ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

Android Gadgets Apps Scheduling Across Local and Cloud

Tejaswini Satav[1], Pooja Gawade[2], Sonali Awhale[3], Rutuja Hargude[4], Prof. Vijay Sonawane[5]

Student, Department of Computer Engineering, Jspm's Bhivarabai Sawant Institute of Technology and Research, Wagholi, Savitribai Phule Pune University, Pune, Maharashtra, India[1,2,3,4]

Professor, Department of Computer Engineering, Jspm's Bhivarabai Sawant Institute of Technology and Research, Wagholi, Savitribai Phule Pune University, Pune, Maharashtra, India[5]

ABSTRACT: We propose a multi-layer mobile application (app) scheduling method to extend the capability of low-end Android devices. We try to reduce this dilemma by a Multi-layer App Scheduling (MAS) schema, along with a cloud service. For the first layer, we utilize the "freeze" feature of Android to prevent non-essential background activities. For the second layer, itis a network scheduler, which automatically schedules the available apps, together with their data, between local and cloud according to user's personal policy generated by big data analysis. By dynamically scheduling the apps among three states, Quality of experience of a low-end Android device is improved. At the same time, with the help of an app state recovery mechanism, the user can directly access a large number of apps provided by the cloud with consistent app view.

KEYWORDS: App Scheduling, Low-End Device, Mobile Device, Android, Cloud backend Platform, Multi-layer Application Scheduling.

I. INTRODUCTION

The past decade has witnessed the great success of smart devices, such as Android and IOS devices, Besides the mainstream consumer market, there still is a vast market of low-price devices, especially low-end Android devices, in less developed areas. The available resources of those low-end Android devices are constrained either by price or by power capacity multiple running apps may deteriorate Quality of Experience of the user. Moreover, this contradiction between the user requirements and the limited resources is even severe for low-end devices. On the other hand, the people's Requirements of function and performance of end devices are promoted by the rapid development of mobile technology. Therefore we propose an app scheduling schema to extend the capability of android device. Devices. An app, together with its user profile and data, can transit among three states.

For example,
1) When the scheduler predicts that the user may use an app with high probability, it reloads the app from the cloud to the local device in advance.
2) If the app is not frequently-used, the scheduler may freeze or
3) Defrost it on demand.
4) If the storage is running out, the scheduler can offload less important apps to the cloud.

1. Background

A great deal of researches have been carried out to extend the scalability and QoE of smart devices, where some are related to this paper. The edge computing technology puts forward many ways to solve the service cache problem of edge nodes, so as to provide services more accurately and quickly to IoT devices[3]. The mobile computing offloading technology uses the terminal Monitoring program to monitor the performance of terminals [5]. The monitoring &

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

reminding mechanism is an easy way and the mainstream in the software market, and the one of the representatives is 360 Mobile Assistant [2].
The small applet technology such as Android Instant app and We Chat applet framework , implement a “come and go” mode to run apps that allows users to use apps as loading a web page, without having to install. Besides, iOS 11 comes with a new feature named “Offload Unused apps” at WWDC. It lets users free up space for more urgent needs, while still maintaining one-click access to the offloaded apps. The transparent computing technology is also a hot topic in recent years. It is originally a way to solve the performance problems of traditional terminals, but can also be applied to mobile devices. TCID could improve device scalability by efficiently querying and effectively loading application data from the network at runtime.

2. Motivation

System propose to boost the demand of low-capability smart devices and also to improve the performance of smart devices, QOE of User. Moreover, to develop storage extension scheme for low-end smart phones.

II. LITERATURE SURVEY

[1] K. Akherfi, M. Gerndt, and H. Harroud. Mobile cloud computing for computation offloading: Issues and challenges. Applied Computing Informatics, 14(1), 2016

This paper presents the current offloading frameworks, computation offloading tech- niques, and analyzes them along with their main critical issues. In addition, it explores different important parameters based on which the frameworks are implemented such as offloading method and level of partitioning. Therefore, Mobile Cloud Computing (MCC) integrates mobile computing and Cloud Computing (CC) in order to extend capabilities of mobile devices using offloading techniques. Computation offloading tackles limitations of Smart Mobile Devices (SMDs) such as limited battery lifetime, limited processing capabilities, and limited storage capacity by offloading the execution and workload to other rich systems with better performance and resources.

