VERIFICATION OF MOBILE CLOUD COMPUTING USING MODEL CHECKING

Mr. Shivalal D Parmar[1] | Mrs. Shubha Raj K B[2]
[1](Dept of CSE, PESIT Bangalore South Campus, Visvesvaraya Technological University, India, shakashid@gmail.com)
[2](Dept of CSE, PESIT Bangalore South Campus, Visvesvaraya Technological University, India, shubharajkb@pes.edu)

Abstract— *In Mobile Cloud computing, offloading the computation to the remote resources has become a crucial issue at runtime. This project proposes a new methodology that is formal framework, whose objective is to allow runtime support for offloading decisions. By means of a domain specific language (MobiCa), a developer can outline each system and application structure, in terms of involved devices, memory required computational power, code and partitioning. Here, we have used one of the formal verification techniques like model checker that is UPPAAL, which is the framework it provides comprehensive and automatic way of providing decision support for offloading using formal verification technique. In UPPAAL, at runtime analytic traces are generated which are driven by some query verified on the timed automata model associated with the specification of MobiCa language. This technique permits one to reduce memory utilization, battery utilization and guaranteeing the most effective system performance. In this project, navigator application is taken as a case study, probably one of the foremost used applications on mobile devices*

Keywords— UPPAAL; Offloading; MobiCa

1. INTRODUCTION

In the most recent decade, Mobile gadgets have progressively invaded our everyday lives. Advancement in equipment has ceaselessly given effective computational assets in lighter what's more, littler electronic handsets. This new era has advanced equipped has set off large new mobile applications has emerged in the market [3]. The achievement in this market is related to the enhancement in the omnipresent and all time access to the computation and data availability at everywhere.

The greater part of Mobile applications that keep running on a compact gadget are emphatically associated with the system for getting to information, yet keep up the computational load domestically. Because of limited battery life in the handheld devices there exist the large workstations and servers, it is hard to do the computation in the mobile devices hence we are choosing the cloud system. It provides the facility of computation without being worried about the limited energy consumption and size and weight of devices. The paradigm popularly known as “Mobile Cloud Computing (MCC) which is a combination of the mobile device technology and cloud computing infrastructures for the development of mobile applications”4, which helps for mobile devices to access the information at any place and at any time. To explain this in simple words, ability of transforming the data as well as computational power from mobile devices to remote storage called cloud.

The principal clear potential advantage is enhancing stockpiling limit and execution. In reality, regardless of the fact that the computational assets accessible to a cell phone have expanded quickly in the course of recent years, this figuring force is still littler than that of the stationary partners. This is on account of cell phones. And tablets should be abundant littler and lighter than servers and desktop PCs. attributable to this hole within the middle of transportable and cloud getting ready power, C.P.U. escalated applications are often dead abundant faster on remote than on cell phones. Then again, intuitive applications that need few process assets may well be dead on cell phones as fast as on servers. Execution doesn't rely simply on processor speed, however rather in addition on memory, capacity, and therefore the capability of parallelizing over varied centers and servers, and higher associations. These qualities are often essentially given by a typical cloud base with a tiny low venture.

The second advantage is change vitality utilization. Concordant to, mobile phone purchasers discovered longer battery life to be additional important than each single different element [16]. a transportable appliance have to be compelled to disbursement set up its restricted wellspring of vitality shrewdly in request to the touch base to the top of the day while not enervating all its accessible vitality it slow recently.

To overcome with the challenges facing in the mobile cloud computing that are faced by the developers for developing the devices to make them energy efficient as much as possible. For instance, in the mobile devices to reduce the background computation of the mobile hardware we usually enable the power saving mode of the device which leads to reduced performance rate of the device and extends the battery life of the device. By doing this the devices gets slow down the user experience which becomes another crucial issue. Computation offloading is signally AN appealing choice, since utilizing remote calculation and capability enhances execution and, within the meanwhile, spares battery power giving a superior client expertise.

