CWC: A DISTRIBUTED COMPUTING ENVIRONMENT USING ANDROID MOBILE PHONES

Kanimozhi.R[1],Bakiadarshani.J[2], Bairavi.G[3],Vaishnavi.G[4]

[1]*Assistant Professor,Department of Information Technology, A.V.C. College Of Engineering,Tamilnadu,India.*
[2] *Student, Department of Information Technology, A.V.C. College Of Engineering,Tamilnadu,India.*
[3] *Student, Department of Information Technology, A.V.C. College Of Engineering,Tamilnadu,India.*
[4] *Student, Department of Information Technology, A.V.C. College Of Engineering,Tamilnadu,India.*

ABSTRACT

Every night, a large number of idle smartphones are plugged into a power source for recharging the battery. Given the increasing computing capabilities of smartphones, these idle phones constitute a sizeable computing infrastructure. Therefore, for an enterprise which supplies its employees with smartphones, we argue that a computing infrastructure that leverages idle smartphones being charged overnight is an energy-efficient and cost-effective alternative to running tasks on traditional server infrastructure. While parallel execution and scheduling models exist for servers (e.g., MapReduce), smartphones present a unique set of technical challenges due to the heterogeneity in CPU clock speed, variability in network bandwidth, and lower availability compared to servers. In this paper, we address many of these challenges to develop CWC—a distributed computing infrastructure using smartphones. Specifically, our contributions are: (i) we profile the charging behaviors of real phone owners to show the viability of our approach, (ii) we enable programmers to execute parallelizable tasks on smartphones with little effort, (iii) we develop a simple task migration model to resume interrupted task executions, and (iv) we implement and evaluate a prototype of CWC (with 18 Android smartphones) that employs an underlying novel scheduling algorithm to minimize the makespan of a set of tasks. Our extensive evaluations demonstrate that the performance of our approach makes our vision viable. Further, we explicitly evaluate the performance of CWC's scheduling component to demonstrate its efficacy compared to other possible approaches.

Keyword *: Smartphone, Distributed Computing, Scheduling***.**

1. INTRODUCTION

Today, a number of organizations supply their employees with smartphones for various reasons [1]; a survey from 2011 [2] reports that 66% of surveyed organizations do so and many of these organizations have 75–100% of their employees using smartphones. For example, Novartis [3] (with 100,000 employees in 140 countries) handed out smartphones for its employees to manage emails, calendars, as well as information about health issues; Lowe's [4] did so for its employees to have real time access to key product information and to allow managers to handle administrative tasks. In this paper, we argue that in such settings, an enterprise can harness the aggregate computing power of such smartphones, to construct a distributed computing infrastructure. Such an infrastructure could reduce both the capital and energy costs incurred by the enterprise. First, this could reduce the number of servers to be purchased for computing purposes. For example, Novartis awarded a contract of $2 million to IBM to build a data center for their computational tasks [5]. If they could exploit the smartphones handed out to their employees to run some portion of their workload, it is conceivable that the cost of their computing infrastructure could have been reduced. Due to recent advancements in embedded processor design, now a smartphone can replace a normal desktop or a server running a dual core processor for computation. According to Nvidia, their Quad Core CPU, Tegra 3, outperforms an Intel Core 2 Duo processor in number crunching [6]; for other workloads, one can expect the performance of the two CPUs to be comparable. Our second motivation for the smartphone-based computing infrastructure is that the enterprise could benefit from significant energy savings by shutting down its servers by offloading tasks to smartphones. The power consumed by a commercial PC CPU such as the Intel Core 2 Duo is 26.8W [7] at peak load. In contrast, a smartphone CPU can be over 20x more power efficient, e.g., the Tegra 3 has a power consumption of 1.2W [7, 8]. Since their computing abilities are similar, it is conceivable that one can harness 20 times more computational power while consuming the same energy by replacing a single server node with a plurality of smartphones. In fact, to harness the energy efficiency of embedded processors, cloud service providers are already pushing towards ARM-based data centers [9]. The construction and management of such a distributed computing infrastructure using Smartphone's however, has a number of associated technical challenges. We seek to articulate these chal- 193 lenges and build an efficient framework towards making such a platform viable. In particular, the biggest obstacles to harnessing Smartphone's for computing are the phone's battery-life and bandwidth. If a Smartphone is used for computing during periods of use by its owner, we run the risk of draining its battery and rendering the phone unusable. Further, today data usage on 3G carriers are typically capped, and thus, shipping large volumes of data using 3G is likely to be impractical. Thus, our vision is to use these Smartphone's for computing when they are being charged, especially at night. During these periods, the likelihood of active use of the phone by its owner will be low. Moreover, the phone will be static and, will likely have access to Wi-Fi in the owner's home (today, 80% of the homes in the US have Wi-Fi connectivity [10]); this will both reduce fluctuations in network bandwidth, and allow the transfer of data to/from the Smartphone's at no cost.