[2] E. J. ONeil, P. E. ONeil, and G. Weikum. The lru-k page re- placement algorithm for database disk buffering ACM SIGMOD Record, 22(2):297306, Jun 1993.

This paper introduces a new approach to database disk buffering, called the LRU-K method. The basic idea of LRU-K is to keep track of the times of the last K references to popular database pages, using this information to statistically estimate the interracial times ofreferencesonapagebypagebasis.AlthoughtheLRU-Kapproachperformsoptimalstatistical inference under relatively standard assumptions, it is fairly simple and incurs little bookkeeping overhead. As we demonstrate with simulation experiments, the LRU-K algorithm surpasses conventional buffering algorithms discriminating between frequently and infrequently referenced pages. In fact, LRU-K an approach the behavior of buffering algorithms in which page sets with known access frequencies are manually assigned to different buffer pools of specifically tuned sizes. Unlike such customized buffering algorithms how- ever, the LRU-K method is self-tuning, and does not rely on external hints about workload characteristics. Furthermore, the LRU-K algorithm adapts in real time to changing patterns of access.

[3] Chun, ByungGon, Maniatis, and Petros. Dynamically par- titioning applications between weak devices and clouds In ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond, pages 15, 2010.

This is the first ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond Mobile cloud computing applications run diverse workloads under diverse device platforms, networks, and Clouds.Traditionally these applications are statically partitioned between weak devices and clouds, thus may be significantly inefficient in heterogeneous environments and workloads. We introduce the notion of dynamic partitioning of applications between weak devices and clouds and argue that this is key to addressing heterogeneity problems. We formulate the dynamic partitioning problem and discuss major research challenges around system support for dynamic partitioning.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

[4] L. Zhang. Power, Performance Modeling and Optimization for Mobile System and Applications PhD thesis, The University of Michigan, 2013.

It focuses on to address these challenges by providing a practical, automatic, efficient, and effective framework to help mobile system and application developers to monitor, understand, and optimize the performance and energy of their target designs. My approach consists of three major steps. We first enable developers understanding of energy implication by providing power models and its construction framework. The provided tool, PowerTutor has demonstrated great value by helping a number of developers to monitor the energy usage of the system and applications. We also enable developers understanding of performance in the mobile paradigm by providing a lightweight, system- wide, fine-grained event tracing system that automatically identifies user perceived latency. We then characterize and analyze the real-world smartphone usage scenario by studying traces gathered from PowerTutor and Panappticon. Our study suggests optimization and design guidance for smartphone designers. Motivated by our findings, we proposed technique to optimize the applications energy consumption while maintaining user perceived performance. The diagnosis framework ADEL (Automatic Detector of Energy Leaks) we develop detects and isolates wasted energy resulting from unnecessarynetwork communication. Our study reveals common inefficient design decision in popular applications which were unknown before.

[5] S. Finley and X. Du. Dynamic cache cleaning on android. In IEEE International Conference on Communications, pages 61436147, 2013

This paper introduces caching technique for Android developer's cache data to improve the performance of their applications. Caching is the technique of transparently storing data such that future requests can be accessed more quickly. At times when a mobile device is not under heavy use the cached data, including sensitive data, can remain on the device for an extensive period of time. This poses a security risk, especially when developers do not take the necessary security measures to protect their users' sensitive information. While there does exist ways to clear application caches built within the Android operating system and third-party applications, these approaches require the user to manually perform these tasks. This paper presents a dynamic cache cleaner that more aggressively pushes out unused cache data. We also present other possible solutions to more effectively manage thecache.

[6] B. Waters, Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization, in Proc. 4th Intl Conf. Practice and Theory in Public Key Cryptography (PKC11), 2011, pp. 53-70.

In this paper three constructions within our framework. Our first system is selectively tested safe under the supposition that we call the parallel exponent Bilinear of e-Hellman (PBDHE), which can be considered a generalization of the BDHE assumption. Our the following two constructs provide performance commercials to achieve testable security respectively under the decisive (slope) Bilinear-Diffie-Hellman Exponent and decisional Bi- linear Diffie Hellman Assumptions. In an IBE or HIBE system, Keys and encrypted texts are associated with the same kind of simple object: identity. In an ABE system, encrypted keys, and texts are associated with more complex objects: attributes and access to formulas.