Mistreatment this offloading concept attempting to implement this method of the existing paper’s concept as a technology for easy usage in current technological era.

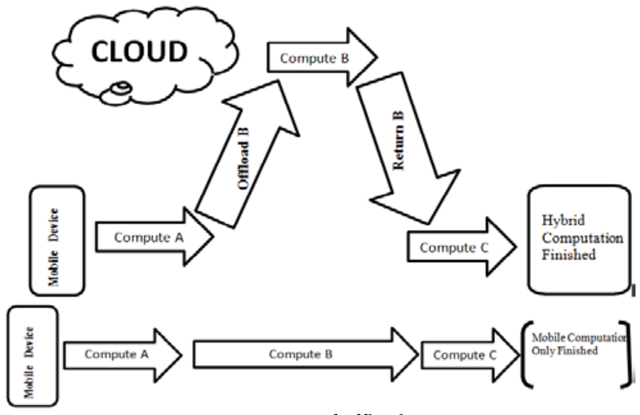


Fig 1: Representation of offloading concept

2. LITERATURE SURVEY

Mobile Cloud Computing (MCC)

MCC: Another perspective for Mobile applications advancement depending on the mix of distributed (cloud) computing frameworks and Wireless Networks. Mobile Cloud computing existing difficulties are confinement of data transfer capacity and correspondence idleness, signal unsettling influence, constrained computational force, memory and battery life, security. One of the good solutions to overcome with the limited battery life and memory and computational power challenge is that offloading the applications to the cloud.

UPPAAL

UPPAAL is a tool kit for acceptance (by means of graphical reproduction) and confirmation (through programmed model- checking) of constant frameworks. It comprises of two principle parts: a graphical client interface and a model- checker motor.

We can display systems of timed automata in UPPAAL, where a timed automata is a finite automata (finite state machine) stretched out with clock variables. The principal part of the area diagrams the finite automata formalism and augmentations.

The systems of timed automata in UPPAAL comprise of simultaneous procedures, every one of them spoke to by a finite automata. The portrayal of the automata of the single procedure is like the depiction of the State Machine in UML.

The menu is depicted in the incorporated help, available through the help menu. The assistance further portrays the utilized grammar and the GUI as a part of subtle element, so this instructional exercise will concentrate on the most proficient method to utilize the apparatus. The three tabs offer access to the three segments of UPPAAL that are the editor, the Simulator and the verifier.

In UPPAAL symbol “!” after any argument refers to sender and symbol “?” after any argument refers to receiver.

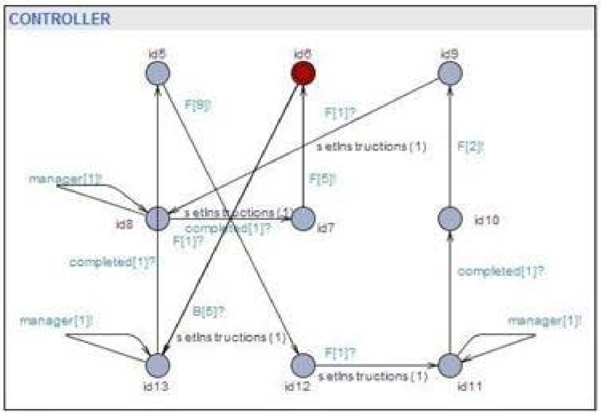
In UPPAAL we can merge the different model to form a large model of real-time system using declaring variable and globally and locally and through system declaration we can merge the different sub models of the real time system which has to use model a whole system.

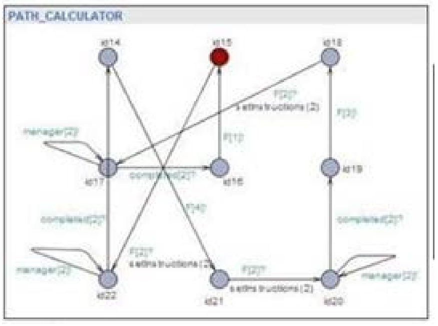
Offloading Concept

Offloading: The procedure of moving the computational force and information stockpiling far from cell phones into the cloud, enhancing the client experience to a wide scope of versatile supporters.

