BALANCED CONTENT ALLOCATION SCHEME FOR PEER-SERVICE AREA CDN ARCHITECTURE FOR IPTV SERVICES

Multimedia Research Group
Universiti Sains Malaysia

smfati@yahoo.com

Putra Sumari

Head of Multimedia Research Group
Universiti Sains Malaysia

putras@cs.usm.my

Rahmat Budiarto

InterNetworks Research La6
School of Computing
UUM College of Atrs and Sciences
Universiti Utara Malaysia

rahmat@uum.edu.my

ABSTRACT

One of the main problems in IPTV technology is how to manage the huge amount of multimedia contents efficiently to meet the demands of users especially for Video on Demand (VoD) services. Content Distribution Networks (CDN) are used to solve this problem but the problem of load imbalance among servers still exists due to the dynamic changes in contents and user interests in an IPTV environment. In the VoD context, many content storage management architecture models are proposed: single point, hierarchal, distributed, and service peer area architectures. In the this paper we choose peer-service area architecture for CDN to study the load imbalance problem and try to handle it by modifying peer-service area architecture and proposing a balanced content allocation scheme that solves the

load imbalance problem by replicating the contents based on their popularity. Experimental results show that this proposed allocation scheme can maintain the load balancing among servers and avoid over/under utilization of servers.

Keywords: IPTV, CDN, content allocation, peer-service area architecture, content popularity, load imbalance, balanced content allocation, VOD.

INTRODUCTION

IPTV improves the delivery of TV related services to be transported over IP-based networks to benefit from the high speed of these networks (Lee, Muntean & Smeaton 2009; Mandal & Mburu, 2008). IPTV services became popular due to the competition of operators during the last few years since it can deliver high quality viewing service at any time (Yarali, 2007; Li & Wu, 2010). The IPTV industry witnesses a rapid growth where the subscribers increased from 2.03 million in 2005, and 4.56 million in 2006 to reach 46.2 million in 2010 and 60 million in 2011, and is also expected to occupy a third of the TV viewing markets in 2012 (Cheng, 2007). The subscribers are expected to approximately double in 2015 to reach 131.6 million (Gupta, 2011). IPTV can provide Live TV, Video on Demand (VoD), and any additional value-added service through the QoS guaranteed IP-based networks using the triple play concept (Gu & Nah, 2008). One of the main issues related to IPTV technology is how to efficiently store the huge amount of multimedia data for reusability purposes within the constraints of limited storage and bandwidth capacity to achieve the goals of both providers and customers (Krogfoss Sofman & Agramal, 2008; Doverspike, Li, Oikonomou, Ramakrishnan, Sinha, Wang & Chase, 2009; and Song, Hassan & Huh, 2011). So, Content Distribution Networks (CDN) become an optimal solution to distribute these multimedia contents over a set of servers among a wide geographical area to reduce the overload on the backbone network and at the same time satisfy the customers' needs efficiently (Nakaniwa & Ebora, 2007; Cranor Ethington,Shgal, Shur & Sreenan, 2003; Plagemann, Goebel, Mauthe, Mathy, Turletti & Urvoy-Keller, 2006; Kim et al., 2006).

However, in a dynamic system like IPTV, the contents are increasing massively, so the management process of these contents is considered a crucial point in achieving a successful IPTV system which still needs more investigation to build efficient and cost-effective architecture without violating the load balancing constraints among storage servers.

In the VoD context, many content storage management architecture models are proposed: single point architecture in which all clients are connected to a single server that stores all the multimedia contents. The main disadvantage of single point architecture is the single point of failure. To reduce the load on the main server, many cache servers are allocated within networks to distribute the load among them (distributed model). The hierarchal architecture is proposed to improve the reliability and QoS level but the cost of this architecture is very high. A novel model called peer-service area architecture is proposed by Li and Wu (2010), in which the CDN is divided into many service areas with a cluster of servers for each. The customer has to belong to only one service area and can request the video from any server within his service area. The requested video that does not exist in the service area must be redirected to the nearest service area. According to Li and Wu (2010), this architecture can satisfy the QoS requirement and also the reliability; and they stated that it is very suitable for IPTV services.