III. EXISTING SYSTEM

The existing system focus on delivering high quality multimedia services that can guarantee the agreed QoS and save energy on the mobile devices to increase their lifetime. The current methodologies for vitality sparing in cloud data center concentrate on scheduling occupations between processing servers and giving vitality proficiency by Methods for some equipment strategies. For example, we can Easily transfer the data as user contents in the form of video, audio, portable document format, text files etc., into the cloud most commonly used in the low end devices is the Google drive. It can store chat backup history in most of the apps, we can also import our contact information into cloud and extract it whenever required. Though it is available on cloud we can access it on any Platform or device

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

without any constraint on exportation. It also suggest you to remove or uninstall the particular application which is seldom use by with the help of data analysis.

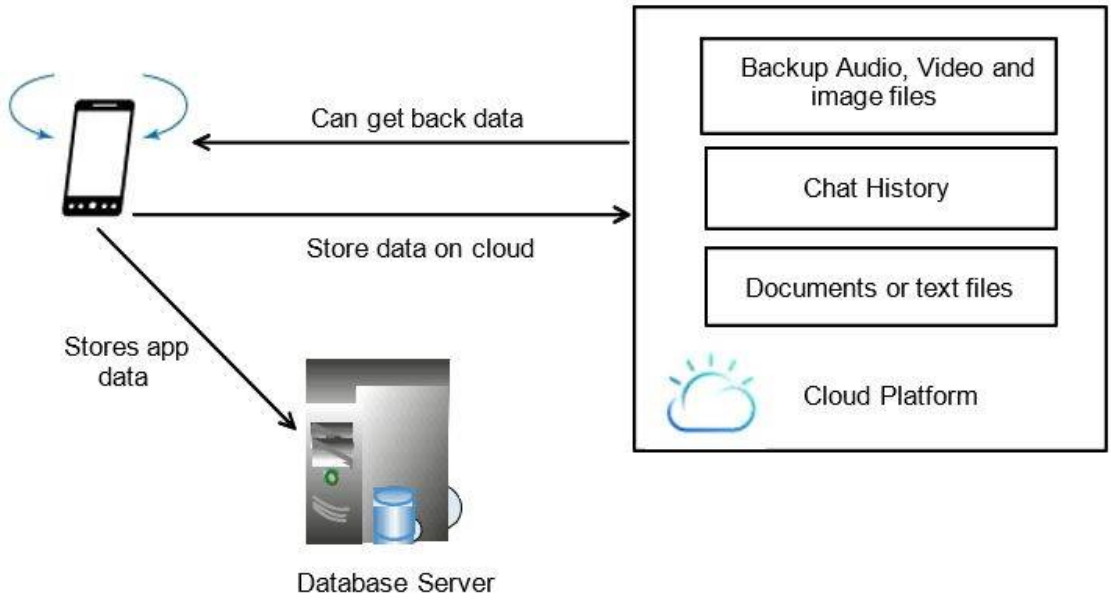


Fig 1. Existing System

Disadvantages of Existing System

1) It can remove application by analyzing but cannot restore the application.
2)Doesn't provide mechanism to schedule application data and installation file on cloud platform.
3)No mechanism to provide backup of data on cross platform.

IV. PROPOSED SYSTEM

Propose multilevel app scheduling system consists of three parts: a local app manager (LA Manager), an application level network scheduler (ALN Scheduler), and a Backend Platform in the cloud. LA Manager and ALN Scheduler are built on top of the original Android system without modification, Communicating with Backend Platform in the cloud side with remote APIs. LA Manager is the local app manager, consisting of three components: a launcher UI, an auxiliary processing service and a cache. The launcher UI is the home page deployed in the local device to provide a unified app access entry. For offloaded apps, a small cue is covered on top of the original app icon to indicate its location. Users can choose an app from the app list to run, no matter it is in the local or in the cloud. If the app is deployed locally, it will start up directly. If the app is in the cloud, it will send a request to ALN Scheduler to start an app reloading process immediately.

ALN Scheduler is the general app scheduling manager, who executes app management actions by calling system tools And auxiliary processing service provided by LA Manager. Users can also set the app display policy and the app execution policy by a pop-out menu when press and hold an app. The auxiliary processing service follows the instructions from ALN Scheduler to perform auxiliary works before or after any app management action. Main services of it includes Fetch (or push) the backed app installation file from (or to) the cloud, and Send response to ALN Scheduler when the Downloading (or uploading) process is completed for app offloading (or reloading).

