ISSN: 0975-766X
CODEN: IJPTFI
Research Article

Available Online through
www.ijptonline.com

A SURVEY ON APPLICATION PARTITIONING FOR MOBILE CLOUD COMPUTING

Thanapal. P*, Dr. Saleem Durai M A
School of Information Technology and Engineering, VIT University, Vellore, Tamilnadu, 632014. India.
School of Computer Science and Engineering, VIT University, Vellore, Tamilnadu 632014. India.
Email: thanapal.p@vit.ac.in

Received on 12-02-2016 *Accepted on 27-03-2016*

Abstract

Cloud computing is one of the incipient innovations which are discovering its utilization all over. Mobile cloud is one stage in front of cloud computing. We will give a brief an outline of cloud computing, mobile cloud computing and will discuss some of the application partition schemes for offloading in mobile cloud computing. In this paper, we give a broad review of application partitioning in mobile cloud computing research, while highlighting the particular suspicions in application partition. We introduce a taxonomy based on key issues in this area, and discuss about the distinctive methodologies taken to handle these issues. We conclude the paper with a critical analysis of challenges that have not yet been completely met, and highlight directions for future work.

Keywords: Cloud computing, mobile cloud computing, application partitioning, offloading, and dynamic partitioning.

I. Introduction

A. Cloud Computing

"Cloud computing alludes to both the applications conveyed as services over the Internet and the hardware and software in the data centers that give those services" [31]. A bunch of computer hardware and software that offer the services to the overall population (likely at a cost) makes up a 'public cloud'. Computing is thusly offered as a utility much like power, water, gas and so forth where you just pay per use.

For instance, Amazon's Elastic cloud, Microsoft's Azure stage, Google's App Engine and Sales force are some public cloud that are accessible today. In any case, cloud computing does exclude 'private cloud' which allude to data center inward to an association. Along these lines, cloud computing can be characterized as the conglomeration of registering as a utility and software as a service.

Virtualization of assets is a key prerequisite for a cloud provider for it is required by measurable multiplexing that is required for scalability of the cloud, furthermore to make the hallucination of vast assets to the cloud user. Ambrust et al. [30] holds the perspective that "diverse utility computing offerings will be recognized in view of the level of abstraction displayed to the software engineer and the level of management of the assets".

To take a sample from the current cloud providers, an example of Amazon's EC2 is all that much like a physical machine and gives the cloud client full control of the product stack with a slight API. This gives the user a great deal of adaptability in coding; in any case it additionally implies that Amazon has minimal automatic scalability and failover highlights. Conversely, Google's App Engine authorizes an API on the client however offers automatic scalability and failover choices. Microsoft's Azure stage is something in the middle of the previously stated suppliers by giving the client some decision in the dialect and offers to some degree programmed scaling and failover capacities. Each of the previously stated suppliers has distinctive alternatives for virtualizing computation, storage and communication.

Various cloud server-based application offloading systems have been proposed for outsourcing computational concentrated parts of the versatile applications to cloud data centers. Runtime computational offloading includes surrogate disclosure [1][2][3]. Asset estimation [4][5], and application partitioning [6].

B. Mobile cloud computing

There are a few existing meanings of mobile cloud computing, and diverse research implies distinctive ideas of the 'mobile cloud'.

1. Regularly, the term mobile cloud computing intends to run an application, for example, Google's Gmail for Mobile on a remote asset rich server (for this situation, Google servers), while the cell phone acts like a thin client associating over to the remote server through 3G. Some different illustrations of this sort are Facebook's area aware services, Twitter for mobile, mobile weather gadgets and so on.

2. Another methodology is to consider other cell phones themselves as well as asset suppliers of the cloud making up a mobile peer-to-peer as in [33]. Accordingly, the aggregate assets of the different cell phones in the local vicinity, and other stationary gadgets as well if accessible, will be used. This methodology underpins client versatility, and perceives the capability of portable mists to do aggregate detecting too. Peer-to-peer frameworks, for example, SATIN [35] for mobile self-organizing exist, yet these depend on segment model frameworks speaking to frameworks made up of interoperable local parts as opposed to offloading jobs to nearby portable assets.

