International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Novel Approach of Distributed Cooperative Caching in Social Wireless Networks

A. Kalyankumar[1], K. V. Srinivasa Rao[2]

[1]M.Tech student, CSE Department, Prakasam Engineering College, Kandukur, Prakasam (Dt), India

[2]Associate Professor, CSE Department, Prakasam Engineering College, Kandukur, Prakasam (Dt), India

Abstract: ***This paper introduces cooperative caching policies for minimizing electronic content provisioning cost in Social Wireless Networks (SWNET).SWNETs are formed by mobile devices, such as data enabled phones, electronic book readers etc., sharing common interests in electronic content, and physically gathering together in public places. Electronic object caching in such SWNETs are shown to be able to reduce the content provisioning cost which depends heavily on the service and pricing dependences among various stakeholders including content providers (CP), network service providers, and End Consumers (EC). Drawing motivation from Amazon's Kindle electronic book delivery business, this paper develops practical network, service, and pricing models which are then used for creating two object caching strategies for minimizing content provisioning costs in networks with homogenous and heterogeneous object demands. The paper constructs analytical and simulation models for analyzing the proposed caching strategies in the presence of selfish users that deviate from network-wide cost-optimal policies. It also reports results from an Android phone based prototype SWNET, validating the presented analytical and simulation results.***

Keywords: SWNET, ASMCC, IAAS, SAAS

1. Introduction

Mobile cloud computing has received large interest recently as it allows storage and processing of data outside the mobile device. It has a growing popularity due to the proliferation of smart phones which act as mini PCs. The limitations of the mobile device such as smaller size, low battery life and other features can be overcome by offloading the processing and storage to a cloud. The offloading can happen to a remote data center, nearby computer or cluster of computers, or even to nearby mobile devices.

Cloud computing is a frame work for sharing resources, information and software capabilities to different mobile devices. The resources will be available on the cloud and can be shared by the devices on demand. In mobile cloud computing environment the client can use the cloud to back up data in the mobile devices. Generally, there are two approaches to realize mobile cloud computing namely General Purpose Mobile Cloud Computing (GPMCC) and an Application Specific Mobile Cloud Computing (ASMCC) [5]. GPMCC is utilizing the internet by the mobile devices to use the computing resources of remote computers without any applications specifically developed for this purpose. In ASMCC, specific applications are developed for mobile devices to use the cloud computing facility. Mobile Service Clouds proposed in [1] is a cloud service which uses ASMCC approach for the deployment of autonomic communication services. In [4], mobile cloud computing is broadly classified in into two, those which use mobile devices as thin clients, offloading computation to cloud resources on the internet and the one using mobile devices as computational and storage nodes as a part of cloud computing infrastructure. Although mobile devices have improved much in processing speed, memory and operating systems, they still have some serious drawbacks. The major challenge for a mobile device in cloud computing is the data transfer bottle neck. Battery is the major source of energy for these devices and the development of battery technology has not been able to match the power requirements of increasing resource demand. The average time between charges for mobile phone users is likely to fall by 4.8% per year in the near future [2]. As the cloud grows in popularity and size, infrastructure scalability becomes an issue. Without scalability solution, the growth will result in excessively high network load and acceptable service response time.

Data caching is widely used in wired and wireless networks to improve data access efficiency, by reducing the waiting time or latency experienced by the end users. A cache is a temporary storage of data likely to be used again. Caching succeeds in the area of computing because access patterns in typical computer applications exhibits locality of reference [3]. Caching is effective in reducing bandwidth demand and network latencies. In wireless mobile network, holding frequently accessed data items in a mobile node's local storage can reduce network traffic, response time and server load. To have the full benefits of caching, the neighbor nodes can cooperate and serve each other's misses, thus further reducing the wireless traffic. This process is called cooperative caching. Since the nodes can make use of the objects stored in another node's cache the effective cache size is increased. In this paper we discuss a cooperative cache based data access frame work for mobile cloud computing.

