World Applied Sciences Journal 29 (Data Mining and Soft Computing Techniques): 41-47, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.dmsct.8

Mobile Recommendation Engine for Offloading Computations to Cloud Using Hadoop Cluster

Uma Nandhini and Latha Tamilselvan

Department of Information Technology, School of Computer,
Information and Mathematical Sciences, B.S.Abdur Rahman University,Chennai-600048, India

Abstract: The growth of mobile applications due to the increasing popularity of smartphones and ubiquity of wireless access has fueled cloud computing to converge into mobile cloud. The intrinsic storage, processing and battery power constraints of mobile devices can be solved by adopting unlimited storage and computing power offered by cloud servers and turn the issues into a favorable opportunity for mobile cloud computing. We propose to build a mobile recommendation engine that collects context and client sensitive information's from smart phone users for effectivelymaking a decision to offloadapplications that are compute intensive into cloud through nearby resource rich server. The information so received by the cloudis clustered and partitioned using Hadoop MapReduce framework. The provisioned Hadoop cluster extracts the information's by mining similarity co-occurrences. The resultant output is used to build collaborative filtering based recommendation model where possible offloading scenarios are listed against each application based on the client awareness. This mobile recommender model provides the advantage of federated and automated movement with very little human interaction for making offloading decision.Earlier models use offloading into cloud only for backing up the data to cloud storages like SkyDrive or Google Drive. Hence computation offloading with recommended service through mining clients information is a novel idea for the near future assuming the growth of smart phones and social networking.

Key words: Mobile cloud · Data offloading · Mapreduce framework · Cloudlet · Prediction and recommendation

INTRODUCTION

As the technology progresses the need for a scalable, manageable and accessible computing infrastructure also grows. Traditional data centers that have earlier hosted various services are changing fast to a new paradigm called 'cloud' that provides services on demand based on the users specific needs with reduced cost and flexibility. It encapsulates underlying technology like SOA (Service Oriented Architecture) and Virtualization from the user's perspective and provides apparently user friendly automated service over the internet. Already much traditional install-and-use software have migrated to cloud to have better product viability avoiding maintenance through updates and releasing service packs. Some of the well-known developments for software packages include Microsoft Office 365 and Adobe Creative Cloud, in-terms of computational aspects Amazon Web Services (AWS) and Google App Engine (GAE) have really made their presence in Cloud Computing market. Nevertheless the successes of these suites are yet to be achieved because of its adoption that needed more market penetration. The important factor that hinders the growth of these applications is how good they arecatered to the needs of smartphone user, because going by the trends the number of smartphones exceeds computer [1].

Around half the mobile users are using smartphones for various purposes apart from just making basic communications. So cloud computing needs to penetrate this ever-increasing mobile phone market and make better applications to be hosted in cloud that in a way should be used by smartphone users. In order to achieve the goal of cloud in mobile, multiple hardware vendors developed

Corresponding Author: Uma Nandhini, Department of Information Technology, School of Computer, Information and Mathematical Sciences, B.S. Abdur Rahman University, Chennai-600048, Tamil Nadu, India. Cell: +9841593116.

processors with multicore and small farm factors, similarly many software developments for smartphones like Android and iOS offered cloud based services. These future software applications connect to cloud servers to reap the benefits of computing power, storage space and other virtual service through online, has laid the foundation for next generation computing paradigm called as Mobile Cloud Computing (MCC) [2][3].However, again the success of MCC lies in how efficiently cloud applications like, image editing, voice recognition, augmented reality, audio video processing and other rich digital contents can be processed with smartphones. Because traditionally these types of applications require at least a desktop or laptop computer for processing, but in comparison a mobile phone can hardly match its features with it, apart from that its battery efficiency which bother the continuous usage of mobile phones.

Hence these resource poor mobile phones need a resource rich middleware that could exchange the service via high speed wireless connectivity like WiFi or WiMax. When these applications are executed within the mobile, than it will take much time may be in hours together leading to processor and battery heating up and in turn poor efficiently of resources [1]. Also meanwhile when these applications utilize full computation power and memory of the already resource poor devices, then other task and application may get delayed to start or even may not start. Hence an offloading server acts as a surrogate system for smart devices to compute and process the task for general applications that require immediate response. For other applications it may forward to registered cloud services via dedicated internet, once the task is accomplished the results can be stored in any of the cloud drives.