3. The cloudlet idea proposed by Satyanarayanan [34] is another way to deal with mobile cloud computing. In this methodology where the cell phone offloads its workload to a nearby "cloudlet" included a few multi-center computers with availability to the remote cloud servers. Plug Computers can be viewed as great possibility for cloudlet servers on account of their structure component, differing qualities and low power utilization. They have the same general design as a typical computer, however are less capable, littler, and less costly, making them perfect for role small scale servers introduced in people infrastructure. These cloudlets would be arranged in like manner territories, for example, cafés so that cell phones can interface and capacity as a dainty customer to the cloudlet rather than a remote cloud server which would exhibit idleness and data transfer capacity issues.

The points of advantages and disadvantages of mobile cloud computing are [36]

Advantages of Mobile Cloud Computing

1. Reduces Battery Consumption: As a large portion of the processor intensive task is offloaded to cloud, it consequently can spare significant measure of vitality by sparing battery life.

2. Provides more data storage: Cloud can offer data storage and cell phones can utilize this storage room for putting away the information. Amazon Simple Storage Service (Amazon S3) [37] offers such sort of service by giving a database to utilize.

3. Enhances Reliability: The information can be put away and reinforcements are made on various servers, consequently guaranteeing that information is not inclined to misfortunes, regardless of the possibility that the cell phone is stolen.

4. Privacy: By scrambling the information put away on the cloud the security of the client is kept up.

Mobile cloud computing would likewise be based under the essential cloud computing ideas. As talked about by Mei et al. in [32] there are sure necessities that should be met in a cloud, for example, versatility, adaptability, accessibility and mindfulness. These are likewise legitimate necessities for mobile cloud computing.

Disadvantages of Mobile Cloud Computing

1. Low Bandwidth: Bandwidth and absence of sign or flag blunders is one of the principle reasons that Mobile cloud computing is not getting to be as valuable for the clients as it ought to be. Some other arrangements should be discovered to overcome data transmission constraints

2. Heterogeneous Networks: There are different kinds of innovation utilized as a part of cell phones like GSM, CDMA, WiMAX and taking care of such heterogeneous system is truly a troublesome assignment.

3. Interoperability between Platforms: Different cell phones use diverse Operating System for instance Phones based out of Android innovation like HTC use Android operating system. Symbian is utilized by Nokia, Apple Phones use IPhone IOS, subsequently making a portable application which can work in those is very much a troublesome assignment.

4. Security and Data Privacy: It is fundamental to spare information amid travel and also information when put away on the cloud servers. A few security assaults and dangers on remote system should be dealt with.

5. Data Access: It is vital to have a productive system of data access, which is financially savvy and spares the correspondence overhead.

6. Computational Offloading: Atrade off in the middle of communication and calculating costs must be assessed before choosing to offload any portable application and it's a perplexing procedure which relies on upon a few elements.

7. Pricing: The estimating model in the event of mobile cloud must be settled on the three partners i.e. application license provider, mobile network provider and cloud service supplier and there could be conflicts among three.

In programmed location plans for offloading frameworks, for example, [7][8], it is imperative to design partitioning and offloading plans for applications since offloading brings about system overhead from transmission delay and cost. Static partitioning, for example, [7] is not suitable if there are continuous transmission capacity changes in the portable correspondence.

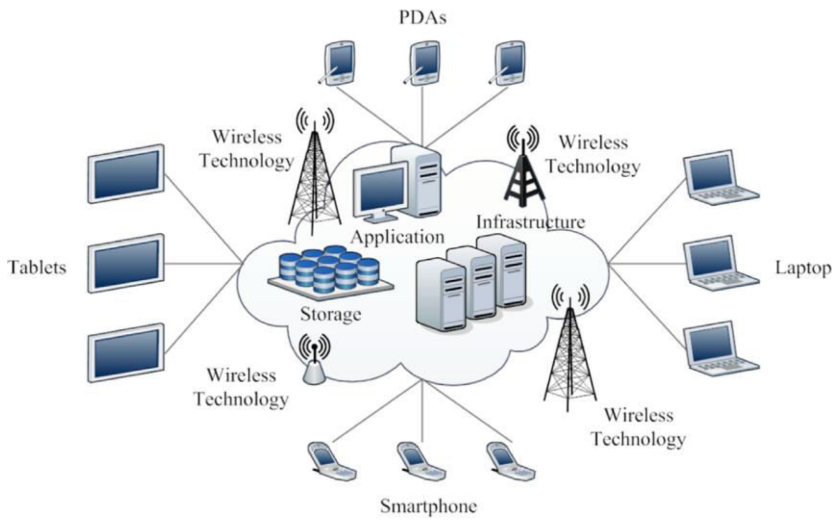

Fig. 1: Model of mobile cloud computing.