2. Related Work

With the existence of such SWNETs, an alternative approach to content access by a device would be to first search the local SWNET for the requested content before downloading it from the CP's server. The expected content provisioning cost of such an approach can be significantly lower since the download cost to the CSP would be avoided when the content is found within the local SWNET. This mechanism is termed as cooperative caching. In order to encourage the End-Consumers (EC) to cache previously downloaded content and to share it with other end-consumers, a peer-to-peer rebate mechanism is proposed. This mechanism can serve as an incentive so that the end-consumers are enticed to participate in cooperative content caching in spite of the storage and energy costs. In order for cooperative caching to provide cost benefits, this peer-to-peer rebate must be dimensioned to be smaller than the content download cost paid to the CSP. This rebate should be factored in the content provider's overall cost. Due to their limited storage, mobile handheld devices are not expected to store all downloaded content for long. This means after downloading and using a purchased electronic content, a device may remove it from the storage.

In this paper drawing motivation from Amazon's Kindle electronic book delivery business, this paper develops practical network, service, and pricing models which are then used for creating two object caching strategies for minimizing content provisioning costs in networks with homogenous and heterogeneous object demands. The paper constructs analytical and simulation models for analyzing the proposed caching strategies in the presence of selfish users that deviate from network-wide cost-optimal policies. It also reports results from an Android phone based prototype SWNET, validating the presented analytical and simulation results.

3. Mobile Cloud Architecture

The general architecture of MCC is shown in Fig1. The mobile clients are connected to the internet via base stations, access points or by a satellite link. The shared pool of resources in mobile cloud computing are virtualized and assigned to a group of distributed servers managed by the cloud services. The cloud services are generally classified based on a layered concept [6].The frame work is divided in to three layers, Infrastructure as a Service (IaaS), Platform as Service(PaaS) and Software as a Service (SaaS).

Infrastructure as a Service (IaaS): IaaS includes the resources of computing and storage. It provides storage, hardware, servers and networking components to the user. The examples of IaaS are Elastic Cloud of Amazon and S3 (Simple Storage Service). Platform as a Service (PaaS): Paas provides an environment of parallel programming design, testing and deploying custom applications. The typical services are Google App engine and Amazon Map Reduce/Simple Storage Service. Software as Service (SaaS): SaaS provides some software and applications which the users can access via Internet and is paid according to the usage. Google online office is an example for SaaS.

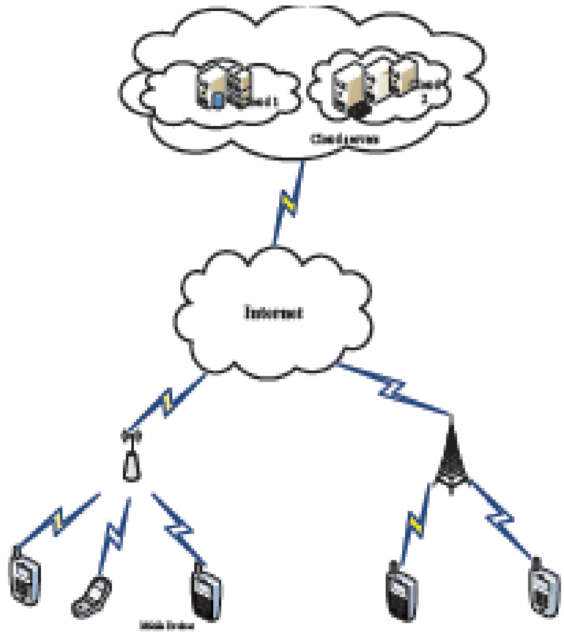

Figure 1:.Mobile Cloud Computing Architecture

3.1 Major Issues in Mobile Cloud Computing

The key elements in a mobile cloud computing approach are: mobile devices, networks through which the devices communicate with the cloud and mobile applications. The major challenge in cloud computing comes from the characters of the first two elements, mobile devices and wireless network .This makes the implementation of mobile cloud computing more complicated than for fixed clouds. This section lists the major issues in Mobile Cloud Computing. Limitations of the Mobile devices: Compared to personal computers mobile devices have limited storage capacity, poor display, less computational power and energy resource. Although smart phones have improved a lot, they still have battery power constraint. Network Bandwidth and Latency: As the mobile cloud computing uses wireless networks for data transfer bandwidth is a major issue compared to wired networks which uses a physical connection to ensure bandwidth consistency. Furthermore, the cloud services may be located far away from mobile users, which in turn increase the network latency.

