ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor & UGC Approved Journal)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

Performance and Battery Optimization in AOSP

Alok Sharma[1], Sunil Nimawat[2]

Student, Department of Computer Science and Engineering, IES IPS Academy Indore, Madhya Pradesh, India [1]

Asst. Professor, Department of Computer Science and Engineering, IES IPS Academy Indore, Madhya Pradesh, India [2]

ABSTRACT: Android is becoming successful in mobile market presenting various opportunities in this field. Since the evolution of Android, various features like better User Interface (UI), better efficiency, and better device performance it has been ported to various embedded device. The android framework can be used beyond mobile devices weather they are powered by android or any of other developer. This research present the opinion of the Custom ROM with the possibility of increasing the performance of old devices by using some modification and removing various unnecessary objects which may not be useful for the user also removing the unwanted application to increase battery life and performance The main orientation of this research is to optimize and increase overall efficiency and performance for a outdated devices.

KEYWORDS: Root Android, Custom ROM, Stock ROM, Custom recovery, TWRP, Bloatware, Root access, Android open source project (AOSP).

I. INTRODUCTION

Since the starting era of using the smartphones the users using Symbian, iOS and Windows a new operating system had been introduces into the market known as Android on 23rd September 2008. Android is a operating system developed by Google for various devices now a days like smartphones, setup boxes, locomotives, various enterprise use, various machines used in industry, etc. This operating system is based on Linux-based software system. Due to the similarity between Linux and android is that they are free and open source it is becoming the most widely used mobile OS these days.

On various collected data that Android has covered more than 50% of total shares Smartphone market. Various applications (e.g. Google Play Services, gaming, entertainment and various communications) can be installed by the user so as to simplify the daily ongoing life purposes and basic life necessity. Since the implementation of an application constant use can arise some vulnerability like shortage of memory caused by running applications. It is happened when allocated objects live longer than the expected lifetime inside an Activity in an application (an instance for the running application) which creates reference of unused memory objects for a long period of time which in results deny request for new memory allocation. Android has built-in memory manager regularly checking for garbage collection regardless of this, the adequacy of managing the resource allocation table is not fully optimized. According to "The Verge" Android and iOS accounted for 99.6% of current Smartphone's market in the 4th quarter of 2016. Statistically out of 432 million smartphones user, 352 million were Android (81.7%) and 77 million were iOS (17.9%), Windows Phone rounded up the 0.3% of the market, while BlackBerry cannot be reduced giving a rounding error. Smartphones can be described as general-purpose computers with an attached phone but many people use smartphones for their daily consumption, storage and communications tasks and hence, altering functionality is a way of thwarting an analysis. Android and iOS smartphones are designed to allow the installation of third-party application. Since, these applications work under the restrictions imposed by the operating system, such as application isolation and responsiveness demands.

Android was developed by Google which is generally based on Linux kernel and GNU software. Android, binded with open Handset Alliance in July 2005, it was initial development of Android Inc. (Firm taken under by Google).

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor & UGC Approved Journal)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

1. Android Releases

Android has a compiler which helps the system Apps open faster; java script and android browser is also faster with the added support of adobe flash. Speed as well as Performance boost is really a great advancement in this version

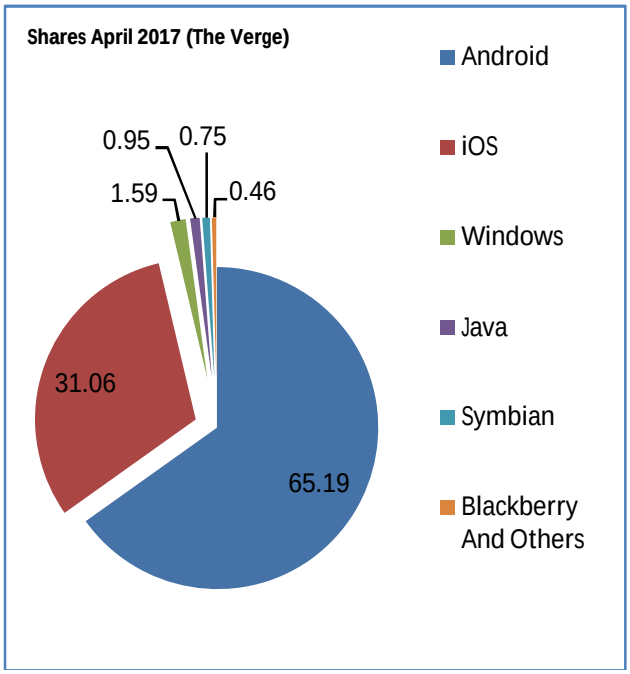

Figure 1: Share of Various OS User April 2017(Millions)