Figure 1 [29] demonstrates that MCC is empowered by three noteworthy parts - SMDs, (for example, cell phone, PDA, and tablet PC); advanced wireless technology, (for example, Wi-Fi, WiMAX, 3G and LTE); and computational cloud, (for example, Microsoft Azure and AWS)

One mainstream arrangement that has pulled in the consideration of specialists for over 10 years is to offload part of, or move the entire calculation to remote servers. This arrangement looks progressively engaging in light of the fact that loads of server assets are accessible and remote innovations, for example, LTE-A, WiMAX and rapid Wi-Fi (IEEE 801.11n) are turning out to be quick and omnipresent. Truth be told the late overall accessibility of vast scale cloud computing system from Amazon, Google, Microsoft, and Yahoo has prodded the development of an assortment of portable applications which influence assets in the cloud [9].

[4] Proposed a multilevel chart dividing heuristic that parcels a class diagram by progressively combining exceptionally coupled vertices to build progressively littler charts

In adaptive computation offloading, the position of parts to gadgets (segment topology) is commonly processed responsively because of a gadget coming up short on assets. The methodology consolidates the self-improvement model of autonomic figuring [11] with the engineering model [12][13] of programming adjustment. Adjustment choices enhance the non-practical conduct (execution, asset use and so on.) of programming and in that capacity don't expand or change useful necessities or goals.

More recently, [14] proposed a derived multi-level graph partitioning heuristic to adjust over different obliged gadgets by progressively coarsening an application chart. This was finished by haphazardly selecting vertices and blending them with their lightest (low vertex weight) yet exceptionally coupled (high edge weight) neighbour, until the quantity of vertices was identical to the quantity of teaming up gadgets. Every vertex in the resultant coarse chart is then mapped to a gadget in the coordinated effort and speaks to a partition of the first diagram.

II. Literature Survey

Early takes a shot at computation offloading [15][16][17]assumed unsurprising application execution in homogenous settings, and in this manner statically decided the position of parts or strings to gadgets. Likewise, different work on segment arrangement [18][19][20] statically decide the situation of use parts to gadgets. In any case, this is not appropriate to heterogeneous and dynamic situations in which asset accessibility and asset necessities are continually changing and consequently part topologies must be powerfully reconfigured to keep up application devotion in any given connection.

Spectra [21] and Chroma [22] are two illustrations of frameworks that utilization pre-installed services reachable by means of RPC to offload calculation. Applications use RPC to conjure usefulness in remote and local Spectra servers. At the point when the gadget needs to offload an application, the Spectra customer counsels a database that stores data about Spectra servers, for example, their current availability, CPU load and so on. These servers are pre-installed with application code going about as services. Developers need to physically segment the applications by indicating which techniques may be contender for offloading. Spectra chooses at runtime relying upon the asset pool, which of those previously stated techniques, if any will be offloaded and to which surrogate

The information access charges, issues with inactivity and data transmission [24], furthermore the levels of popularity of vitality when utilizing 3G network, the nearby cloud would be a superior distinct option for the remote cloud [23]. Moreover, utilizing the neighbourhood versatile assets is an effective method for making utilization of accessible calculation control, that would some way or another be sitting out of gear [25]. Following ordinarily cell phones are outfitted with detecting capacities, a cloud made up of cell phones will have the capacity to give the clients setting and area mindful services too, prompting a more customized experience.

Various arrangements are proposed for handover service in the remote systems. Handover happens when cell phone change system terminal. In [26] a handover management and decision algorithm is proposed for heterogeneous remote information systems. It introduces a hypothetical vertical handover plan for heterogeneous wireless network. Alnas and Awan contended that Mobile IP based handover management absence of pocket loss and latency during handover.

In [27] displayed the adaptive e-resource revelation and asset determination models in Grid. They examined three models: Adaptive pull model, Adaptive push model and Adaptive push-pull model. In adaptive pull model, Grid environment comprises of a few hubs, out of which one of the hub is made as coordinator where a solitary daemon is running. This daemon can force and gather dynamic condition of data, for example, CPU speed, CPU burdens, and memory size from different remote hubs. The offloading is an answer for enlarge these mobile systems' abilities by relocating calculation to more ingenious computers (i.e., servers). This is not the same as the conventional client-server engineering, where a thin-client dependably relocates computation to a server. Computation offloading is additionally unique in relation to the movement model utilized as a part of multiprocessor systems and Grid computing, where a procedure might be relocated for load balancing [28].