Offloading computation to nearby server requires identifying the applications parameters and establishing a clear picture of where and when to offload it for a feasible solution. Some of the decisions that need to be taken as soon as the application is launched are…

- Whether the mobile processor can able to process the application
- If it can, then how much time will it take to complete the task
- Power utilized if the computation is done within
- Calculate time taken for the application to compute at the server
- Calculate time taken to synchronize with server
- Time taken to transfer and receive the code for the given network bandwidth
- Compare and take a decision whether to offload or compute within

Once the application is offloaded as per the decision made at the user's end, then for better service the surrogate system must..

- Identify its clients and analyze the attributes of the clients
- Identify the application and its attributes,
- Identify context and behavior
- Security capabilities of client devices
- Location based service offloading

With these decisions in process while making an offloading task, there are quite a large number of data sets that are transferred between user and server. Now, consider a scenario in an shopping mall where there is an offloading server for smartphone users, then daily there would be thousands of customers visiting the place and almost everyone would use their devices for high computation application like video voice over, picture editing, games and business intelligence etc. The server will receive request from hundreds of users to do processing and storage due to lack of resource in their mobile phones. So, in order to execute the task the server have to collect and analyze large sets of big data which in turn can get accumulated to terabytes and may even require a separate storage facility. The analysis of these data sets can help judge the efficiency of offloading process for a particular application of the client device capabilities. The analyzed results may help users to decide on future associations with the service or may wish to get advertisement of the new applications and associated offloading services. Thus to do all these a recommendation engine must support to do big data analytics of the user features.

The objective is to propose a recommendation framework in mobile cloud environment that identifies client, context and security attributes that will dynamically adjust the process of offloading. Further a collaborative filtering enables the client devices to identify the applications that could be offloaded to cloud. This has to be achieved keeping in mind the client's location and without disturbing or introducing latency to the users interface or minimizing the applications functionality.

Literature Survey: The increasing popularity of mobile devices calls for effective execution of mobile applications. Many researches towards outsourcing

computing intensive tasks to external public or private resources have being undertaken keeping in view of insufficient computing resources on mobile devices. In[3] the author compares the performance of offloading computation based on the communication time that it takes to do the process, also compares various applications and come out that not all applications are energy efficient when migrated to cloud servers. The pioneering paper from Sathyanarayana et.al [4] proposed a cloudlet model that is used when applications are transferred using a one hop communication path in wireless media to multiple VMs that can be quickly instantiated and customized based on the service requested. In his vision of cloudlet, the advantages of avoiding long latency in accessing cloud services and having virtual machines for offloading are important. In the concept of Clone Cloud [5], an emulated service of mobile applications is created on the remote machine whenever the mobile devices run low on hardware resources. A WiFi based offloading solution [6] for Metropolitan Advance Delivery Network (MADNet) is proposed for improving the energy efficiency using energy aware offloading decision algorithms are employed. A prototype implementation using Nokia N900 smartphone to evaluate its performance when data is offloaded in mobility is presented. Thus many researches have tried to create a mobile cloud environment. However, little attention has been paid to the overall users' response time, where the network bandwidth may dominate.In our proposed work, we set to investigate how to effectively minimize users' response time for mobile applications through collaborative filtering by mining the available data collected through mobile social networks. Further the recommendation engine analyses computationsthat are outsourced to nearby computers and send back to the mobile device based on the jobs priority and compute intensity.

Model of Mobile Recommendation Engine: In order to help users to find and evaluate any set of items of interest like music, books, applications or any nearby services that have earlier been used by others, than recommendation system is the best method to find it. A recommendation engine is a feature of data mining technique that filters items by predicting how a user might rate them. It solves the problem of connecting your existing users with the right items in your massive inventory (generally terms as big data) of products or content. There are two basic approaches to building a recommendation system: the collaborative filtering method and the content-based approach [8]. Collaborative filtering algorithms take user ratings or other user behavior and make recommendations based on what users with similar behavior liked or purchased. The content-based approach requires deep knowledge of your massive inventory of products. Each item must be profiled based on its characteristics. In our proposed architecture we recommend the a suitable application that could be offloaded to nearby cloudlet. Hence the following section provides an idea of how and when computations can be offloaded and benefits of offloading.

