International Journal of Advanced Technology and Innovative Research
ISSN 2348–2370
Vol.06,Issue.04,
June-2014,
Pages:198-201

www.semargroup.org

Network Failure Recovery Model by using Backup Topology

G. ABHINAV[1], CH. RATHAN KUMAR[2], N. SATHISH KUMAR[3]
[1]PG Scholar, Dept of CSE, S.V.S Group of Institutions, Warangal, A.P, India, Email: abhinav.sai4u@gmail.com.
[2]Asst Prof, Dept of CSE, S.V.S Group of Institutions, Warangal, A.P, India, Email: Rathankumar505@gmail.com.
[3]Professor, Dept of CSE, S.V.S Group of Institutions, Warangal, A.P, India, Email: dean@svsit.ac.in.

Abstract This paper introduces cooperative caching techniques for reducing electronic content provisioning cost in Social Wireless Networks (SWNET).SWNETs are formed by mobile devices, such as modern cell phones etc. sharing common takings in electronic content, and actually meeting 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).This paper develops practical network, service, and pricing models which are then used for creating two object caching strategies for reducing content provisioning costs in networks with homogenous and heterogeneous object demands. This paper studies the distributed caching managements for the current flourish of the streaming applications in multi hop wireless networks. Many caching managements to date use randomized network coding approach, which provides an elegant solution for ubiquitous data accesses in such systems. However, the encoding, essentially a combination operation, makes the coded data difficult to be changed. In particular, to accommodate new data, the system may have to first decode all the combined data segments, remove some unimportant ones, and then re encode the data segments again. It also reports results from an Android phone based prototype SWNET, validating the presented analytical and simulation results.

Keywords: Cooperative Cache; Cost-Optimal Policies; Distributed Search Engines; Selfish Users.

I. INTRODUCTION

A. Motivation

Modern appearance of data enabled mobile devices and Wireless-enabled data applications have fostered new content dissemination models in today's mobile ecosystem. A record of such devices includes iPhone, Android phones Amazon's kindle and electronic book readers from additional vendors. The set of data applications includes mobile phone apps. The level of proliferation of mobile applications is indicated by the example fact that as of October 2010, apple's app store offered over 100,000 apps that are downloadable by the smart phone users. With the conventional download model, a user downloads contents directly from a content providers' (CP) server over a communication service providers' (CSP) network. Downloading data through CSP's group involves a price which must be paid either by customers or by the content provider. In this effort we take on Amazon kindle electronic book delivery selling model in which the CP (Amazon) pays to sprint, the CSP, for the cost of network usage due to downloaded electronic books by kindle clients.

B. 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.

C. 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.

II. CACHE DEPLOYMENT OPTIONS

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. 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 unacceptable 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. Caching is effective in reducing bandwidth demand and network latencies.

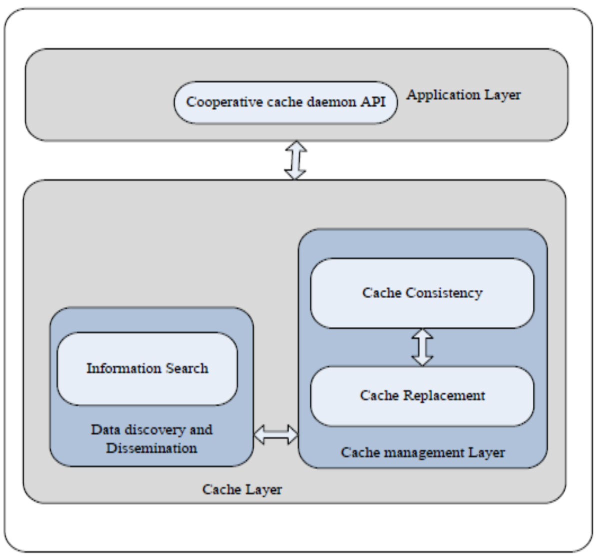

Fig.1. Different components of cache layer.