Heterogeneity: Heterogeneity in mobile cloud computing comes from two sources: mobile devices and mobile networks. There is a wide range of mobile devices used by the group of people sharing then network. The operating system and the application software used by these devices vary which cause a major issue in the interoperability of the devices. Another area is the different radio technologies used for accessing the cloud. This will lead to changes in bandwidth and network overlay. Service Availability: Availability of service is an important issue in mobile cloud computing. Mobile clients may not be able to connect to the cloud due to traffic congestion, network failures and out of signal. Privacy and Security: Offloading computation and storage to cloud pose security and trust issues. The cloud services are vulnerable and the mobile clients may lose their data if the services fail due to some technical issues.

3.2 Cloudlet Architecture

One of the key issues in mobile cloud computing is the end to end delay occurring in servicing a request. Since the cloud services and resources in Internet service provider is distant from the mobile clients network latency is increased. The proposed an architecture for mobile cloud computing called Cloudlet, to reduce the bandwidth induced delay between devices and cloud. A cloudlet is a set of computers connected to the internet and is accessed by the nearby mobile devices using a Wi-Fi or WLAN. In this architecture the mobile devices act as a thin client and all the computations occur in the cloudlet. Fig 2 shows cloudlet architecture. The basic idea of this approach is to reduce the distance between the mobile users and cloud services. By this architecture mobile users can access the cloudlet which is one hop away, thus reduces bandwidth utilization and efficiency.

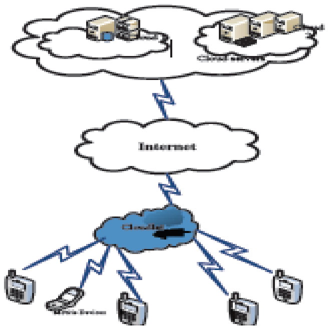

Figure 2: Mobile Cloudlet Architecture

To overcome the challenges of mobile devices, parts of computation and data storage is migrated to resource providers outside the mobile device. Offloading transfers the information needed for processing and storage to a remote server which will complete the computation task and send the results back to the mobile client. The different offloading techniques currently available are client server communication, virtualization and mobile agents [20]. Cloudlet architecture reduces the gap between the mobile devices and remote servers by offloading its workload to a local cloudlet with connectivity to remote cloud servers. Offloading in cloudlet infrastructure is based on a Virtual Machine (VM) technology. To create a transient client software environment cloudlet uses dynamic VM synthesis. In this approach the mobile device transmits a small VM overlay to the cloudlet and applies it to a compatible base VM to generate the Launch VM. Launch VM is the virtual environment temporarily created for a mobile client to execute the task. After execution results are given back to the mobile client. The cloudlet infrastructure is restored to its previous state after each execution.

The performance of cloudlet depends mainly on two factors. Overlay transmission time and overlay synthesis. Overlay transmission time can be improved by using a higher bandwidth wireless network. The different techniques proposed in [7] includes partitioning the overlay in to different chunks that can be executed in parallel, caching and perfecting and pipelining i.e. pipelining the overlay execution with transmission. In this paper we discuss a cooperative cache design for the cloudlet architecture. Mobile cloud computing has found wide applications in many areas like speech synthesis, natural language processing, image processing, augmented reality, information sharing ,information searching , social networking, etc. While many applications like information sharing or social networking are not dependent on the speed of processing, some computation intensive applications like augmented reality, image processing demand high level of responsiveness. Cooperative caching tries to improve the response time by reducing VM synthesis time by caching previous states. If the users that use cloud services have similar interest, cooperative caching increases the response time considerably. A language translator is an interesting application, which we could look into. This is a useful tool for foreign travelers. Using mobile cloud computing, different words, sentences or paragraphs can be independently processed in the cloud. Commonly used words or sentences will be available in the local cache, which can be accessed faster during subsequent searches, thereby improving the responsiveness of the system.