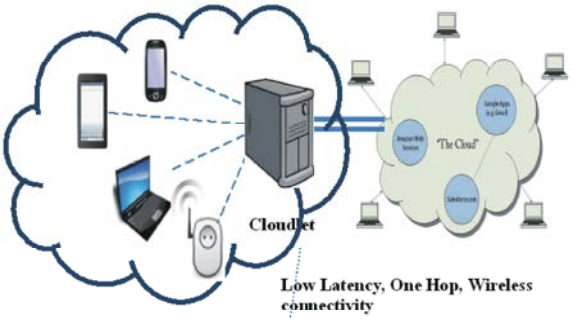

Fig. 1: A representation of Cloudlet

Computational Offloading: The need for offloading the data to the nearby cloudlet, which are primarily due to overloading of 3G cellular traffic, heavy monetary cost for 3G/4G cellular subscription. Further the advantage of using the WiFi communication as the medium of data transfer instead of relying on 2G/3G network is that the overall energy consumption by adopting WiFi is just $1/6^{th}$ and the popularity of WiFi hotspots.

Our proposed model provides a low latency because of the usage of nearby cloudlet which may be any system that are registered with our networks to provide the cloudlet service. It is a one-hop wireless connectivity; thereby reducing the task of inter-network transfer of data as well there is no major routing policies that need to be adopted. As shown in the Figure 1 our cloudlet is under the bigger cloud depicting the proposed activity that's going to happen with respect to mobile cloud computing.

To explain the model with a scenario, some computation intensive applications cannot be run on the smartphones since their computing power and battery life are still limited for such resource demanding applications as video encoding/decoding, image detection/recognition, 3D graphics rendering and so on. For example, a face detection application on iPhone 3GS takes 11 seconds to detect a face from a given picture and people might feel it is too slow. To mitigate this delay there are two categories

of computation-intensive applications. Applications in the first category are single-purpose and moderately slow (more than 10 seconds) to process on a single smartphone, but can be processed reasonably quickly when offloading the module to a fast computer. A face detection application is an example application in this category. Applications in the second category are extremely slow (a few minutes to hours) to process on a smartphone, but their execution time can be dramatically reduced by offloading computationally heavy modules to multiple computers. A face recognition application, which searches against a database of images to find similar faces, belongs to this category. Another is in an agriculture environment a pest detection through video processing of crop is also requires computation [7].Further, client awareness to seamlessly adapt to each end-users device regardless of the types of client platform, or the capabilities of the client devices based on its performance, we try to optimize the application delivery and end-user experience in a secure mode by providing certain key verifications without complicating the process of user friendliness.

Next the client attributes so collected from thousands of smartphone users are clustered. This information's are classified based on similarity and concurrency and applied to recommendation engine for a collaboration algorithm as shown in Figure. The algorithm collects and analyses large amount of information on user's behavior, activities and preferences and predicts what users will like based on the similarity to other users. A k-nearest neighbor algorithm can be used for training the object for classifying the closest pair. Once the classification is over, the prediction can be adjudged based on the recommendation results. This results are feed to the smartphone users for selecting suitable applications and as well decide whether this computations that are generated can be offloaded or not.

Map Reduce Process: As mentioned above in a shopping mall example the usage of our cloud middleware in the mall leads to big data analytics for evaluating similarity between different sets of users [10]. The evaluation criteria should be based on what the user has accessed in the past during its previous visits and if he has accessed the cloud middleware for any offloading purpose than all these attributes must be stored as in Table 1.

These information about the attributes of the user needs to be analyzed for similarity cooccurrence. For example, if the any user has previously offloaded a movie to the cloudlet and now another user wishes to view the same movie then, the similarity output from map-reduce will provide a solution. The framework will process earlier attributes and based on the access patterns it will segregate the users based on cooccurrence.