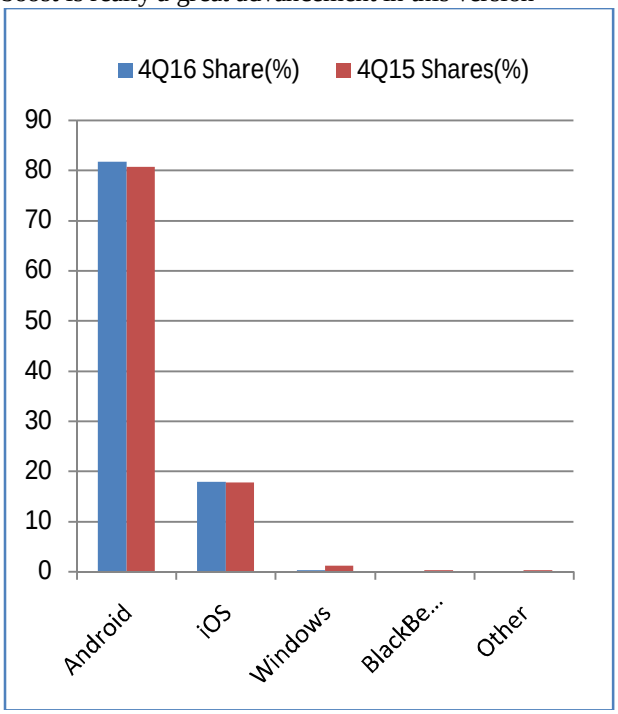

Figure 2: Numbers of Users April 2017 (In Millions)

2. Availability of Android

Android source is open source and thus freely available for improving and developing. Since android has friendly commercial license, one can do whatever changes and improvement, without the intention of blaming Google, if the devices is bricked or damaged. Though there is one exception i:e the Linux kernel, which is falls under the GNU Public License. Thus the manufacturers had to release their device's Linux kernel source code and OEM code after the product is shipped.

3. Linux Based

Android works on Linux kernel with different kernel for each release versions of android in Linux kernel tree. Google has come up with various versions of Android, each of which is based on different Linux versions that are given below.

Table 1: Distribution of Android

Android Version	Android Name	API Level	Kernel Version
1.0	Alpha	1	1.0
1.1	Beta	2	2.0
1.5	Cupcake	3	2.6.27
1.6	Donut	4	2.6.29
2.0/2.1	Éclair	5-7	2.6.29

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor & UGC Approved Journal)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

2.2.x	Froyo	8	2.6.32
2.3.x	Gingerbread	9,10	2.6.35
3.x.x	Honeycomb	11-13	2.6.26
4.0.x	Ice Cream Sandwich	14,15	3.0.1
4.1.x	Jelly Bean	16	3.0.31
4.2.x	Jelly Bean	17	3.4.0
4.3	Jelly Bean	18	3.4.39
4.4	Kit Kat	19,20	3.10
5.x	Lollipop	21,22	3.16.1
6.0	Marshmallow	23	3.18.10
7.0	Nougat	24	4.4.1
7.1	Nougat	25	4.4.19

4. Architecture of android

The Android operating system consists of five divisions -

- **Applications –** It include basic application such as email client, SMS application, calendar, maps, browser, contacts, and others. These applications are basically written in JAVA Programming.
- **Application Framework** - The developers develops framework architecture APIs used by application bases to simplify and to reuse the components, application can have lots of capabilities so that any other application can make use of those capabilities . User also uses this mechanism which allows components to be replaced. These applications work on a set of following services which include
    - Content Providers
    - Resource Manager
    - Notification Manager
    - Activity Manager
    - Reusable Layout/Views
- **Libraries** - Various components are based on a set of core C / C + + libraries in android. These functionalities are exposed to the developer through the Reusable Android application framework. Some core libraries are specifically designed are bionics, media support for audio, video, and SQlite for database
- **Runtime Android –** Except core libraries, Android have a base library which is based on Java. Each Application runs its own processes with its own instances in Dalvik Virtual Machine (DVM) featuring Just in time (JIT). DVM uses Dalvik Executable(.dex) which is faster to load, uses less memory and can run on multiple DVMs
- **Linux Kernel –** The intermediate layer known as Linux kernel which is in between hardware and software stack. The OS depends on Linux kernel for its services for security memory management, process management, Hardware abstraction layer (HAL) compilation, and network stack and driver model. Approximately there are 3.7 million lines in XML, 3.1 million lines in C, 2.5 million lines in Java, and 1.9 million lines in C++.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor & UGC Approved Journal)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