From the perspective of load balancing, this architecture has the following limitations: (1) storing popular contents in special servers (Type 1 servers) and unpopular contents in other separated servers (Type 2 servers) may leads to overutilizaing the servers of popular contents (Type 1 servers) while the servers of unpopular contents are still underutilized because of the popular contents that attract most of the users' requests. (2) storing popular videos without replication may lead to ahigh rejection rate of the users, requests and then degrade the reliability and QoS level. So, the replication process allows us to store popular contents in more than one place which leads to distributing the load of that content.

Based on the aforementioned limitations, the load of the overall system will be imbalanced and may cause a high rejection rate of users' requests. So, we propose a modified peer-service area architecture that overcomes these problems in order to build a balanced CDN for IPTV services. In our proposed model as depicted in Fig.1, the servers in each service area will be considered the same type and the popular videos will be replicated based on their popularity and the available number of servers to prevent redundant replication and also prevent under-replication that may lead to rejection of user requests. Another feature of our proposed architecture is to add a request dispatcher to each service area that controls the distribution of requests for a certain video among the servers containing a copy of the requested video. The load dispatcher is out of this paper's scope.

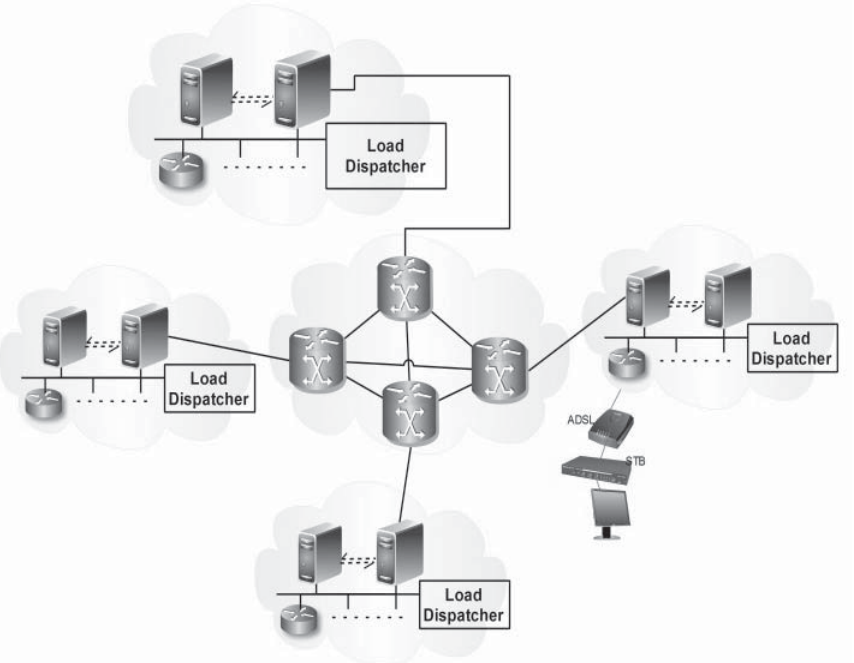

Figure 1. Balanced proposed peer-service area architecture for CDN.

In this paper, the main contribution is to demonstrate the load balance of the proposed architecture in Fig.1 by building a balanced content allocation scheme based on the expected load of contents.

RELATED WORKS

Content allocation is considered an important point in designing the Content Distribution Network (CDN) for IPTV technology; many studies have been proposed to solve the problem of content allocation. These studies can be classified into central, hierarchical, distributed, and finally peer-service area models based on the network architecture that is exploited. In the central model, the authors allocate the contents into an array of disks for single servers using striping, replication, or both. In Scheuermann, Weikum and Zabback (1998), the file is divided and then associated with heat ratio to allocate it to the disk with the lowest heat. Unlike the heat ratio-based allocation, Choe (2007) replicated each stripped block of content randomly into two disks. Tang, Wong, Chan and Ko (2004) considered the content allocation to multiple disks

as an optimization problem to minimize both storage capacity and waiting time and solved it using the Hybrid Genetic Algorithm. Genetic algorithm was also incorporated with modified Bin-packing algorithm by (Tang Ko Chan & Wong, 2001) to allocate contents with minimum storage capacity and block probability. The trade-off between the storage capacity and concurrent access for each video is discussed in Wang, Liu, Du and Hsieh (1997), to find the optimal allocation on RAID-3 based on this trade-off.