Cooperative caching consists of multiple distributed caches to improve system response time. Having distributed caches permits a system to deal with concurrent client request as well as sharing contents. We can also reduce response time by concurrently retrieving objects from different cache sites. Concurrent retrieval of objects from different cache sites is beneficial as opposed to the remote cloud server which will result in latency and bandwidth issues.

In cloudlet, when a mobile client requests for a cloud service, the network searches for data in the local cloud. If the service is not available, the users should contact a distinct cloud which involves network transfers and latency. If we are able to cache different VM synthesis states, the users can get the service from cache. When the object is not present in the cache, request is given to the base layer to get the corresponding launch VM. If t the corresponding base h VM is not present, we have to contact distant cloud for the service. Thus data caching also increases battery life in mobile devices by reducing wireless communication.

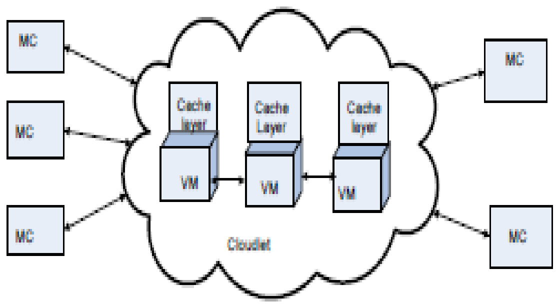

Figure 3: Cooperative Cache Framework

3.3 Optimal Solutions

For contents with changeable level of popularity, a greedy approach for each node would be to store as many distinctly popular contents as its storage allows. This approach sums to noncooperation and can grow to heavy network-wide data duplications. In the other excessive case, which is fully cooperative, a terminal would try to make the best of the total number of single contents stored within the SWNET by avoiding duplication.

In this paper, we show that none of the above excessive approaches can reduce the content provider's charge. We also show that for a given rebate-to-download-charge ratio, there is present an object placement policy which is somewhere in between those two ends, and can increase the content provider's cost by striking a stability between the greediness and full cooperation. This is referred to as optimal object placement policy in the rest of this paper. The proposed cooperative caching algorithms strive to attain this best object placement with the target of reducing the network-wide content provisioning price.

3.4 User Selfishness

The probability for earning peer-to-peer rebate may encourage selfish activities in some clients. A selfish client is one that diverges from the network-wide finest policy in order to receive more rebates. Any distinction from the optimal policy is expected to incur higher network-wide provisioning cost. In this work, we revise the impacts of such selfish behavior on object provisioning cost and the earned refund within the context of a SWNET. It is given that beyond a threshold selfish node population, the amount of per-node rebate for the selfish users is lower than that for the unselfish users. In supplementary terms, when the selfish terminal population exceeds a certain point, selfish actions discontinue producing more advantage from a refund standpoint.

4. Network, Service and Pricing Model

Network Model Fig.4 describes a model SWNET within a University grounds. People carrying mobile devices form SWNET partitions are the end consumers, which can be whichever multi-hop (i.e. MANET) as shown for partitions 1, 3, and 4, or single hop contact point based as shown for partition 2. A movable device can download some data (i.e., content) from the CP's server using the CSP's cellular system, or from its home SWNET partition. In the remaining paper, the terms object and content are used synonymously. We regard as two types of SWNETs. The foremost one involves motionless SWNET partitions. Meaning, after a partition is formed, it is maintained for sufficiently long so that the cooperative object caches can be formed and reach fixed states. We also consider a second type to explore as to what happens when the still assumption is relaxed. To investigate this effect, caching is applied to SWNETs formed using human interaction traces obtained from a set of real SWNET nodes.