Recommendation System: Any recommendation system needs data to be preprocessed, analysed and interpreted inorder to efficiently predict the behavior and recommend suitably according to the user needs. Therefore it collects and analyses explicit data like.

- Analyzing Client's attributes for recommending offloading process
- Observing the applications that user downloads from online app store
- Frequently used applications to store in cloudlet server
- Mining social networking features for advertising new services
- Obtaining a list of applications that user has listened or watched
- Analyze the users social networks and judging the preferred list
- Feedback information on items that has been used recently
- Collect rank and trust ratings from users observation

Our approach classifies this information using collaborative filtering which is best for any recommendation system. The goal of Collaborative

Table 1: Client Attributes

Device Attributes	Application Attributes	Context Attributes
Mobile Device USSID	Type of App. Offline or Online	Battery remaining
MAC address	Size of Application	Type of Service used
Network Capability	Memory Utilized when running	Social Situation
Security Certificates	No of Threads per Application	Time Context
Privacy Constraints	No of times offloaded to Cloudlet	Geographical Location
Hardware Configurations	Other applications used through WiFi	Mobility pattern

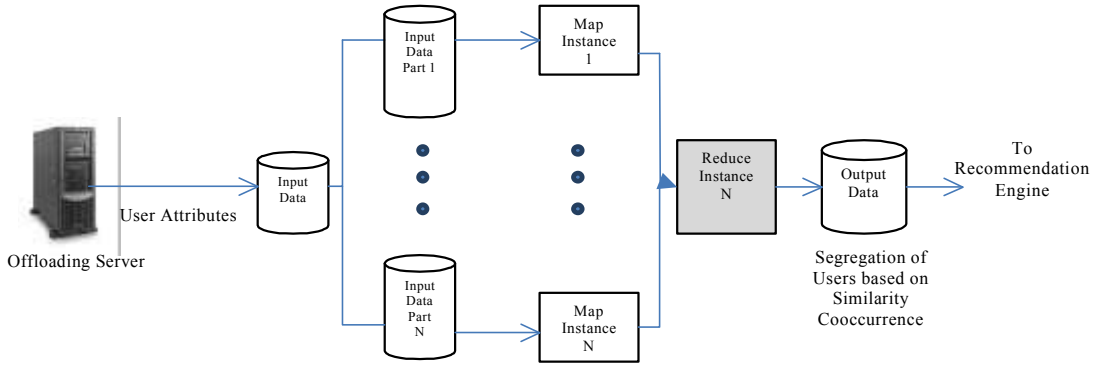

Fig. 2: Map Reduce of Big Data for Similarity

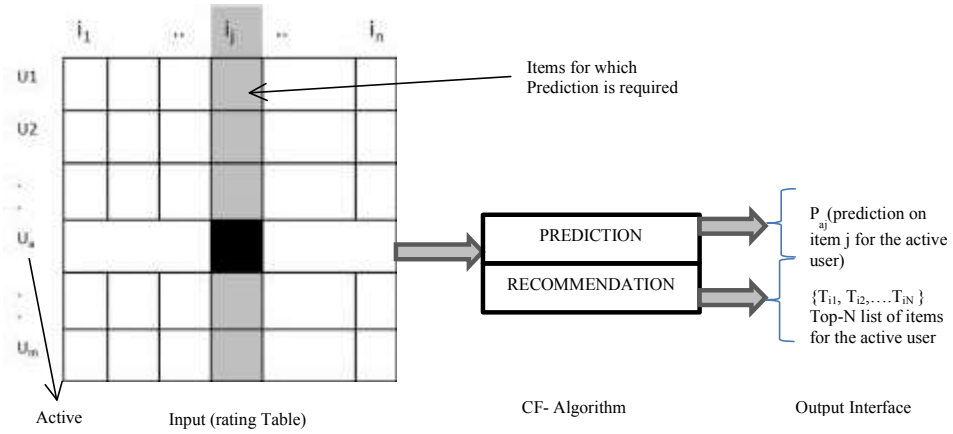

Fig. 3: Collaborative Filtering Process

Filtering (CF) algorithm is to suggest new services or items or to predict the usage of a given service for a particular user based on the previous interest and feedbacks of likeminded users [9].