The cloud is by all accounts a superb buddy of portable frameworks, to ease battery utilization on cell phones what's more, to reinforcement client's information on- the-fly. Without a doubt, numerous late works concentrate on structures that empower portable calculation offloading to programming clones of cell phones on the cloud and on planning cloud-based reinforcement frameworks for the

Information put away in our gadgets. Both portable computation offloading and information reinforcement include correspondence between the genuine gadgets and the cloud. This correspondence does absolutely not want free. It costs in wording of transmission capacity (the activity overhead to speak with the cloud) and regarding vitality (calculation and utilization of system interfaces on the gadget). the transition from each state to reach the next by means of instruction sets.





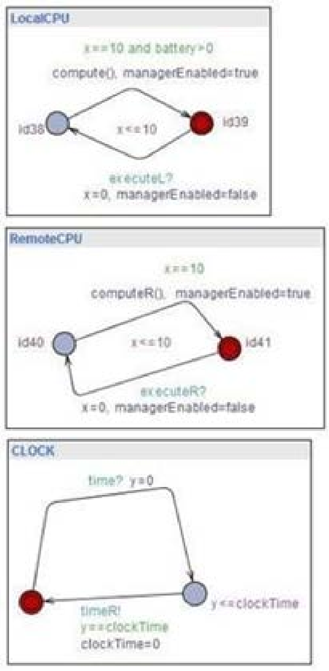


Fig 2: Automata showing semantics of UPPAAL tool.

In manager automaton we have a committed location hence it is an "obs" automaton, where for one possible transition it will always one going out of the committed state. Each transition in committed state will make a time delay to make transition to next state only if the instruction set satisfies the condition given.

The automata named Local will acts different for each of the transition and named as Local(0) to local(9) for each different transitions in the network. The automata's Path Calculator, Traffic Evaluator, Map, navigator, navigation panel, GPS, Manager and Local are the execution phases of the network. The Local CPU, Remote CPU are the completion phases of the network and the clock sets and monitors the time to take transitions from one state to another transition state and also from one automaton to another automaton.

In the previous chapter we put up a clause to describe the detailed procedures and curriculum involved in the proposed system. However in most cases we offer to terminate the procedure of implementing the scenario this is however nonetheless in many situations however, we should prepare a detailed which should be considerable by a layman who does not know about the virtualization. Hence in order to achieve this we provide a detailed preview by visualizing the concepts of flowcharts, dataflow diagrams.

The procedure of the implementation should be prefixed by initiating with the SRS which set up the premise for understanding, practicality, measured quality, and expense between the end users on what the product item is to do.
The complete depiction of the capacities to be performed by the product indicated in the SRS will help the potential client to figure out whether the product indicated addresses their issues or how the product must be adjusted to address their issues. It gives a premise to building up the product plan which decreases the development inconsistencies to a great effort.
Thus it helps in avoiding the misunderstandings in early stage of SDC and also it avoids inconsistencies in early stage of SDC. If problems are detected in early stage then it will be easy to correct and thus provides platform verification and validation.

3. EXPERIMENTAL ANALYSIS AND RESULTS

Here the task performed is to analyses the results that are obtained after performing the operations. The application is undergone through distinct inputs and the results obtained are analyzed for the performance and the accuracy of the results in fig 3.

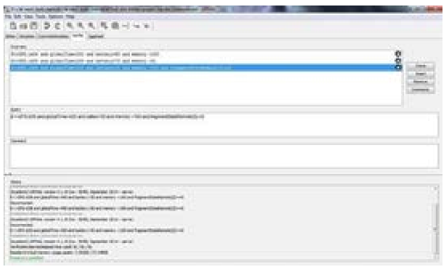


Fig 3: Verification of the results in UPPAAL Tool.