In the hierarchical model, the proposed schemes tried to allocate the contents as close to the users as possible to increase the availability of data and minimize the waiting time of users. In Lin, Lai and Lai (1996), the contents are divided into three classes based on their popularity: 1st class of popular contents stored in the Local Service Center (LSC), 2nd class of popular contnets stored in the Local Central Service Center (LCSC), and 3rd class which will be stored in the Central Service Center (CSC) beside copies of members of the 1st and 2nd class. The cost function of capacity and links between the three levels are used to determine the number of movies in LSC and LSCS. The videos can also be associated with weights to classify the popular from the unpopular movies as discussed in Cholvi and Segarra, (2008) who replicated the popular movies into the leaf cache servers and allocated the unpopular movies into the node 0 (main servers). The threshold value is used in Brubeck and Rowe (1996) to decide which movies are popular (bigger than the threshold) and which movies are unpopular (less than the threshold) in order to replicate the popular movies into the cache servers and discard the others from the replication process. Tsao, Chen, Ko, Ho and Huang (1999) took into consideration the connectivity and access probability of each server to produce a balanced content allocation. They replicated the high ratio movies to the cache servers with the lowest connectivity and access probability based on the determined number of copies. The low ratio videos will be stored in tertiary storage devices. Fetching distance as cost function is used by Laoutaris, Zissimopoulos and Stavrakakis, (2005) to optimize the content allocation process. Greedy heuristic algorithm is proposed to allocate content based on cost function. Greedy algorithm is also proposed to minimize the storage capacity by eliminating the replicas from the ancestor nodes if the video is already stored in their leaf nodes. Nakaniwa and Ebara (2007) proposed optimal content allocation by maximizing the system reliability as an objective function and satisfying time delay as a constraint. SMART servers are proposed by Kim, Bak, Woo, Lee, Min and Kim, (2006) to distribute the contents efficiently from global server to local servers. Dynamic Programming is exploited in Cidon, Kutten and Soffer (2001) to allocate contents among the network nodes to minimize the storage and transmission cost of the contents. Bisdikian and Patel (1996) replicated the most requested contents into n*k nodes where N: number of nodes and K: number of servers.

Unlike the hierarchical model, in the distributed model, the servers are allocated in wide geographic areas without central control and users can access the movie from any site via the User Interface Module. The video object can be stripped, distributed and also replicated sequentially into many storage nodes as discussed in Nowsu, Bobbie and Thuraisingham (1995). According to Karlapalem, Ahmad, So and Kwok (1997), the videos are allocated optimally by minimizing the total cost of data transfer using Genetic Algorithm, Mean Field Annealing, and Simulated Evolution algorithms. Wang and Guha (2001) proposed two data allocation algorithms: Bandwidth Weighted Partition (BWP) and Popularity Based (PB) algorithms. In BWP, the video is partitioned into unequal chunks and each chunk is allocated to one server such that the larger video chunk will be allocated into the server with the higher bandwidth and so on. In PB, the whole video with the highest popularity will be allocated to the server with the highest bandwidth. No replication process is applied in this study. Kangasharju, Roberts and Ross (2001) tried to minimize the number of traversed hops in the CDN to deliver contents efficiently using randomization, popularity, greedy single and greedy global replications. Tsang and Kwok (2000) proposed a predictive video allocation and replication algorithm based on the predictive popularity of videos.

Finally, in peer-service area architecture, the whole area is divided into a set of service areas with a cluster of servers for each. According to Li and Wu (2010), each service area contains two types of servers: Type 1 to store popular contents and Type 2 servers to store unpopular contents. Li and Wu (2010) proposed a content allocation scheme in which a certain percentage of the contents are considered popular and stored in Type 1 servers and the rest of contents are considered unpopular and stored in Type 2 servers. No replication process is applied in Li and Wu (2010).

The above literature review, shows that the most important factor in the content allocation process is how to replicate the contents such that the storage cost must be minimized and, at the sametime, the load must be distributed among the servers in a balanced manner. According to Li and Wu (2010), most of the conventional CDNs replicate the content on each server (full replication) which increases the storage cost. On the other hand, Li and Wu (2010) tried to build their content allocation process for peer-service area architecture without any replication which violates the load balancing condition and increases the request rejection rate. Therefare, we propose a balanced content allocation scheme that replicates the contents based on their popularity such that each content will be replicated and its expected load will be calculated according to their popularity.

