED: Jan-07-2016].
- [4] Karim Yaghmour. *Embedded Android Porting, Extending, and Customizing*. O'Reilly Media, 2013.
- [5] Reto Meier. *Professional Android Application Development*. Wrox Press, 2008.
- [6] App install location [online]. <http://developer.android.com/guide/topics/data/install-location.html>, [ACCESSED: Jan-07-2016].
- [7] Description [online]. <http://www.link2sd.info/description>, [ACCESSED: Jan-07-2016].
- [8] Nextbit phones [online]. <http://gadgets.ndtv.com/mobiles/nextbit-phones>, [ACCESSED: Jan-09-2016].