International Journal of Engineering Technology, Management and Applied Sciences

IJETMAS

www.ijetmas.com February 2016, Volume 4, Issue 2, ISSN 2349-4476

Energy Conservation in Android using Computational Offloading Energy Efficient Downloader

Abhijeet Panchal
Dept. of Computer Engineering
AISSMS's Institute of Information Technology
Pune, India

Vishal Divekar
Dept. of Computer Engineering
AISSMS's Institute of Information Technology
Pune, India

Bharat Bansode
Dept. of Computer Engineering
AISSMS's Institute of Information Technology
Pune, India

Hasan Mujawar
Dept. of Computer Engineering
AISSMS's Institute of Information Technology
Pune, India

Guided By Prof. Girish Navale

Abstract — *Development of new and innovative android applications made energy consumption a very important issue. The full potential for android phones may be constrained by certain technical limits such as battery life and computational performance. Modern mobile applications have more powerful functions but need larger computation and energy resources, which consume more battery power. Using the proposed offloading framework, mobile devices can offload computational intensive workload to other devices to save battery energy consumption and reduce the execution time. JADE (Java Agent Development Framework) allows software developers to easily build and deploy services on the servers to support mobile devices to run computationally intensive tasks. Compared with other offloading schemes for android phones, the JADE enables developers to choose which parts of the codes are supposed to be offloaded. We proposed a Downloader app which is capable of offloading the downloading task to other device, based on the resources available with the other device. Experiments showed that the proposed offloading scheme can successfully reduce computational time and battery energy consumption.*

Index Terms— Code offload, Distributed Computing, Scheduling, Client/Server, Energy Management

I. INTRODUCTION

In the world of smart systems, everyday thousands of android applications are uploaded on the play store, they require many hardware resources to execute applications along with enough battery power. But battery life becomes a biggest obstacle in android phones, which causes the application perform poorly. In order to overcome this drawback we have designed a system which is capable to reduce energy consumption by transferring (Offloading) code to other remote location with resource rich servers.

It is seen that most of the android devices spent much of their time in fetching content or resources from the internet, which consumes much of the energy of the android devices. In this paper we present JADE which allows mobile devices to offload computation such as downloads or any other remotely tasks. JADE is energy aware framework which detect the battery capacity of the users, hardware resources and communication resources such as Network, Bandwidth, Download and Upload speed. Based on this resources JADE decides where to execute user application. It is required that the downloader application is available on both devices such that the devices can communicate with the help of functionalities provided by the application.

Our contributions are summarized as follows:

1. We present a downloader which is capable of offloading computation from an android device to another android or non-android device.

2. We made a multi-level scheduling algorithm which detect status of mobile devices and perform computational offloading on resource rich device.

II. COMPUTATIONAL OFFLOADING

Computational offloading means transferring of certain computing tasks to another platform. Offloading is necessary to overcome the hardware limitation, it can be used to minimize the burden on the processors by transferring the computing task to another devices. In this paper we use computational

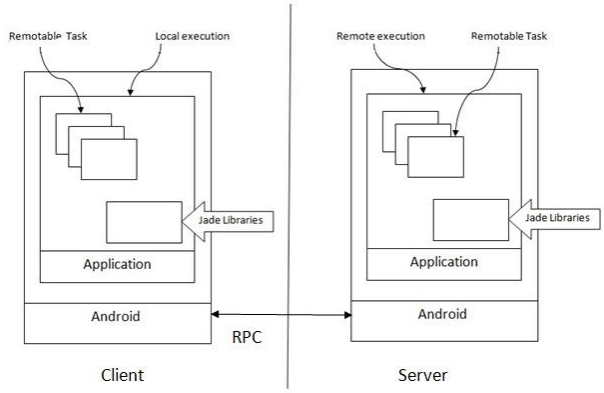
offloading to transfer the task of downloading to another devices. Computational offloading can be performed by using the libraries provided by JADE which provide the necessary functions and system calls to share the data required for computational offloading. The JADE is designed to support mobile agents where the agents can communicate with each other.

III. SYSTEM DESIGN

Here we present the architecture of the proposed system which demonstrates the communication between two devices using JADE environment to support Distributed Computing on mobile devices. Computational offloading is a very efficient technique to reduce energy consumption and decrease the execution time. But computational offloading requires special skills of client/server and distributed programming, which is a very challenging task for the developers. Hence we designed the system to minimize the extra efforts which provides the developers with the following functionalities.

1. Runtime Engine which provides the framework needed for wireless communication, computational offloading and device profiling.
2. Programming model to build mobile application with computational offloading capability.

The overall architecture of the proposed application is as shown in the following figure.



IV. IMPLEMENTATION

While implementing of the project we are going to use following technical details of resources with regards to device which would help us for success in getting desire result.

1)Profiler

2)Optimizer

3)Communication Manager

Profiler: When distance process is created ,JADE has to take decision where should be process execute weather on native device or distance device. Such decision are judge by the Device profiler while transferring process from one device to another. Profiler continuously sensing the device status for getting information such as charging event, battery level, number of task executing on CPU and their quantity Its also gather information of distance task such as time required for execution, weight of process, and power expenditure. Two types of procedure available for measuring power consumption online and offline. offline procedure is preferred under laboratory condition which need outside tool for measurement. Its also need person with skill who can calculate very accurately. Online process uses battery monitoring unit for collecting information regards with battery status. Online process does not provide accurate result because time required to update device status is much lower than outside tools. Inspite of that JADE profiler prefers online method for calculating energy uses.

Program profiling: While generation of distance process program profiler define unique name for each task. During execution of distance task profiler collect status information such as Execution time and size of the process. after that above collected data is forwarded to optimizer for further decision. Based on program profiler information decision is made weather task should be executed on native device or distance device. Also this information is stored in database for future reference. Whenever the same process executed jade decides whether it would be worthwhile to offload process or not.