III. Application Partition Schemes

Application apportioning is the initial move towards offloading process. It can mostly be sorted as static and dynamic. Static partitioning manages dividing the application into allotments amid the outline time. Li Z. et al. [38] have characterized a static partitioning scheme by considering offloading at the level of strategy calls (assignments).

For a given system, they have profiled the data about computation time and information sharing, then built a cost graph and have proposed an assignment mapping algorithm to statically isolate the project into server undertakings and client tasks. In the wake of building the cost diagram, they have connected the branch-and-bound calculation to the cost chart to minimize the aggregate vitality utilization of calculation and the aggregate data communication cost.

Dynamic Partitioning considers the run time parameters which assume a critical part in deciding asset utilization in pervasive environment. Ou et al [39] have proposed a multi requirement (k+1) partitioning plan, to partition an application into k offloadable and one unoffloadable segments. They have made a cost graph by considering the measurements like communication cost, CPU expense and memory cost for each of the node of the graph .The run time parameters which are considered are acquired by taking rehashed screenshots of asset utilization by each of the node and any expansion or lessening in asset utilization is caught or mulled over. Another element plan is adjusted in Maui [40], here the programmer first at first checks beginning partition of the application utilizing explanations. At that point three sorts of profilers i.e. system profilers, gadget profilers, program profilers are utilized to assess the application execution and vitality utilization and after that a Maui solver chooses which technique to offload. The Problem of manual dividing is that it is an extra overhead on programmer and can't adjust to changes in nature. Subsequently automatic dividing Techniques like Spectra [42] have appeared. As of late a few foundations like Clonecloud [41] have risen which relocate the whole OS and its individual applications to cloud by making a clone? A "Clone" is a mirror picture of a Smartphone running on a virtual machine. By appear differently in relation to advanced cells, such a "Clone" has more equipment, programming, system, vitality assets in a virtual machine which gives more suitable environment to prepare convoluted undertakings. The flowchart [29] of application partitioning and computational offloading component for MCC is appeared in Fig 2.

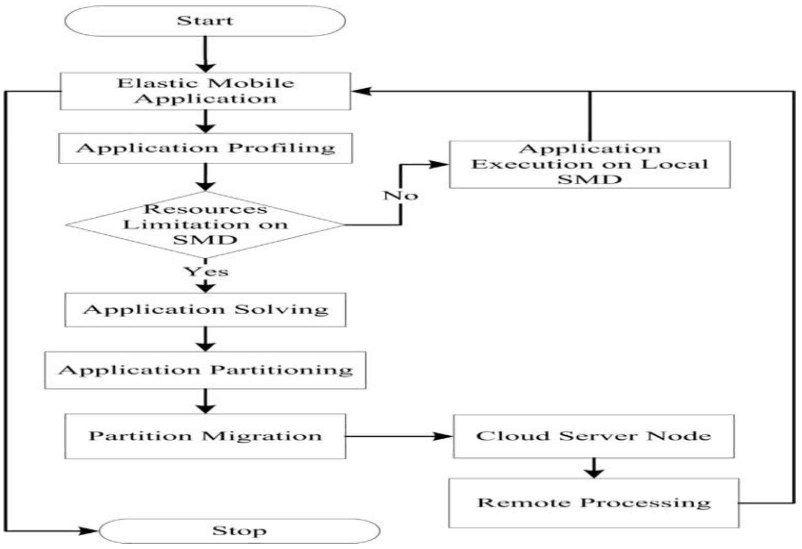

Fig. 2: Flowchart of application partitioning and computational offloading mechanism for MCC.

I. Review of application partition based on the application partitioning models

This area presents are perspective of the current application partition schemes and talks about their suggestions and other imperative viewpoints taking into account the diverse application apportioning models [29, 51] introduced in the scientific categorization

A. Object-level Granularity Decision Making

Object-level [43, 44, 45] granularity for adjustment choice computation offers the most adaptability in the resultant article topology on the grounds that the area of every individual item is chosen freely by figuring the utility of its situation on each of the powerfully pre-chosen applicant nodes amid an adjustment.

The most point by point case of this methodology is that of Ryan and Rossi [43] in which a score for every object-to-node coordinating is figured in view of a fancied goal, for example, customer burden alleviation. The asset necessity of each article and the asset accessibility of every applicant hub are utilized as a part of processing this score.