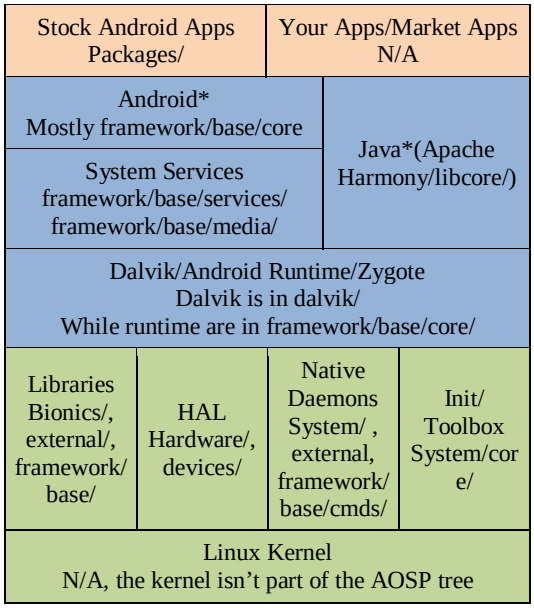

Figure 3: Architecture of Android

II. RELEVANT WORK

1. According to the latest Research investigation by *Hossain Shahriar et al., [1]* Android mobile have limited memory resources which generally have its own garbage collector. The memory leak arises when the system is out of memory which generally creates much kind of vulnerabilities. With the following problems they proposed various solutions like Application fuzzing, Resource fuzzing, API fuzzing etc. They finally conclude that as the leak pattern to test case so that is automatically detected when the leak happen through API or resource invocation.
2. *Zheng Wenxuan, et al., [2]* modified android application and system performance for fast boot and fast shutdown of the devices. Thus this paper introduced various methods for booting up a device 66% faster than a usual boot up process. Some methods which includes creating a Virtual file system (VFS), instead of a physical memory or flash memory.
3. *Abhyudai Shanker & Somya Lai [3]* shows the basic android porting concepts. This paper proposed various concepts for porting an android operating system into a particular hardware, using layered architecture of android and a developer for gaining access for working of system. They described the file system used while porting the system for other devices, which encouraged the Orignal equipment manufacturers (OEMs) to base their mobile for android devices.
4. *Hao Hu & Jinran Song [4]* as per rapid growth of mobile devices, they proposed system oriented architecture (SOA) based design methods for integrating the application with the exiting application to work it in efficient and fast way.
5. *Benjamin Henne et al., [5]* as per limitation of mobile devices that application can't be given independent permission to access the location which in result revealing the user's location. They developed algorithm which when integrate with the application obfuscate the location of user while using various application seamlessly with some false location.
6. *Geunsik Lim, et al., [6]* Proposed various partitioning technique for enhancing the application performance in android platform. Thus enhancing performance for reducing the memory reclamation time for each application.
7. *Hyun-Joo Yoo, et al., [7]* researched that as Android application's configuration requires a bitmap handle, and also the network and the various sensors and consists of various processes, such as DB, and these features

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor & UGC Approved Journal)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

must be processed in real-time. Thus they proposed, various techniques based on Android OS version changes with changes in memory management policies and android garbage collection algorithm for applications with enhanced performance analysis technique.

8. *Minho Ju, et al., [8]* researched as Mobile devices based on flash memory have unique hardware characteristics. Thus, they propose an efficient memory management module called memory orchestration system (MOS) using dynamic page cache size and killed application counts.
9. *Thiago Soares Fernandes, et al., [9]* proposed as the mobile applications development is guided by the careful attention to a number of non-functional requirements. They present two case studies regarding the performance evaluation process for mobile applications and its impact on development. The 1st case study have System under test (SUT) and Component under study (CUS), various services, metrics, various parameter and factors to be studied to optimize work load characterization and statically relevance while the 2nd case study have code metrics & code refactoring. Thus covering each aspect of Dalvik virtual machine (VM).
10. *Eric Chen, et al., [10]* they proposed that rising popularity of modern mobile phones results in an increased demand for manifold applications for these devices. As a result of their review and taking into account current initiatives and trends in the market, they come up with a novel approach, implementation architecture and a prototype.