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

Advantages of Proposed System:
1) Improve Quality of Experience of local devices with the help of cloud computing.
2) Resulting in a better performance of the device.
3) The user no longer has to uninstall apps due to storage budget.
4) Storage is extended with the help of cloud computing platform.

Applications

1. Can be used and accessible in any
2. Android device with lower or latestversions.
3. Especially can used in any device with a low storage capacity and without flexibility to increase storage space of the device.
4. Useful for mobile users in less developed areas who uses low end smartphonedevice.

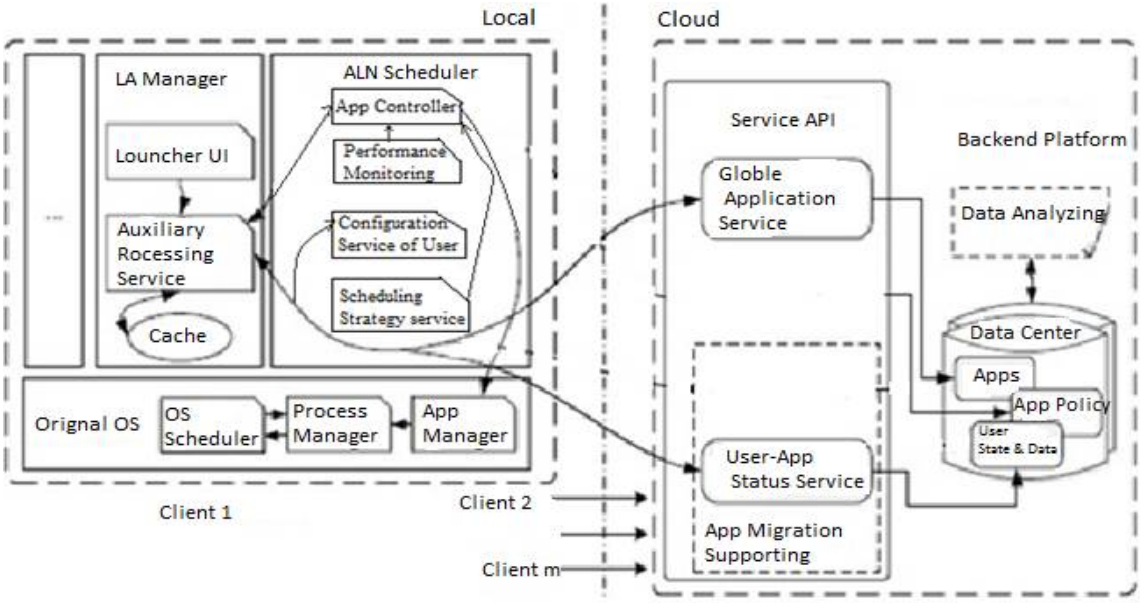


Fig 2: Proposed Architecture

V. IMPLEMENTATION

Algorithms Details

LRU Algorithms

Least Recently Used (LRU) is effective scheduling method to list apps according to their last access time. LRU evicts the app that was used the longest time ago and Keep track of when apps are referenced to make a better decision. Use past behavior to predict future behavior. LRU uses past
Information, while opt uses future information.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

Implementation –

1) Every time an app is accessed,
2) Record a timestamp of the accesstime.
3) When choosing an app to evict, scan over all apps and throw out app with oldest timestamp.

Pseudo Code -

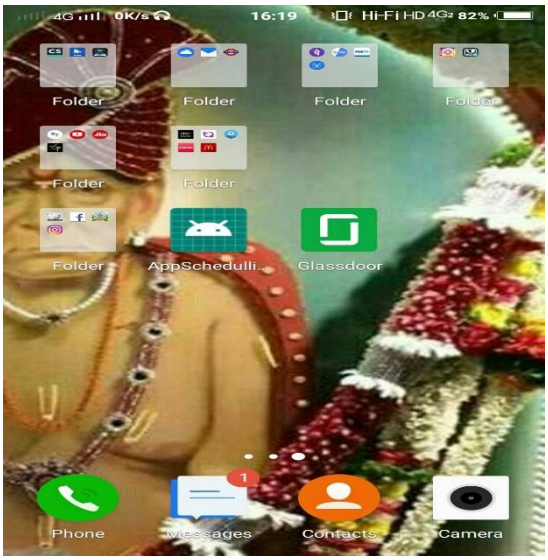
LRU (app a)
If a is in the buffer then LAST (a) = current time;
Else
i) Min = current time +1;
ii) For all apps b in the bufferdo
a) If (LAST(b) < min) victim = b

Min = LAST(b)

iii) If victim is dirty then flush it todisk
iv) Fetch a into the buffer frame held byvictim
v) LAST(a) = current time

VI. RESULTS

1) Mobile Screen prompt allow us to select the launcher from the list.
Which will run on top of the android operating system



2) When you open application it gets connected to server within couple of seconds and display a message ”Connected to server”.And Enables user to access the application

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

3) After opening application it shows all application of mobile in grid view after pressing the selected application it display settings as configure,push,pull and backup and uninstall.