Other unclassified works include Lee, Muntean and Smeaton, (2009), Cranor et.al., (2003), Ebara, Abe, Ikeda, Tsutsui, Sakai, Nakaniwa and Okada, (2005), Xie, Li, Wei and Cao (2007), and Feng and Lingjun (2006). Cranor et al., (2003) proposed a general framework for content allocation called the spectrum content management system which consists of three modules: content manager, policy manager, and storage manager. The content migration method called SXS is proposed by Ebara et al., (2005) which chooses the best server to move content to the target server with minimum transmission cost. Lee et al., (2009) proposed the user utility function which reflects the user viewing interests to replicate the movies with high user utility function on the users' devices for IPTV pre-recorded contents. Xie et al., (2007) proposed an efficient allocation method to deal with addition, deletion of nodes using the tiger hash function and mapping methods. In Feng and Lingjun (2006), object placement adjustment with replication is proposed to minimize block probability, in which the object moves from high traffic intensity storage servers to lower traffic intensity servers.

Proposed Content Allocation scheme

Content allocation scheme for Video on Demand Systems should balance the workload among video servers while considering the server capacity constraints. In real VoD systems, the user request is highly skewed such that the new contents are popular and get most of the user requests while the old contents are unpopular and get only a few user requests. Thus, this characteristic is widely exploited to design the popularity-based content allocation schemes. In peer-service area architecture (Li & Wu, 2010), the contents are allocated into two types of servers: popular contents are stored in Type1 servers, and unpopular contents are stored in Type 2 servers. No replication process is applied, so although the storage is efficient, the load imbalance problem will arise due to the load of the contents not being considered during the allocation process. In this paper, the balanced content allocation scheme is proposed that replicates the contents based on their popularities. Our scheme differs from others in its ability to replicate the contents based on their popularity such that the number of replicas of any content depends on the degree of popularity. After that, the load of each video will be calculated and considered during the allocation process to maintain the load balanced within the service area. Before demonstrating the proposed solution, we have to explain the used terms and equations.

L_i denotes the maximum number of requests per unit time that may be serviced by video *i* simultaneously and can be computed by multiplying the number of subscribers connected with server *j* by request rate per user by popularity of the requested video. It can be written mathematically as the following:

$$L_i=\lceil \frac{U}{S} * \lambda * p_{ij} \rceil \quad (1)$$

where U/S represents the expected users who can access the server j, represents the request rate per user within unit time, and p_{ij} represents the popularity of video i that is stored in server j. Note that the expected load of server j can be expressed by summing the load for all videos that are stored in this server.

After computing the expected load of the content, the number of replicas that must be allocated for each video must be controlled by multiplying two factors: the popularity of the video pi $\in$]0,1] and the available servers S for example, if p_i=1 (very popular video) and S=4 then video i will be allocated on all servers. But if p_i=0.5 then video i will be allocated on only two servers. This can be formulted mathematically as:

$$R_i=[S * p_i] \quad (2)$$

where C_i represents the expected number of replicas for video i, and S represents the number of available servers. After computing the expected number of replicas for video i, we can now calculate the expected load for each replica by dividing the expected load for video i by the number of replicas.

In our proposed content allocation scheme, there is a set of contents C= {c_1, c_2, c_3, …, c_m,} with their corresponding workload L_c={l_1, l_2, l_3, …, $l_{m,}$} computed according to the equation (1), and also corresponding popularities P={p_1, p_2, p_3, …, p_m,} where and the popularities are sorted as $p_1 > p_2 > p_3 > \cdots > p_m$. There is also a set of servers S = {s_1, s_2, s_3, …, s_n,} with corresponding workload L_s = {l_1, l_2, l_3, …, l_n} initialized to the sum of workload of already-stored contents (zero if empty).