2. EXISTING SYSTEM

The existing has Enterprise computing using smart phones. The system that is closest in spirit to CWC is CANDIS, where the authors proposed using employee smart phones (being charged) for executing enterprise applications. 2 Similar to our effort in CWC, they implemented an execution environment for Android that allows for running desktop Java applications on smart phones in an automated fashion. They also made similar observations about scheduling tasks based on computational capabilities of smart phones. While we envision similar applications and system implementation in CWC, we provide a sophisticated algorithm that minimizes the make span based on both CPU capabilities and bandwidths of smart phones, which has not been explicitly addressed in CANDIS.

3. ARCHITECTURE

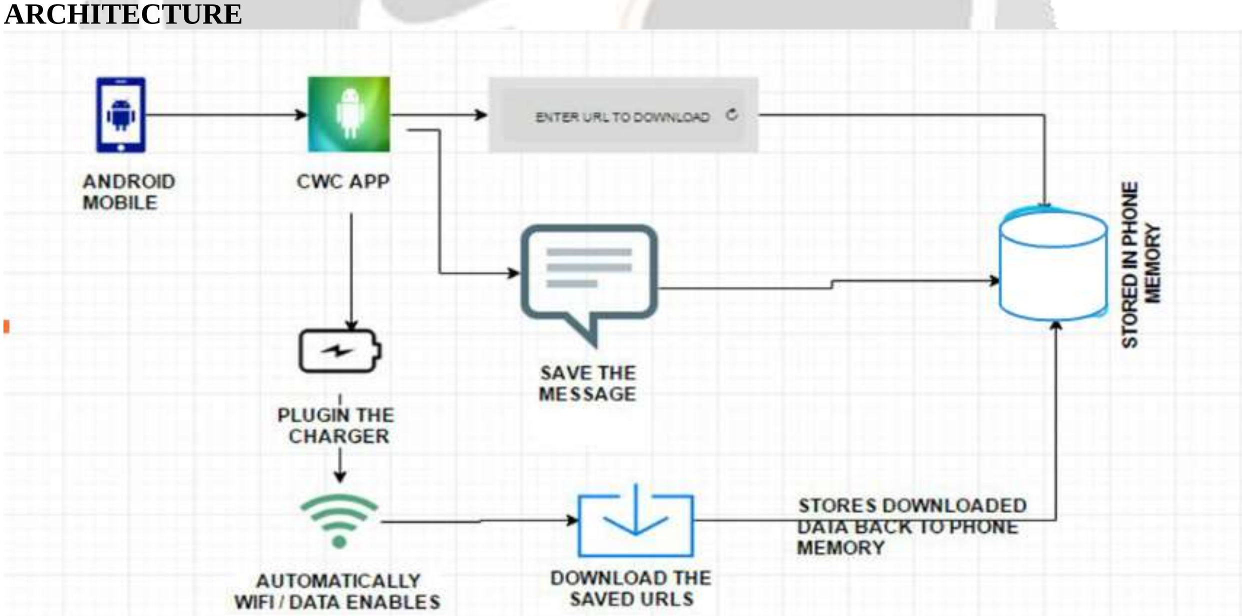

Fig -1:Architecture of CWC

4. PROPOSED SYSTEM

In this paper, we envision building a distributed computing infrastructure using smart phones for enterprises. Our vision is based on several compelling observations including (a) enterprises provide their employees with smart phones in many cases, (b) the phones are typically unused when being charged, and (c) such an infrastructure could potentially yield significant cost benefits to the enterprise. We articulate the technical challenges in building such an infrastructure. We address many of them to design CWC, a framework that supports such an infrastructure. We have a prototype implementation of CWC on a test bed of 18 Android phones. Using this implementation, we demonstrate both the viability and efficacy of various components within CWC.

5. SYSTEM DESIGN

5.1 User Login Module

Android application that develop, gather charging statistics. A typical user charges her phone for up to eight hours each night on average, which offers a long idle period for computing.

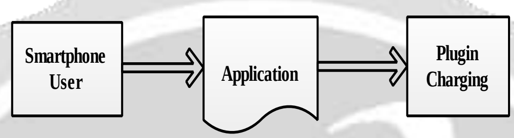

Fig -2:User login module

5.2 Task Scheduling module

Design a scheduler that minimizes the completion time (makespan) of the jobs at hand, taking into account both the CPU and bandwidth available to each smartphone. Since optimal job allocation is NP-hard, show that it outperforms other heuristics.

Fig -3:Task Scheduling module

5.3 Task Execution Module

Remote task execution on phones can be automated, without the user downloading and installing an application. While recognize the potential privacy implications of bypassing user control, simply assume here that an enterprise would not run malicious tasks on its employees' smartphones.