Device Profiling: In this process device profiler frequently monitors device information such as Battery, Wireless connection status. Because different kind of devices wireless connection can be alter as soon user move from current location to another location. Fresh information about wireless connection status is important in order to offload process. Whenever job transfer from one device to another JADE measure duration required for transferring the process while considering bandwidth and latency in wireless connection.

Optimizer: The intention behind the JADE optimizer is to correct distance process in order to offload to server while keeping mind that strategy which would minimize application energy expenditure. Optimizer make the decision for offloading based on data forwarded by profiler. Nature of task decided where to process actually executed. The code has be offloaded only when energy expenditure for execution is more than energy worth required to transfer it. In order to offload the process a class must consist of one of the two interface such as RemotableTask interface and RemotableGenTask interface.when class contain android device code In order to offload process it should implement RemotableTask interface which ensure that task will be offloaded to android server. If a class consist of java code then user need to implement RemotableGenTask so the code could be transfer to appropriate server. In Jade class which extend ReomotableTask interface that will be referred as remotable class. Following types code must be executed locally.

1)code which creates user interface of the application 2)Code related to user interaction 3)Code which access special hardware that not available on server 4)those code which not suitable for re-execution.

Communication Manager: The related to code offloading is tackle by communication manager. Code manager for an interface for code execution and code offloading. Communication manager decides code offloading base on following information.

1)It store information about code offloading in code offload table. Goal of this information is identify status of code offload 2)transfer the code to the server 3)Server resend the offloaded code to client after execution 4)The client accept this information and store in table. For distance failure of task communication manager assure the correctness of application for example during distance process execution if wireless connection loss then client communication manager execute the process once again locally while sever communication manager stops the execution.

V. EVALUATION

In this section we are going to test the ability of JADE programming model. For this purpose we have developed one download manager using JADE programming mode which have process offloading capability. Our android application will be installed on different device. Here we are using wifi connection for detection of near by device and to transfer the process to appropriate device. The developed application will continuously check the status of device for offloading the device. For example in download manager we have one task which would have size of 100mb while in the middle battery goes low for example if task is downloaded about for 80mb then as soon as battery level goes down then remaining process will be offloaded to nearby device. After completion of downloading the task will returned to original device. For the detection battery level we are using different methods provided by android.

VI. FUTURE SCOPE

In future we can extend the process offloading by connecting to cloud platform. Cloud has ability to provide exetensible and powerful operation mechanism which suitable for process offloading. Today large number of device are moving from 3G device to 4G device so cloud is best platform for offloading the process. Another concern issue is security JADE transfer code to authorized device but if multiple device accessing the single server running native code and distance will interfere with each other.

VII. CONCLUSION

In this paper we have presented the new way to perform Distributed Computing on Android devices using JADE, which have the capability to reduce the energy consumption and speed-up the execution of tasks. We have implemented the jade framework for the creation of the downloader application for the Android OS, which can transfer the process of downloading to other android or non-android device over the Wi-Fi network. The developed application is a very reliable and efficient downloader which perform the device profiling i.e. checking the status of the device and then based on this check, decides whether the task should be offloaded to other device or should run on the same device. The decision of execution is based on the algorithm which we have designed especially for this application. Algorithm designed by us schedules the downloads that are currently being downloaded at runtime. Results showed that the JADE can efficiently reduce the

energy consumption and save up to 30% of energy, and we expect our downloader application to be helpful in conservation of energy resources for android devices.

VIII. REFERENCES

[1] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: Making smart phones last longer with code offload. In MobiSys, 2010.

[2] B. Chun, S. Ihm, P. Maniatis, M. Naik, A. Patti. Clone Cloud: Elastic Execution between Mobile Device and Cloud. In ACM Eu-roSys, 2011.

[3] H. Qian and D. Andresen. Jade: An Efficient Energy-aware Computation Offloading System with Heterogeneous Network Interface Bonding for Ad-hoc Networked Mobile Devices. In Proceedings of the 15th IEEE/ACIS International Conference on Soft-ware Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014.

[4] H. Qian and D. Andresen. Extending Mobile Device's Battery Life by Offloading Computation to Cloud. In Proceedings of the 2ndACM International Conference on Mobile Software Engineering and Systems (MOBILE Soft), 2015.

[5] H. Qian and D. Andresen. An Energy-saving Task Scheduler for Mobile Devices. In Proceedings of the 14th IEEE/ACIS International Conference on Computer and Information Science (ICIS),2015.

[6] H. Qian and D. Andresen. Emerald: Enhance Scientific Work-flow Performance with Computation Offloading to the Cloud. In Proceedings of the 14th IEEE/ACIS International Conference on Computer and Information Science (ICIS), 2015.

[7] Q. Chen, H. Qian et al. BAVC: Classifying Benign Atomicity Violations via Machine Learning. In Advanced Materials Research, Volumes 765-767, pp. 1576-1580, Sep, 2013.

[8] F. Wei, S. Roy and S. Ou. Amandroid: A Precise and General Inter-component Data Flow Analysis Framework for Security Vet-ting of Android Apps. In proceedings of the 2014 ACM Conference on Computer and Communications Security. 2014.

[9] L. Peng, Y. Yang et al. Highly Accurate Video Object Identification Utilizing Hint Information. In proceedings of the International Conference on Computing, Networking and Communications (ICNC), 2014.

[10] S. Zhang, X. Zhang and X. Ou. After We Knew It: Empirical Study and Modelling of Cost-effectiveness of Exploiting Prevalent Known Vulnerabilities Across IaaS Cloud. In proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIACCS), 2014.