The main idea of our scheme is to sort the contents according to their popularities in decreasing order, and based on the predefined partitioning threshold t, partition the contents list into two sub-lists: popular sub-list $pl = c_1$, c_2, c_3, …, c_t} and unpopular sub-list = {c_{t+1}, c_{t+2}, c_{t+3}, …, c_m}. The contents will be allocated to servers as follows: the unpopular contents in *ul* are allocated to servers using Round Robin Algorithm, and then the popular contents will be replicated to x versions according to equation (2) as follows: rep_i={rep_{ir}:$r \leq x_i$}, and based on the current workload of the servers, each replica will be allocated into the least loaded server. The workload of any $server_j$ has to be updated after allocating a replica of content i by adding workload l_{ri} as follows: $l_j=l_j+l_{ri}$ where $l_{ri}=l_i/x_i$.

This proposed scheme tries to maintain the load of all servers balanced to solve the problem of load imbalance. Unlike the scheme of Li and Wu (2010) which didn't replicate the contents, our scheme replicates the contents based on their popularity to satisfy the trade-off between storage cost and load balance.

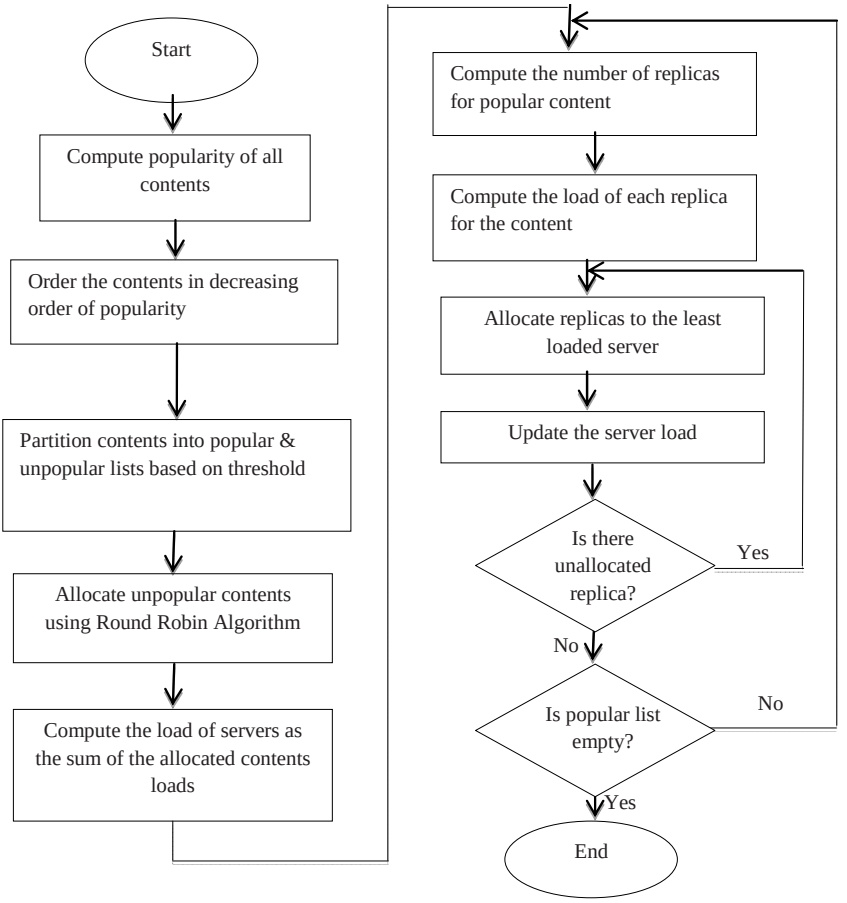

Figure 2. Flowchart of proposed balanced content allocation scheme.

Experimental Result

In this section, we demonstrate the superiority of the proposed content allocation scheme in the modified architecture from the perspective of load balancing. We compare our scheme with the scheme proposed by Li and Wu (2010). The experimental test is done by simulating the proposed content allocation using our own simple simulation under Visual C++ environment.

For the sake of comparison between the two schemes to prove the superiority of the proposed scheme, we evaluate the workload on each server to show the ability of our scheme to maintain the load balance among all servers.

To evaluate the performance of the two schemes, we used the concurrent requests at each server as a metric of workload to decide the violations of load balance conditions. In other words, if the concurrent requests at all the servers are equal/ almost equal then the scheme satisfies the load balance conditions. Otherwise, the scheme violates that condition.