Let us assume that a distinguished user u_a U called the recent active user for whom the task of CF algorithm is to find a service likeliness that can be either to predict or recommend. Consider a scenario where there are a list of m users $U= \{u_1, u_2... u_m\}$ and a list of n items$I = \{i_1, i_2,..,i_n\}$. Each user has a list of services I_{ui}, for which the users have expressed their opinion or given feedback.

Opinions or feedbacks can be given either explicitly as a rating score in a numerical range, or can be implicitly derived from how the user have behaved to the services that has been used. For example in our case, the implicit values may be how good the offloading has completed, time spent with the server and other behavior.

- Prediction is a numerical value $P_{a,j}$, expressing the predicted likeliness of item i_j $I_{ua}$for active user u_a.
- Recommendation is a list of N items, I_rI, that the active user will like the most.

The best algorithm for filtering the given data set is the memory based collaborative filtering algorithm which uses entire data sets to predict. It uses statistical techniques to find the set of mobile users; known as neighbors. Hence we have considered kNN (k- Nearest Neighbor) classifier to measure the similarities [11]. This is a nearest neighbor classifier that classifies (Figure 3) for a given value by finding the k closest points to the nearest neighbor from the training records.

Consider a new user offloaded his computation and provides all his attributes to the offloading server. Now, the attributes are storied and sent for recommendation system, the engine has to filter and collect only relevant attributes and classify if among a particular group in order to get feedbacks and recommendation service or an advertisement.

For a given mobile user m we need to know in which class c it belongs to, for this a training set $S= \{\{s_1, c_1\}..... \{s_n\}\}$, where $s_j$is the j^{th} element and c_j is its class identity.The k-nearest neighbors will find a subset $T= \{\{t_1, c_1\}....\{t_k\}\}$ such that T S and ● is minimal.T contains the k points in S which are closest to the user m. Then the class c that m belongs to is $c = f(\{c_1...c_k\})$.

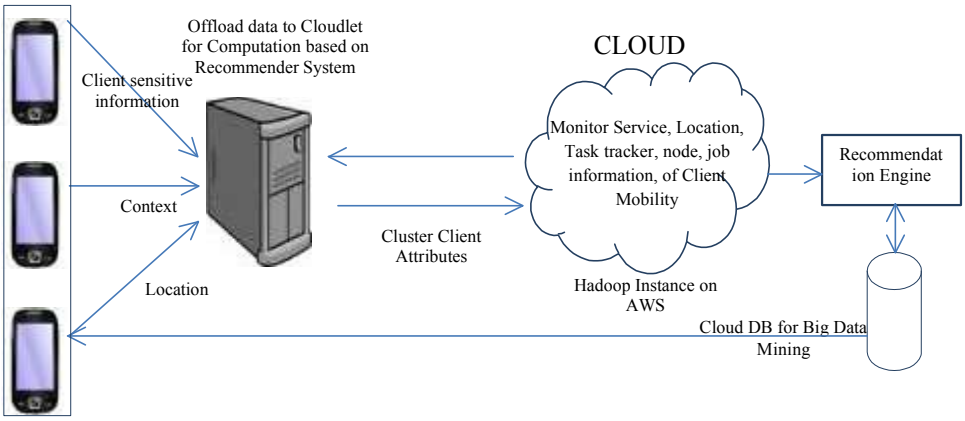

Fig. 4: Recommendation Engine access scenario

Since *k* NN algorithm for classification in recommender system is simplest for known available data and hence it is considered as a lazy learner. Also it classifies for dynamically changing attributes and holds no memory models to run it hence it is a light weight algorithm which is best suited for our mobile recommender engine.

System Design and Implementation: Each mobile phones agent is with a decision support system that receives the recommendations from the engine for prediction analysis. Initially these systems sent the client configurations like context, location and other hardware and software information's to the cloud. The server, that is the cloudlet is developed using LINUX platform Ubuntu 12.04 LTS with a HFS (Hadoop File System). By identifying the service from the client, the cloudlet gets necessary applications from the cloud dynamically and completes the task.