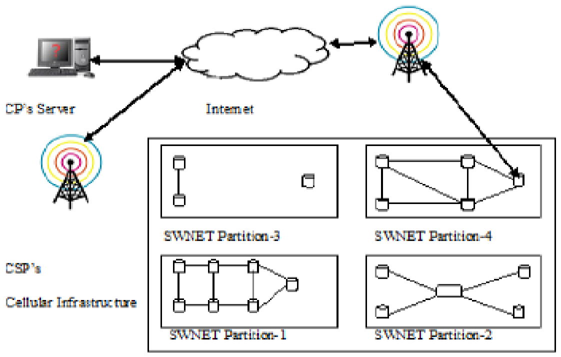

Figure 4: Content access from a SWNET in a University Campus

4.1 Search Model

After an object call is originated by a mobile tool, it first finds in its local cache. If the local search fails, it searches the object within its SWNET division using limited transmit note. If the search in division also fails, the data is downloaded from the CP's server using the CSP's 3G/4G cellular arrangement. In this paper, we have designed objects such as electronic books, music, etc., which does not vary on time, and therefore cache constancy is not a serious issue. We first suppose that all objects have the equivalent size and each terminal is able to store up to "C" dissimilar data in its cache. Later on, we let go this supposition to sustain objects with variable size. We also believe that all objects are popularity-tagged by the CP's server. The popularity-tag of an object points out its universal recognition; it also indicates the chances that a subjective request in the network is produced for this specific object.

4.2 Pricing Model

We use a pricing model similar to the Amazon Kindle business model in which the CP pays a download cost Cd to the CSP when an End-Consumer downloads an object from the CP's server through the CSP's cellular network. Also, whenever an EC provides a locally cached object to another EC within its local SWNET partition, the provider EC is paid a rebate Cr by the CP. Optionally, this rebate can also be distributed among the provider EC and the ECs of all the intermediate mobile devices that take part in content forwarding .The selling price is directly paid to the CP by an EC through an out-of-band secure payment system. A digitally signed rebate framework needs to be supported so that the rebate recipient ECs can electronically validate and redeem the rebate with the CP. We assume the presence of these two mechanisms on which the proposed caching mechanism is built.

5. Results

Figure 1: Admin Login

Figure 2: File Upload

Figure 3: User Searching the File

Figure 4: Downloading a File

6. Conclusion

The objective of this work was to develop a cooperative caching strategy for provisioning cost minimization in Social Wireless Networks. The key contribution is to display that the best cooperative caching for provisioning cost reduction in networks with homogeneous content demands requires an optimal split between object duplication and individuality. Furthermore, we experimentally (using simulation) and analytically evaluated the algorithm's performance in the presence of user selfishness. It was shown that selfishness can increase user rebate only when the number of selfish nodes in an SWNET is less than a critical number. It was shown that with heterogeneous requests, a benefit based heuristics strategy provides better performance compared to split cache which is proposed mainly for homogeneous demand. Ongoing work on this topic includes the development of an efficient algorithm for the heterogeneous demand scenario, with a goal of bridging the performance gap between the Benefit Based heuristics and the centralized greedy mechanism which was proven to be optimal Removal of the no-collusion assumption for user selfishness is also being worked on.

References

[1] M. Zhao, L. Mason, and W. Wang, "Empirical Study on Human Mobility for Mobile Wireless Networks," Proc. IEEE Military Comm. Conf. (MILCOM), 2008.

[2] "Cambridge Trace File, Human Interaction Study," http://www.crawdad.org/download/cambridge/haggle/Exp6.tar.gz,2012.

[3] E. Cohen, B. Krishnamurthy, and J. Rexford, "Evaluating Server-Assisted Cache Replacement in the Web," Proc. Sixth Ann.European Symp. Algorithms, pp. 307-319, 1998.