III. PROPOSED METHODOLOGY AND DISCUSSION

Android is being widely used in mobile phones such as smart phones and tablets and various other devices. Since the release of android more patches and powerful versions have come up with the support of new functionality, more multimedia processing, and more storage. Thus the number of problem are rapidly increasing day by day some of which are–

1. **Wakelock –** To avoid the battery drain when an android device is left idle it quickly falls asleep, but sometime application needs to wake up the CPU or screen to complete the remaining work. The default setting for this wakelock system is only to sleep when the system is idle for a long period of time. Thus we changed the wakelock counter so that to maximize the battery.
2. **Brightness –** The Brightness of various screen of devices lie between various range depending on the maximum and minimum lumen value provided in the system. Thus changing these values in the integrated source code will change the battery drain.
3. **Importing Processor Features –** Android operating system is made in accordance to every device to be more specific the even the processor is also taken into account while designing it. Thus we will import some of the processor features to make the system more optimized and faster.
4. **Boosting CPU –** Sometime user need more performance for an application then the system allow us to have. Thus by implementing various CPU profiles the user can boost performances for a limited period of time.
5. **Power Whitelist –** Some of the application which is mostly used by the user can be added to the whitelist so that the system can keep the application awake to minimize the wakelock trigger count which can minimize the waste memory lock in the primary memory. Thus adding some application to default whitelist to enable a particular receiver instead of loading the whole application into the memory. This will reduce the wakelocks and thus improving the battery drain.
6. **Idletimer –** The idletimer is a function in which the system cannot find the specific time in which the system will put every application, broadcast receiver, various services, etc on hold till the next wakelock arrives. So adding and idletimer to the whole system the system will go to deep sleep while some of the receivers would be active because of the whitelist.
7. **Limiting Frequency –** The step-up frequency of a system is not defined so when system call a wakelock or an application is triggered by the user the CPU divert all the power for maximizing the CPU frequency for the user resulting in battery drain. Thus limiting the step-up frequency so as to reduce the power consumption in enabling all cores even for a single small application will reduce the CPU load and thus operating more effectively.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor & UGC Approved Journal)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

IV. CONCLUSION

Performing the changes in the basic structure of the operating system is quite challenging and a very complex method. This research demonstrate that all the methods implemented to improve various features will reduce the battery drain, improve cpu performance and will provide a better user experience however it may also cause system instability and cause the hardware crash or failure and isn't perfect in some respects. These methods won't cause an imminent failure but may have some effects on the user's experience. The research empirically support that a user must not make the changes not knowing the results. Thus in future these research must focus the user experience and to improve the stability.

REFERENCES

[1]. Hossain Shahriar et al., "Testing of Memory Leak in Android Applications on High-Assurance Systems Engineering (HASE)", 2014 IEEE 15th International Symposium , pp. 176-183, 2014.
[2]. Zheng Wenxuan et al., "Fastboot and fast shutdown of Android on the embedded system", 2013 IEEE 11th International Conference on Electronic Measurement & Instruments (ICEMI), vol. 2, pp. 1003 – 1008, 2013.
[3]. Abhyudai Shanker & Somya Lai "Android porting concepts" 2011 3rd International Conference on Electronics Computer Technology (ICECT), vol.5 pp. 129 – 133, 2011.
[4]. Hao Hu & Jinran Song "Integration and optimization of Android applications based on service-oriented architecture", 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 2098- 2103, 2016.
[5]. Benjamin Henne et al., "Selective Cloaking: Need-to-know for Location-based Apps", 2013 Eleventh Annual Conference on Privacy, Security and Trust (PST), pp. 19-26, 2013
[6]. G.Lim et al., "Enhancing application performance by memory partitioning in Android platforms" , Consumer Electronics (ICCE), 2013 IEEE International Conference on Consumer Electronics(ICCE), pp. 649-650 , 2013
[7]. Hyun-Joo Yoo et al., "Study of Garbage Collection Performance on Dalvik VM Heap Considering Real-Time Response", 2013 International Conference on IT Convergence and Security (ICITCS), pp. 1-3, 2013.
[8]. Minho Ju et al., "Efficient memory reclaiming for mitigating sluggish response in mobile devices", 2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), pp. 232-236, 2015.
[9]. T.S.Fernandes & Erika, Alvaro Freitas, "Performance Evaluation of Android Applications: A Case Study", 2014 Brazilian Symposium on Computing Systems Engineering (SBESC) , pp. 79-84, 2014.
[10]. Eric Chen et al., "Offloading Android Applications to the Cloud without Customizing Android", Eighth IEEE PerCom Workshop on Pervasive Wireless Networking 2012, Lugano, pp.788-793, 23rd March 2012.