Thus the data that are offloaded from mobile devices are instantly gets computed and the result is returned back to the device, the users assumes that the task is executed in the cloud, whereas in practice it is done at the location of the service requester itself. High intensive applications with less support from cloudlets like image search and mining are forwarded to the cloud to proceed as it is. By analyzing the activities and behavioral pattern of the users around the cloudlet, we can further modify the scenario for computing high end applications also. Whatsoever the intermediate information's like location, timestamp, usage pattern, task tracker, nodes and jobs for every clients mobility is tracked, monitored and saved in the recommendation engine. The cloud here uses an

Amazon Web Services instance of Hadoop cluster as mentioned in the Figure 4. The recommendation engine is created using Apache Mahout with the algorithm of similarity cooccurrence. The client awareness of the users is established by having a quick search from the cloud database for its client ID. This is useful when theclient moves from place to place thereby enabling a continuous connection without any re-registration orhaving to pass the connection setup protocol. Asymmetric key encryption for connection establishment can be provided for a secure communication.

CONCLUSION

The Concept of Mobile Cloud computing is fast emerging with the advancement of mobile device and cloud computing. Having devised an offloading technique which is not usual, that is by adopting a cloudlet framework and by using the MapReduce concepts; we have clearly shown that our model could be a feasible solution in a private environment, where application executions are well known in advance. A well-formed recommendation system with collaborative filtering will enable the clients sensitive information's to be mined for effective prediction of what the user will prefer and whether the decision of offloading the data to cloud or not.

REFERENCES

1. Yang, K., S. Ou and H.H. Chen, 2008. On effective offloading services for resource constrained mobile devices running heavier mobile internet applications, IEEE Communication Magazine, 46(1): 56-63.
2. Umanandhini, Latha Tamilselvan and Udhayakumar, 2012 Dynamic authentication forconsumer supplies in mobile cloud environment, 3rd ICCCNT, Coimbatore, India, pp: 1-6.

3. Karthik Kumar and Yung-Hsiang Lu, 2010. Cloud computing for mobile users: Can offloading computation save energy. In IEEE Computer Society. IEEE Press.
4. Satyanarayanan, M., P. Bahl, R. and N. Davies, 2009. The case for vm-based cloudlets in mobile computing. IEEE Pervasive Comp., 6(4): 14-23.
5. Byung-Gon Chun, S. Ihm, P. Maniatis, M. Naik and A. Patti, 2011. Clonecloud: elastic execution between mobile device and cloud, in: Proceedings of the Sixth Conference on Computer Systems, EuroSys'11, Salzburg, Austria, ACM, NY, pp: 301-314.
6. Aaron Yi Ding, *et al.,* 2013. Enabling Energy-Aware Collaborative Mobile Data Offloading for Smartphones" Proceedings of 10th IEEE International Conference on Sensing, Communication and Networking, USA.
7. Patil, V.C., *et al.,* 2011. Information and Communication technologies for Agriculture Knowledge Management in India, World Applied Sciences Journal, 14(5): 794-802.
8. Ehsan Emadzadeh, *et al*., 2010. Learning Material Recommendation Using a Hybrid Recommender System with Automated Keyword Extractor, World Applied Sciences Journal, 9(11): 1260-1271.
9. Ijaz Ali, Abdullah Al-Dhelaan and Mohsin, 2013. A Virtualized Information Indexing and Filtering Method for Web Contents Reside on Remotely Communicated Networks, World Applied Sciences Journal, 23(8): 1053-1060.
10. Mohammed Anowarul and Songqing, 2012. Mobile MapReduce: Minimizing Response Time of Computing Intensive Mobile Applications. LNCS-Springer, 95: 41-59.
11. Ibrahiem, El Emary and S. Ramakrishnan, 2008. On the Application of Various Probabilistic Neural Networks in Solving Different Pattern Classification Problems, World Applied Sciences Journal, 4(6): 772-780.