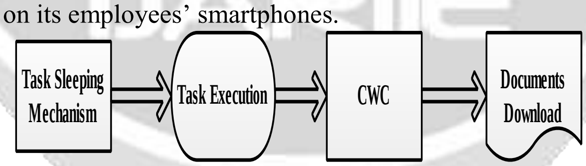

Fig -4:Task execution module

5.4 Computing infrastructure module

Blindly executing tasks for extended durations on a smartphone being charged, can prolong the time taken for the phone to fully charge. Show that intensive use of a phone's CPU can delay a full charge by 35 percent. Design and implement a task sleeping mechanism, which ensures that task executions do not impact the charging times.

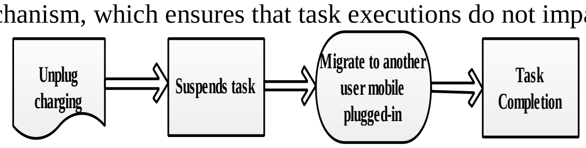

Fig -5:Computing Infrastructure module

6. FUTURE WORK

Focus on developing heuristic based job scheduling and resource management algorithms and validate the practicality of these algorithms in real cloud environment using realistic cloud environment. In this domain can be extended by developing static and dynamic stochastic simulation experiments for different processing environments such as cloud, embedded systems which will be asset to IT industry in order to minimize the cost of software products, Software Risk Management issues and congestion in network management and other developments related to web based applications.

7. BENEFITS

Savings in infrastructure costs: Since the idle compute resources on already deployed smartphones are used, the cost borne by corporations to bootstrap the platform will be minimal in comparison to that in setting up a similar service on a server-based infrastructure. Companies have either to invest in buying hardware (e.g., servers, switches) or in outsourcing their tasks to third party cloud services. In addition, establishing computing infrastructure requires careful planning with regards to factors such as space, federal and state regulations, and the provisioning of power and cooling support. In contrast, the use of a smartphone infrastructure obviates such considerations. To leverage existing smartphones as the elements of a utility computing service, an enterprise will need no more than a central, lightweight server to identify idle resources and allocate them to computational tasks.

8. CONCLUSION

In this paper, we envision building a distributed computing infrastructure using smartphones for enterprises. Our vision is based on several compelling observations including (a) enterprises provide their employees with smartphones in many cases, (b) the phones are typically unused when being charged, and (c) such an infrastructure could potentially yield significant cost benefits to the enterprise. We articulate the technical challenges in building such an infrastructure. We address many of them to design CWC, a framework that supports such an infrastructure. We have a prototype implementation of CWC on a testbed of 18 Android phones. Using this implementation, we demonstrate both the viability and efficacy of various components within CWC.

9. REFERENCES

[1]M. Y. Arslan, I. Singh, S. Singh, H. V. Madhyastha, K. Sundaresan, and S. V. Krishnamurthy, "Computing while charging: Building a distributed computing infrastructure using smartphones," in Proc. 8th Int. Conf. Emerging Netw. Experiments Technol., Dec. 2012, pp. 193–204.

[2]J. Dean and S. Ghemawat, "**MapReduce: Simplified data processing on large clusters**," in Proc. 6th Conf. Symp. Operating Syst. Des. Implementation, 2004, p. 10.

[3]T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, "**PRISM: Platform for remote sensing using smartphones**," in Proc. 8th Int. Conf. Mobile Syst., Appl. Services, 2010, pp. 63–76.

[4]E. Oliver, "**The challenges in large-scale smartphone user studies**," in Proc. 2nd ACM Int. Workshop Hot Topics Planet-Scale Meas., 2010, pp. 5:1–5:5.

[5][A. Shye, B. Scholbrock, G. Memik, and P. A. Dinda, **"Characterizing and modeling user activity on smartphones:Summary**," in Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst., 2010, pp. 375–376.

[6]S. Schildt, F. B€usching, E. J€orns, and L. Wolf, "**CANDIS: Heterogeneous mobile cloud framework and energy cost-aware scheduling**," in Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things and IEEE Cyber, Phys. Social Comput., Aug. 2013,pp. 1986–1991.

[7]S. Harizopoulos and S. Papadimitriou**, "A case for micro-cellstores: Energy-efficient data management on recycled smartphones,"** in Proc. Int. Workshop Data Manage. New Hardware, 2011, pp. 50–55.

[8]P. R. Elespuru, S. Shakya, and S. Mishra, "**MapReduce system over heterogeneous mobile devices**," in Proc. 7th IFIPWG10.2 Int. Workshop Softw. Technol. Embedded Ubiquitous Syst., 2009, pp. 168–179.

[9]T. Sekiguchi, H. Masuhara, and A. Yonezawa, "**A simple extension of java language for controllable transparent migration and its portable implementation**," in Proc. 3rd Int. Conf. Coordination Languages Models, 1999, pp. 211–226.

[10]M. Hajjat, X. Sun, Y. E. Sung, D. Maltz, and S. Rao, "**Cloudward bound: Planning for beneficial migration of enterprise applications to the cloud**," in Proc. ACM SIGCOMM Conf., 2010, pp. 243–254.