B. Class-level Granularity Decision Making

Keeping in mind the end goal to figure coupling data while keeping up computational achievability, class-level [46, 47] choice making approaches have been proposed. In such methodologies, a class is the unit of relocation and serves as a reflection of the asset prerequisites of all its case objects. Thus movement of a class results in the aggregate relocation of all objects of that class. Basic to the work in class-level adjustment is the representation of an application as an element weighted undirected class diagram. The diagram comprises of vertices which speak to classes, and edges which portray the coupling designs amongst them. Both vertices and edges are weighted, with various works utilizing diverse plans to speak to the heaviness of vertices. For instance, Gu et al.[44, 47] speak to vertex weight as the memory usage of a class while Shumao et al. [46] propose the utilization of a composite weight plan which calculates the system, memory and processor use of a class into a bound together vertex weight. Then again, edge weight is normally spoken to as the aggregate conjuring number between all strategies for the two classes, which just roughly approximates the genuine expense of an edge, and in this way a more precise methodology

C. Graph-based application partitioning algorithms

By and large, the components of diagram vertices and edges are utilized to speak to the parameter or setting of an application. These incorporate the accessible assets, information size, communication overhead, computation cost, and memory cost. Application partition schemes receive the chart model for displaying execution states, cost models, interior reliance, data flow, and control flow. A mobile application is displayed at various granularity levels, which incorporate better granularity adjustment those outcomes in bigger number of vertices with very mind boggling communication designs when contrasted with class charts. Partitioning an application chart into various disjoint segments is an imperative stride for distinguishing the rundown of classes that will be offloaded to remote servers. A fitting chart model can enhance the proficiency of partitioning choice respect less of whether fine-grained or coarse-grained granularity is connected. Fundamentally, every partition satisfies the condition that its weight is not exactly or equivalent to the asset accessibility of the gadget. Getting the ideal dividing choice in diagram based application partition schemes is a Non-deterministic Polynomial-Complete (NPC) issue [48].

D. Linear programming-based application partitioning algorithms

Linear programming (LP) is a mathematical technique for discourage mining endlessly to accomplish the best result, for example, most extreme benefit or least cost with a rundown of prerequisites which are spoken to as a straight comparison in a scientific model. Formally, LP is characterized as a procedure for the improvement of a straight target application partition scheme city, subject to direct uniformity and direct disparity requirements, and it is helpful for tackling most pessimistic scenario issues [49]. The benefit of executing LP is that it generally creates ideal results for a specific target capacity. Tackling LP issues, in any case, frequently requires a ton of calculation time [50].

E. Hybrid application partitioning algorithms

The hybrid application partition scheme consolidates the joined elements from both the graph based and LP-based application apportioning strategies. They tend to separate the imperative components of graph based application partition schemes and LP-based application partition schemes so as to enhance the execution of application partition schemes. Mobile Assistance Using Infrastructure (MAUI) speaks to the application as a call chart for illuminating diagram enhancement utilizing 0-1 LP for application apportioning [24]. It gives fine-grained code offloading to enhance vitality use with negligible weight to the developer. MAUI empowers programmer to deliver a starting partitioning of utilization through explaining the neighbourhood and remote parts of the application. An imperative component of MAUI is that it uses oversaw codes to lessen the weight on the programmer in managing program dividing, and it likewise amplifies vitality sparing amid calculation offloading.

II. Issues and challenges for partitioning of applications in MCC

In disseminated application handling, elastic application is partitioned and the concentrated parts of the application are offloaded to cloud server hubs [24, 52, 41, 53, and 54]. Subsequently, a disseminated application preparing stage is set up and the application processing platform of cell phones is outsourced to the cloud server. A few application partition schemes at present actualize a component for disseminated handling of concentrated portable applications in which the execution on the versatile application is blocked unless the outcomes are come back from the remote server hub [24, 52, and 41]. Subsequently, it is a test to utilize an ideal synchronization instrument for guaranteeing consistency in appropriated allotments of the portable applications.

The granularity of utilization apportioning for dispersed application preparing is essential for an ideal dividing algorithm design. The granularity level of the application partitioning can influence the complexity of distributed application processing and the assets use overheads in runtime component migration [55]. Thus, it is trying to actualize ideal granularity level for application partitioning which includes negligible correspondence overhead for component offloading and light weight system for the foundation and management of distributed stages.

Current dividing calculations are intended to offload concentrated parts of a versatile application from the cell phone to the cloud, there by drawing out battery life and improving execution. Distributed application handling systems use distinctive target capacities for partitioning of flexible applications. Such structures concentrate on which segment of the application to segment, and when to offload application partitions for remote execution [52].