For simplicity, there are a set of assumptions that are considered: S=4 servers, U=10000 users/area. Finally $\lambda = 0.01$ request/user/second. These assumptions can be changed as needed. For the purpose of comparison, we set the value of threshold to be 60% according to the assumption of Li and Wu (2010). Hereafter, we will refer to the proposed scheme by "our" and to the scheme of Li and Wu (2010) by "Li & Wu".

From Fig. 3 we can notice that in Li and Wu's (2010) scheme, the workload at server1 is 50 concurrent requests while the workload at server 4 is around 10 concurrent requests. Server 2 and server 3 are around 30 concurrent requests. In our scheme, the workload at all the four servers is equal (around 30 concurrent requests).

This can be interpreted as the following: the Li and Wu (2010) scheme stores the unpopular movies in server 4 (Type 2 server) and the popular movies are distributed among the other servers without any consideration for the load of those servers; so this model wastes the resources of some servers without benefit from them that causes over-utilizing the other servers. This scheme may lead to user request rejection at server 1 if it exceeds its maximum load at any time. As depicted in Fig. 3, our proposed scheme distributes the load among the servers evenly due to the proposed scheme allocating the contents according to load of the content and the current load of servers.

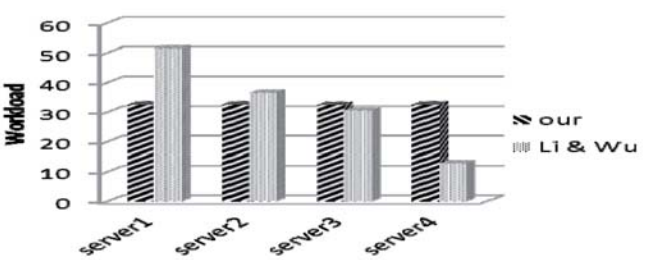

Figure 3. Comparison of Load between the two schemes.

The effects of changing the threshold values as a parameter that determines the size of popular and unpopular contents are examined in order to study the effects of separating popular/unpopular contents on the load distribution for both schemes as depicted in Figs. 4 and 5. We fixed the number of contents to be 100 movies and then we carried out both schemes on the following threshold values % 10%, 30%, 60%, and 90%. Fig. 4 shows the current load of servers in the Li and Wu (2010) scheme. In this figure we can see clearly that the load of server 4 behaves unlike other servers such that when the threshold value is small (10%) then the load of server 4 becomes high (around 55 concurrent requests) while the load is degraded when the threshold value is increased. So, server 4 is affected by threshold value significantly because the threshold value determines the workload of unpopular contents that must be stored in server 4 that is interpreted as follows: when the threshold value decreases, the number of contents that will be stored in server 4 increases which leads to increase of the load on server 4. As shown in Fig. 4 also, the other servers suffer from the same problem of load imbalance. In Fig. 5, we can note that our proposed model not affected by the change of this threshold where the proposed model is retains its stability and the load of both servers is approximately equal with a slight variation and we think it is negligible. This proves that the proposed model is scalable and efficient under any variation in contents classification (i.e. the variation in size of popular/unpopular contents).

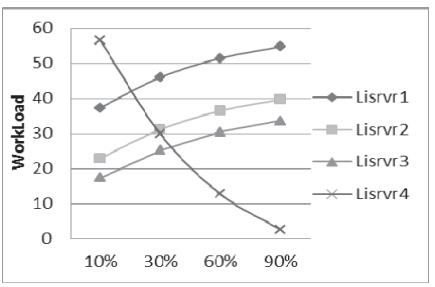

Figure 4. Threshold effect on Li and Wu scheme.

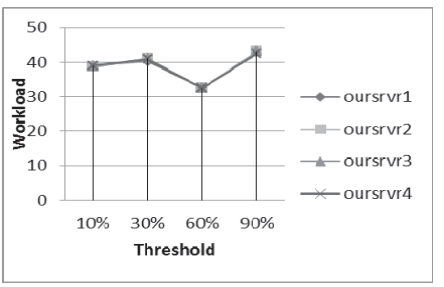

Figure 5. Threshold effect on our scheme.