[4] S. Banerjee and S. Karforma, "A Prototype Design for DRM Based Credit Card Transaction in E-Commerce," Ubiquity, vol. 2008,2008.

[5] L. Breslau, P. Cao, L. Fan, and S. Shenker, "Web Caching and Zipf-Like Distributions: Evidence and Implications," Proc. IEEE INFOCOM, 1999.

[6] C. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," Proc. IEEE Second Workshop Mobile Systems and Applications,1999.

[7] S. Podlipnig and L. Boszormenyi, "A Survey of Web Cache Replacement Strategies," ACM Computing Surveys, vol. 35, pp. 374-398, 2003.

[8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott,"Impact of Human Mobility on Opportunistic Forwarding Algorithms," IEEE Trans. Mobile Computing, vol. 6, no. 6, pp. 606-620, June 2007.

[9] "BU-Web-Client - Six Months of Web Client Traces," http://www.cs.bu.edu/techreports/1999-011-usertrace-98.gz, 2012.

[10] A. Wolman, M. Voelker, A. Karlin, and H. Levy, "On the Scale and Performance of Cooperative Web Caching," Proc. 17th ACM Symp. Operating Systems Principles, pp. 16-31, 1999.

[11] S. Dykes and K. Robbins, "A Viability Analysis of Cooperative Proxy Caching," Proc. IEEE INFOCOM, 2001.

[12] M. Korupolu and M. Dahlin, "Coordinated Placement and Replacement for Large-Scale Distributed Caches," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 6, pp. 1317-1329, Nov. 2002.

[13] L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 1, pp. 77-89, Jan. 2006.

[14] Y. Du, S. Gupta, and G. Varsamopoulos, "Improving On-Demand Data Access Efficiency in MANETs with Cooperative Caching," Ad Hoc Networks, vol. 7, pp. 579-598, May 2009.

[15] C. Chow, H. Leong, and A. Chan, "GroCoca: Group-Based Peerto-Peer Cooperative Caching in Mobile Environment," IEEE J. Selected Areas in Comm., vol. 25, no. 1, pp. 179-191, Jan. 2007.

[16] F. Sailhan and V. Issarny, "Cooperative Caching in Ad Hoc Networks," Proc. Fourth Int'l Conf. Mobile Data Management, pp. 13-28, 2003.

[17] C. Chow, H. Leong, and A. Chan, "Peer-to-Peer Cooperative Caching in Mobile Environments," Proc. 24th Int'l Conf. Distributed Computing Systems Workshops, pp. 528-533, 2004.

[18] A. Schrijver, Theory of Linear and Integer Programming. Wiley-Interscience, 1986.

[19] H.K. Kuhn, "The Hungarian Method for the Assignment Problem," Naval Research Logistics, vol 52, no. 1, pp. 7-21, 2005.

[20] N. Laoutaris et al., "Mistreatment in Distributed Caching: Causes and Implications," Proc. IEEE INFOCOM, 2006.

[21] B. Chun et al., "Selfish Caching in Distributed Systems: A Game-Theoretic Analysis," Proc. 23th ACM Symp. Principles of Distributed Computing, 2004.

[22] M. Goemans, L. Li, and M. Thottan, "Market Sharing Games Applied to Content Distribution in Ad Hoc Networks," IEEE J. Selected Areas in Comm., vol. 24, no. 5, pp. 1020-1033, May 2006.

Author Profile

Mr. A. Kalyan Kumar received B. Tech. degree in Computer Science and Engineering from Chebrolu Engineering College Guntur from JNTU Kakinada University in 2012 currently. He is doing M. Tech. in Prakasam Engineering College, from JNTUK University, Kakinada, India.

K. V. Srinivasa Rao received M.TECH degree in Computer Science and Engineering from JNTUA University, and currently he is working as an Associate Professor, Department of CSE in Prakasam Engineering College, Kandukur, India