Consequently, high programming quality is required for tending to the previously stated issue, which can affect the streamlining of the dividing computation, and rich client involvement in the distributed deployment of elastic application for MCC.

Current programming designing practices for application advancement require least endeavours from application developers. This has contributed enormously to the fast advancement of programming, and lessened the developmental time and cost. It is trying to minimize the formative endeavours for the arrangement of the components of the application as local or remote.

The issue is considerably additionally trying for better granularity level dividing when contrasted with coarse granularity level apportioning. Calculation offloading is executed as a noteworthy application layer answer for empowering computationally intensive mobile applications on smart mobile devices. Utilizing the offloading procedure, the principle issue experienced dis in apportioning the computational heap of the intensive mobile application between the cell phone and the cloud [56]. It is trying to plan calculations with various target capacities for application partition and part offloading, for example, disconnected from the net ease of use, rich client encounters, battery influence protection and consistent application execution.

III. Discussion on Gap analysis.

This paper recognizes application partitioning as a free part of the element computational offloading method for MCC. It explored the best in class application partition schemes in disseminated application preparing to distinguish the primary issues and difficulties. Current application dividing methods, classified in view of topical scientific classification, and other critical perspectives are assessed. We looked at the likenesses and contrasts among the current application partition schemes in view of applicable parameters. A few calculations are limited to the utilization of a single programming language, though others use multi-programming language support. A few algorithms use automatic annotation for characterizing the extent of the parts of the mobile application, while despite everything others need manual explanation by application developers. Application partition schemes have been created to accomplish various targets, for example, diminishing network overhead, saving energy, improving performance, decreasing memory limitations, lessening software engineers' weight, dynamically updating application, and multi-site offloading.

IV. Conclusion

This paper gives a comprehensive study, portrayed, and sorted a several aspects of application partition techniques in mobile cloud computing. We have developed taxonomies for mobile cloud computing to characterize the common partitioning techniques. Moreover, we are additionally ready to find the inadequacies and identify gaps in the current application partition schemes through the characterization. These show that the future direction can be taken by researchers in this area. Thus, this paper not only provides a comprehensive classification framework and serves as a tool for comprehension application partition schemes, additionally serves as a reference for future works.

References

1. Balan R, Flinn J, Satyanarayanan M, Sinnamohideen S, Yang HI. The case for cyber foraging. In: Proceedings of the 10th workshop on ACM SIGOPS Europeanworkshop (EW'02), Saint-Emilion, France. ACM; 2002. p. 87–92.
2. Ou S, Yang K, Hu L. Cross: a combined routing and surrogate selection algorithm for pervasive service offloading in mobile ad hoc environments. In: Proceedings of the IEEE global telecommunications conference (GLOBECOM'07), Washington, DC, USA. IEEE; 2007. p. 720–725
3. Navimipour NJ, Rahmani AM, Navin AH, Hosseinzadeh M. Resource discovery mechanisms in grid systems: a survey. J Netw Comput Appl 2014;41:389–410.

4. Levine DA, Akyildiz IF, Naghshineh M. A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept. IEEE/ACM Trans Netw 1997;5:1–12.
5. Reddy KS, Ranjan M. Solar resource estimation using artificial neural networks and comparison with other correlation models. Energy Convers Manag2003; 44:2519–30.
6. Tilevich E, Smaragdakis Y. J-orchestra: automatic java application partitioning. In:Proceedings of the 16th European conference on object-oriented programming(ECOOP'02), Malaga, Spain. Springer-Verlag; 2006. p. 178–204.
7. Diaconescu RE, Wang L, Mouri Z, Chu, M. A compiler and runtime infrastructure for automatic program distribution. In: Proceedings of the 19th IEEE international parallel and distributed processing symposium (IPDPS'05); 2005
8. Chun B-G, Ihm S, Maniatis P, Naik M, Patti A. Clone Cloud: elastic execution between mobile device and cloud. In: Proceedings of the sixth conference on computer systems (EuroSys '11); 2011.
9. “Apple iphone app store,” http://www.apple.com/iphone/apps-for-iphone/.
10. S. Ou, K. Yang, and A. Liotta, "An adaptive multi-constraint partitioning algorithm for offloading in pervasive systems," in Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on, 2006, pp. 10 pp.-125.
11. Kephart, J.O., Chess, D.M., 2003. The vision of autonomic computing. Computer 36,41–50.
12. Oreizy, P., Gorlick, M.M., Taylor, R.N., Heimhigner, D., Johnson, G., Medvidovic, N.,Quilici, A., Rosenblum, D.S., Wolf, A.L., 1999. An architecture-based approach to self-adaptive software. Intelligent Systems and their Applications, IEEE 14,54–62.
13. Ou, S., Yang, K., Liotta, A., 2006. An adaptive multi-constraint partitioning algorithm for offloading in pervasive systems. In: Fourth Annual IEEE International Conference on Pervasive Computing and Communications, 2006. PerCom 2006, pp.10–125.
14. Philippsen, M., Zenger, M., 1997. Java Party transparent remote objects in Java. In: Proc. ACM 1997 PPoPP Workshop on Java for Science and Engineering Computation.
15. Sakamoto, K., Yoshida, M., 2007. Design and evaluation of large scale loosely coupled cluster-based distributed systems. In: 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), pp. 572–577.