4) After clicking on push button if configurations settings are not done it displays a message "..." else it allows to push the application data to cloud by selecting what data you need to store on cloud.

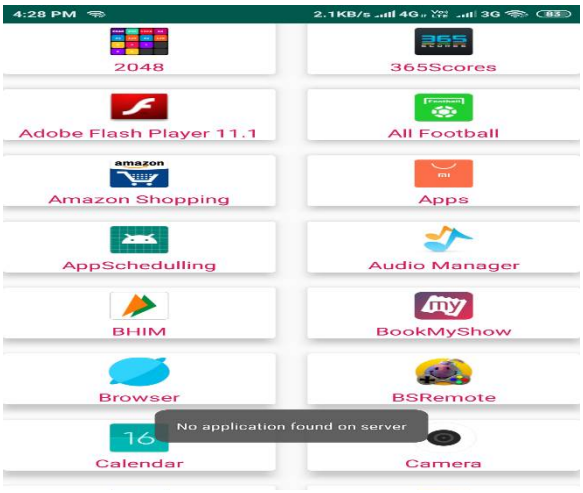
ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

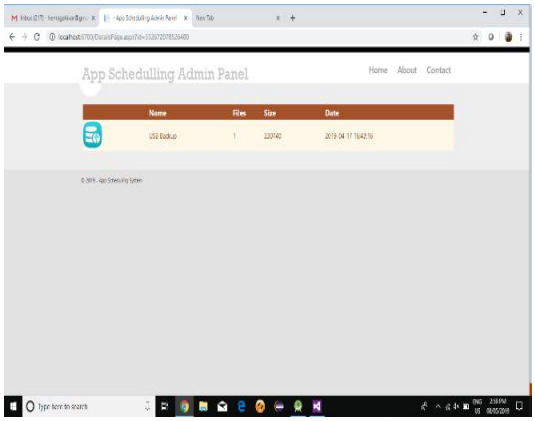
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019



5) When user pushes data to cloud it displays user imei number, add application icon which is pushed and files size of that application in App Schedulling Admin Panel



VII. CONCLUSION

We propose a multi-layer app scheduling schema to extend the capability of low-end Android devices. We find that the increasing number of installed apps may degrade Quality of experience of the user, which is even worse on low-end devices. We propose to improve execution time of local devices with the help of cloud computing. With the help of the powerful Backend Platform in the cloud, Quality of a low-end Android device can be enhanced by proactively scheduling apps based on the analysis of user behavior, device characteristics, etc. At the same time, the user is not needed to be aware of the apps' status, and can directly execute any app provided by the cloud.

REFERENCES

[1] K. Akherfi, M. Gerndt, and H. Harroud. Mobile cloud computing for computation offload- ing: Issues and challenges. Applied Computing Informatics, 2016.
[2] Testing ui performance. https://developer.android.com/training/t
[3] E. J. ONeil, P. E. ONeil, and G. Weikum. The lru-k page replacement algorithm for database disk buffering. ACM SIGMOD Record,Jun 1993.
[4] E. J. ONeil, P. E. ONeil, and G. Weikum. The lru-k page replacement algorithm for database disk buffering. ACM SIGMOD Record,Jun 1993.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

[5] Chun, ByungGon, Maniatis, and Petros. Dynamically partitioning applications between weak devices and clouds. In ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond, pages 15, 2010.
[6] S. Finley and X. Du. Dynamic cache cleaning on android. In IEEE International Confer- ence on Communications, pages 61436147, 2013.
[7] Y. Gao,W. Dong, H. Huang, J. Bu, C. Chen, M. Xia, and X. Liu. Whom to blame? automatic diagnosis of performance bottlenecks on smartphones. IEEETransactionsonMobileComputing, 16(6):17731785, Jun2017.