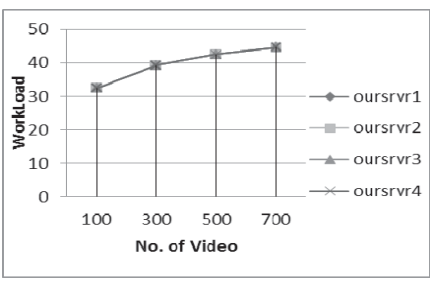

Figure 6. Effect of content size on our scheme.

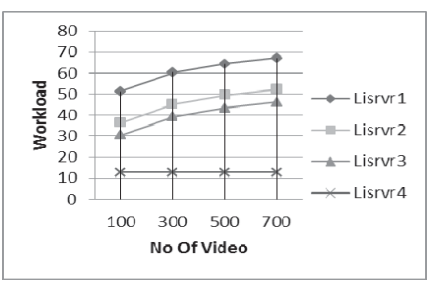

Figure 7. Effect of content size on Li and Wu scheme.

Figs. 6 and 7 show the effect of content size on the workload for both schemes. From Fig. 6, we can notice for our scheme, that the workload of all the servers increases when the content size increases (100, 300, 500, and 700 contents) and also the workload among all servers is equal as depicted in Fig. 6. This is to support our claim that our proposed content allocation scheme is balanced and maintains the load balance among all servers. Unlike our scheme, the Li and Wu (2010) scheme violates the load balance condition as depicted in Fig. 7 where the workload of all servers varies. It is important to notice that the workload at server 4 didn't change with the increase of content size because it is dedicated to storing the unpopular contents only, which are requested very rarely. Fig. 6 and Fig. 7 demonstrate that content size is not a critical factor in the content allocation process which supports the findings of Li and Wu (2010), but is considered a constraint when the server capacity is limited.

CONCLUSION AND FUTURE WORK

The load imbalance problem in peer-service area architecure of CDN for Video on Demand (VoD) services in IPTV is studied in this paper. The Balanced Content Allocation scheme that replicates the contents based on the expected load, popularity, and the load of servers is also proposed. The distribution of load among servers, the effects of threshold value, and the content size are examined. The experimental results demonstrate the superiority of the proposed scheme over the generic scheme of Li and Wu (2010) from the perspective of load balancing.

Many issues are still unsolved including the efficient distribution of users' requests among servers, extending the cost-effective peer-service area architecture to include the load balance factor into consideration to optimize the number of servers and allocated contents dynamically to achieve a balanced and cost-effective architecture.

REFERENCES

Bisdikian, C., & Patel, B. (1996). Cost-based program allocation for distributed multimedia-on-demand systems. *IEEE Multimedia, 3*(3), 62-72. doi:10.1109/ 93.556540

Cheng, J. (2007). *Report: One-third of TV watching to be video-on-demand by 2012*. Report. Retrieved from http://arstechnica.com/old/content/2007/09/ report-one-third-of-tv-watching-to-be-video-on-demand-by-2012.ars

Cholvi, V., & Segarra, J. (2008). Analysis and placement of storage capacity in large distributed video servers. *Computer Communication, 31*(15), 3604-3612. doi:10.1016/j.comcom.2008.06.012.

Cidon, I., & Kutten, S., & Soffer, R. (2001). Optimal allocation of electronic content. (INFOCOM'01), *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Vol 3. Proceedings* IEEE. (pp. 1773-1780). doi:10.1109/INFCOM.2001.916675.

Cranor, C., Ethington, R., Sehgal, A., Shur D., & Sreenan, C. (2003). Design and implementation of a distributed content management system. (NOSSDAV '03). 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video. ACM, USA, (pp.4-11). doi:10.1145/ 776322.776326.

Doverspike, R., Li, G., Oikonomou, K., Ramakrishnan, K., Sinha, R., Wang, D., & Chase, C. (2009). Designing a reliable IPTV network. *IEEE Internet Computing, 13*(3), 15-22. doi:10.1109/MIC.2009.58

Ebara, H., Abe, Y., Ikeda, D., Tsutsui, T., Sakai, K., Nakaniwa, A., & Okada H. (2005). A cost effective dynamic content migration method in CDNs. *IEICE Transaction on Communication.* E88-B (12), 4598–4604. doi:10.1093/ietcom