16. Tilevich, E., Smaragdakis, Y.,2002. J-orchestra: automatic Java application partitioning. In: European Conference on Object-Oriented Programming 2002 (ECOOP 2002). Springer-Verlag, pp. 178–204.
17. Kramer, J., Magee, J., 2007. Self-managed systems: an architectural challenge. IEEE,259–268.
18. Bastarrica, M.C., Caballero, R.E., Demurjian, S.A., Shvartsman, A.A., 2001. Two optimization techniques for component-based systems deployment. Integration 2,3.
19. Medvidovic, N., Malek, S., 2007. Software deployment architecture and quality-ofservicein pervasive environments. ACM, 47–51.
20. Meedeniya, I., Buhnova, B., Aleti, A., Grunske, L., 2011. Reliability-driven deployment optimization for embedded systems. Journal of Systems and Software.
21. J. Flinn, S. Park, M. Satyanarayanan, Balancing performance, energy, and qualityin pervasive computing, in: Proceedings of the 22nd International Conference on Distributed Computing Systems, 2002, IEEE, 2002, pp. 217–226.
22. R. Balan, M. Satyanarayanan, S. Park, T. Okoshi, Tactics-based remote execution for mobile computing, in: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, ACM, 2003,pp. 273–286.
23. M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Pervasive Computing 8 (2009) 14–23.
24. E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R.Chandra, P. Bahl, Maui: making smartphones last longer with code offload,in: Proceedings of the 8th International Conference on Mobile Systems,Applications, and Services, MobiSys'10, ACM, New York, NY, USA, 2010,pp. 49–62.
25. A. Coronato, G.D. Pietro, Mipeg: a middleware infrastructure for pervasivegrids, Future Generation Computer Systems 24 (2008) 17–29.
26. Le D, Fu X, Hogrefe D. A review of mobility support paradigms for the internet. CommunSurv Tutor IEEE 2006;8:38–51.
27. Kovvur RMR, Kadappa V, Ramachandram S, Govardhan A. Adaptive resource discovery models and resource selection in grids. In: Paper presented at the 2010 1st international conference on parallel distributed and grid computing (PDGC); 28–30 October2010.
28. Powell ML, Miller BP (1983) Process migration in demos/mp.ACM SIGOPS OperSyst Rev 17(5):110–119

29. Liu, Jieyao, et al. "Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions." *Journal of Network and Computer Applications* 48 (2015): 99-117.
30. W. Vogels, A heads in the clouds the power of infrastructure as a service, in: Proceedings of the 1st Workshop on Cloud Computing and Applications, CCA'08.
31. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, Above the clouds: a Berkeley view of cloud computing, Technical Report UCB/EECS-2009-28, 2009
32. L. Mei, W. Chan, T. Tse, A tale of clouds: paradigm comparisons and some thoughts on research issues, in: Proceedings of the Asia-Pacific Services Computing Conference, APSCC'08, IEEE, 2008, pp. 464–469
33. E.E. Marinelli, Hyrax: cloud computing on mobile devices using MapReduce, Master's Thesis, Carnegie Mellon University, 2009.
34. M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Pervasive Computing 8 (2009) 14–23
35. S. Zachariadis, C. Mascolo, W. Emmerich, Satin: a component model for mobile self organisation, in: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, in: Lecture Notes in Computer Science, vol. 3291, Springer, Berlin, Heidelberg, 2004, pp. 1303–1321. http://dx.doi.org/10.1007/978-3-540-30469-2_31.
36. Dudeja, Monika, and KritikaSoni. "Offloading Schemes in Mobile Cloud."*International Journal of Computer Applications* 96.8 (2014).
37. Amazon simple storage service [Online]. Accessed January 5 2016 Available: http://aws.amazon.com/s3/
38. Z. Li, C. Wang, and R. Xu, "Computation Offloading to Save Energy on Handheld Devices: A Partition Scheme, "Proc. Int'l Conf. Compilers, Architectures and Synthesis for Embedded Systems, Nov. 2001.
39. Ou S, Yang K, Liotta A (2006) "An adaptive multi constraint partitioning algorithm for offloading in pervasive systems." In:IEEE international conference on pervasive computing and communications, pp 116–125
40. E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smart phones last longer with code offload," in Proc. 8th international conference on Mobile systems, applications, and services. ACM, 2010, pp. 49–62.
41. B.-G. Chun, S. Ihm, P. Maniatis, and M. Naik, "Clone cloud: boosting mobile device applications through cloud clone execution," arXiv preprint arXiv: 1009.3088, 2010.