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. There are two main cache deployment options: those which are deployed in the strategic points in cloudlet based on user access pattern and those which are deployed between the cloudlets. In this paper we consider the first option, deploying cache in different points (virtual machines) in the cloudlet. The cooperative cache frame work for cloudlet architecture. The cloudlet consists of virtual machines which are temporary customization of software environment for each client for their use. The virtual machines separate the transient client software environment from the permanent host software environment. A local cache can reduce virtual machine's synthesis delay by caching virtual machine states that are likely be used again. In a cloudlet we can have more than one virtual machine with a local cache. If we are able to share the cache states, availability and accessibility of different states can be improved. Fig.1 shows the different components of cache layer.

The cooperative cache daemon API acts as an interface between the application layer and the cache layer. The core system consists of two modules: data discovery and dissemination and the cache management. The information search module in the data discovery and dissemination layer locates and fetches the required object from the cache module. The cache management layer includes the cache replacement and consistency modules. Cache consistency module is designed to be configurable to maintain data synchronization with the original data. The cache replacement module handles the replacement of objects when the cache is full. The efficiency of a distributed cache depends on three services, discovery, dissemination and delivery of objects. Discovery refers to how the clients locate the cached object. Dissemination is the process of selecting and storing objects in the cache i.e., deciding the objects to be cached, where they are cached and when they are cached. Delivery defines how the objects make their way from the server or cache site to the client. A query based or directory based approach can be used for information discovery. Dissemination may be either client initiated or server initiated.

In client initiated dissemination, the client determines what, when and where to cache. The advantage of this scheme is that it automatically adapts to the rapidly changing request pattern. In server initiated dissemination the server chooses the object to be cached. Here the server can maintain a historical data to make the dissemination decision. This approach can provide strong consistency compared to client driven approach. For the proposed approach as the mobile devices act as thin client dissemination decision can be taken by the cloudlet. Another issue we must look into is how to replace the objects from the cache when it is full. A number of cache replacement policies are proposed in literature for wired and wireless networks. The important factors that can influence the replacement process are access probability, regency of request for a data item, number of requests to a data item, size, cost of fetching data from server, modification time, expiration time, distance etc. Based on these parameters we can propose different cache replacement policies suitable for mobile cloud computing. Cooperative caching achieves high hit rates and low response time only if caches are distributed, cache sharing is wide spread and discovery overhead is low.

III. MOBILE CLOUD ARCHITECTURE

The general architecture of MCC is shown in Fig 2. 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. 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.

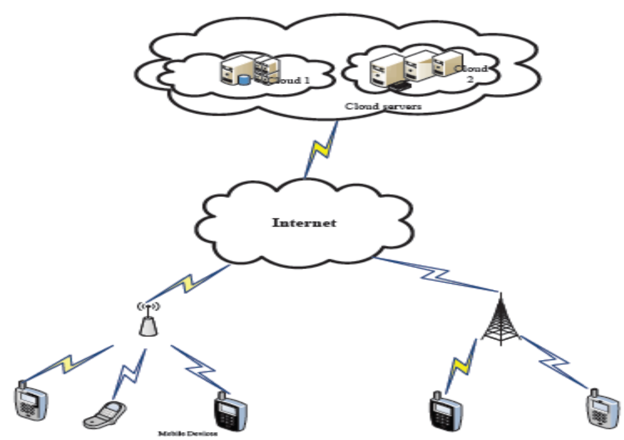

Fig.2. Mobile Cloud Computing Architecture.

A. Applications of Mobile Cloud Computing

Mobile cloud applications move the resource intensive applications and storage away from mobile phones so that the mobile applications are not constrained to certain kind of mobile devices. This helps to overcome the storage capacity and computing power constraints of mobile devices. Mobile cloud computing paradigm is an attractive option for many areas like business, mobile image processing and for computing intensive applications like speech recognition, machine learning augmented reality etc. The typical applications of mobile cloud computing includes Mobile Commerce, Mobile Learning, Mobile Healthcare, Mobile Gaming and other practical applications like social networking, showing maps, storing images and video.

B. 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.

1. **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.
2. **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 the 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.
3. **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.

V. 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.

VI. REFERENCES

[1] Dimple Pandya, Manan Shah, Manvi Sinha, Sanika Patil, Anand Magar, "Distributed Cooperative Caching In Social Wireless Network", 2248-9622, Vol. 4, Issue 4(Version 1), April 2014, pp.335-338.

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

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

[4] 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.

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

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

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

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

[9] 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.

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

[11] 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.

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