4. CONCLUSION AND FUTURE SCOPE OF WORK

In this paper we have displayed a strategy for giving runtime choice backing to android application. The approach depends on android libraries and web applications, another dialect that core interests on communicating the parts of a MCC framework that are most suitable for offloading. This model can be utilized as multiple applications so as to make strides framework execution for different applications through a single interface which is the future scope of work of this project.
For now, the proposed approach for the offloading choice backing has been actualized as a proof of idea. Subsequently, the enhancement of its execution for multiple web applications is cleared out for future work. In this paper we are concerned about how the proposed methodology can able to exchange to the innovation for one web application. A conceivable thought is to incorporate the choice backing as a component of android application foundation, worked over the working frameworks of the cell phone and the cloud machines.

REFERENCES

[1] Mobile Cloud Computing is a domain name referred from https://en.wikipedia.org/wiki/Mobile_cloud_computing.
[2] Referred from internet resource: http://www.ediss.uni-goettingen.de
[3] Aceto, Luca, Andrea Morichetta, and Francesco Tiezzi. "Decision Support for Mobile Cloud Computing Applications via Model Checking", 2015 3rd IEEE International Conference on Mobile Cloud Computing Services and Engineering, 2015.
[4] "The mobile app economy is exploding." [Online]. Available: http://goo.gl/doZ6vr
[5] R. Alur and D. L. Dill, "A theory of timed automata," Theoretical Computer Science, vol. 126, no. 2, pp. 183– 235,1994.
[6] R. K. Balan, M. Satyanarayanan, S. Y. Park, and T. Okoshi, "Tactics based remote execution for mobile computing," in MobiSys. ACM, 2003, pp. 273–286.

[7] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, “To offload or not to offload? The bandwidth and energy costs of mobile cloud computing,” in INFOCOM, 2013. IEEE, 2013, pp. 1285–1293.
[8] G. Behrmann, A. David, and K. G. Larsen, “A tutorial on Uppaal,” inFormal Methods for the Design of Real-time Systems. Springer, 2004,pp. 200–236.
[9] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clone Cloud: Elastic execution between mobile device and cloud,” in EuroSys. ACM,2011, pp. 301– 314.
[10] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu,R. Chandra, and P. Bahl, “MAUI: Making smartphones last longer with code offload,” in MobiSys. ACM, 2010, pp. 49–62.
[11] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: architecture, applications, and approaches,” Wireless Comm. and Mobile Computing, vol. 13, no. 18, pp. 1587–1611, 2013.
[12] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” Future Generation Computer Systems, vol. 29, no. 1, pp. 84– 106, 2013.
[13] J. Flinn, “Cyber foraging: Bridging mobile and cloud computing,” Synthesis Lectures on Mobile and Pervasive Computing, vol. 7, no. 2,pp. 1–103, 2012.
[14] J. Flinn, S. Park, and M. Satyanarayanan, “Balancing performance, energy, and quality in pervasive computing,” in Distributed Computing Systems, 2002., 2002, pp. 217–226.
[15] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, “Cuckoo: A computation offloading framework for smartphones,” in Mobile Computing, Applications, and Services. Springer, 2012, pp. 59–79.
[16] K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” Computer, vol. 43, no. 4, pp.51–56, 2010.
[17] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clientsin cloud computing,” in USENIX, 2010, pp. 4– 4.
[18] D. Narayanan, J. Flinn, and M. Satyanarayanan, “Using history toimprove mobile application adaptation,” in Workshop on Mobile ComputingSystems and Applications, 2000, pp. 31–40.
[19] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan,“ Odessa: Enabling interactive perception applications on mobiledevices,” in MobiSys. ACM, 2011, pp. 43–56.
[20] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, “The remote processing framework for portable computer power saving,” in SAC.ACM, 1999, pp. 365– 372.
[21] Karthik Kumar • Jibang Liu • Yung-Hsiang Lu •Bharat Bhargava, “A Survey of Computation Offloading for Mobile Systems,” DOI 10.1007/s11036-012-0368-0