42. J. Flinn, D. Narayanan, and M. Satyanarayanan, “Selftuned remote execution for pervasive computing,” in Hot Topics in Operating Systems,2001. Proceedings of the Eighth Workshop on. IEEE, 2001, pp. 61–66.
43. Rossi P. and Ryan C., "An Empirical Evaluation of Dynamic Local Adaptation for Distributed Mobile Applications," presented at the Proc.of 2005 International Symposium on Distributed Objects and Applications (DOA 2005), Larnaca, Cyprus,, 2005.
44. X. Gu, K. Nahrstedt, A. Messer, I. Greenberg, and D. Milojicic, "Adaptive offloading inference for delivering applications in pervasivecomputing environments," presented at the Pervasive Computing and Communications, 2003. PerCom 2003. First IEEE International Conference on 2003.
45. Ryan C. and Westhorpe C., "Application Adaptation through Transparent and Portable Object Mobility in Java," presented at the Proc. of 2004 International Symposium on Distributed Objects and Applications (DOA 2004), Larnaca, Cyprus, 2004.
46. S. Ou, K. Yang, and A. Liotta, "An adaptive multi-constraint partitioning algorithm for offloading in pervasive systems," in *Pervasive Computing and Communications, 2006. Per Com 2006. Fourth Annual IEEE International Conference on*, 2006, pp. 10 pp.-125.
47. XiaohuiGu, Alan Messer, Ira Greenberg, Dejan Milojicic, and Klara Nahrstedt, "Adaptive Offloading for Pervasive Computing," *IEEE Pervasive Computing,* vol. 3, pp. 66-73, 2004.
48. Garey MR, Johnson DS. Computers and intractability: a guide to the theory of np- completeness. W.H. Freeman & Co.
49. Gass SI. Linear programming: methods and applications. Courier Dover Publications.
50. Abebe, Ermyas, and Caspar Ryan. "A hybrid granularity graph for improving adaptive application partitioning efficacy in mobile computing environments."*Network Computing and Applications (NCA), 2011 10th IEEE International Symposium on*. IEEE, 2011.
51. Giurgiu I, Riva O, Juric D, Krivulev I, Alonso G. Calling the cloud: enabling mobile phones as interfaces to cloud applications. In: Proceedings of the ACM / IFIP / USENIX 10th international conference on middleware (Middleware'09), Urbana, IL, USA. Springer-Verlag; 2009. p.83–102.
52. Ou S, Yang K, Liotta A. An adaptive multi-constraint partitioning algorithm for offloading in pervasive systems. In: Proceedings of the fourth annual IEEE international conference on pervasive computing and communications (Per- Com'06), Pisa, Italy. IEEE; 2006. p.116–25.

53. Tilevich E,Smaragdakis Y. J- orchestra : automatic java application partitioning. In: Proceedings of the 16th European conference on object-oriented programming (ECOOP'02), Malaga, Spain. Springer-Verlag; 2006.p.178–204.

54. Niemann R, Marwedel P. An algorithm for hardware / software partitioning using mixed integer linear programming. Des Autom Embed Syst1997;2:165–93.

55. Shiraz M, Ahmed E, Gani A, Han Q. Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing. J Super compute 2014; 67:84–103

56. Kumar K, Liu J, Lu Y H, Bhargava B. A survey of computation offloading for mobile systems. Mob NetwAppl 2012; 18:129–40.

Corresponding Author:

Thanapal. P*,

Email: *thanapal.p@vit.ac.in*