Feng, D., & Lingjun, Q. (2006). Adaptive object placement in object-based storage systems with minimal blocking probability. (AINA '06), 20th International Conference on Advanced Information Networking and Applications. Vol.1 (pp.611-616). IEEE Computer Society, Washington, DC, USA. doi: 10.1109/ AINA.2006.73

Gu, J., & Nah, J. (2008). Key management for overlay-based IPTV content delivery. *International Journal of Computer Science and Network Security, 8*(12), 161-167. Retrieved from http://paper.ijcsns.org/07_ book/ 200812/20081223.pdf

Gupta, R. (2011). Global IPTV market forecast to 2014. *Market Research Reports*. Retrieved from http://businessresearc.wordpress. com/2011/02/24/ global-iptv-market-forecast-to-2014

Kangasharju, J., Roberts, J., & Ross, K. (2002). Object replication strategies in content distribution networks. *Computer Communications, 25*(3), 367-383. doi=10.1.1.12.1793

Karlapalem, K., Ahmad, I., So, S., & Kwok, Y. (1997). Empirical evaluation of data allocation algorithms for distributed multimedia database systems. (COMPSAC '97), The Twenty-*First Annual International Computer Software and Applications Conference* (pp. 296-301). Washington DC. USA. doi:10.1109/CMPSAC.1997.624842

Kim, C., Bak, Y., Woo, S., Lee, W., Min, O., & Kim, H. (2006). Design and implementation of a storage management method for content distribution. (ICACT'06). *The 8th International Conference on Advanced Communication Technology, Vol. 2* (pp.1147-1151). Phoenix Park. doi: 10.1109/ICACT. 2006.206173

Krogfoss, B., Sofman, L., & Agrawal, A. (2008). Caching architecture and optimization strategies for IPTV networks. *Bell Lab Tech. Journal, 13*(3), 13-28. doi:10.1002/bltj.20320

Laoutaris, N., Zissimopoulos, V., & Stavrakakis, I. (2005). On the optimization of storage capacity allocation for content distribution. *The International Journal of Computer and Computer Network, 47*(3), 409-428. doi:10.1016/j.comnet. 2004.07.020

Lee, S., Muntean, G., & Smeaton, A. (2009). Performance-aware replication of distributed pre-recorded IPTV content. *IEEE Transaction on Broadcasting, 55*(2), 516-526. doi: 10.1109/tbc.2009.2015985

Li, M., & Wu, C. (2010). A cost-effective resource allocation and management scheme for content networks supporting IPTV sevices. *Journal of Computer Communication, 33*(1), 83-91.doi: 10.1016/j.comcom.2009.08.003

Lin, Y., Lai, H., & Lai, Y. (1996). A hierarchical network storage architecture for video-on-demand services. 21st Proc. (LCN '96), 21st Annual *IEEE Conference on Local Computer Networks,* (pp.355-364). IEEE Computer Society, Washington, DC, USA. doi: 10.1109/LCN.1996.558164.

Mandal, S., & MBuru M. (2008). *Intelligent pre-fetching to reduce channel switching delay in IPTV systems*. Department Technical Report, Texas A & M University. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary? doi=10.1.1.114.8467

Nakaniwa, A., Ebara, H. (2007). Optimal allocation of cache servers and content files in content distribution networks. (IMSA'07), *European*

Conference on Proceedings of the IASTED European Conference: Internet and Multimedia Systems and Applications, (pp. 15-22). ACTA Press, Anaheim, CA, USA.

Nwosu, K., & Bobbie, P., & Thuraisingham, B. (1995). Data allocation and spatio-temporal implications for video-on-demand systems. *Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on Computers and Communications*. (pp.629-635). Scottsdale, AZ , USA. doi: 10.1109/PCCC.1995.472427

Pfeffer, P. (2006). IPTV: Technology and development predictions. *Fiber & Integrated Optics, 25*(5), 325-346. doi:10.1080/01468030600816979.

Plagemann, T., Goebel, V., Mauthe, A., Mathy, L., Turletti, T., & Urvoy-Keller, G. (2006). From content distribution networks to content networks – issues and challenges. *Journal of Computer Communications, 29*(5), 551-562. doi: 10.1016/j.comcom.2005.06.006

Ryn Choe, Y., & Pai, V. (2007). Achieving reliable parallel performance in a VoD storage server using randomization and replication. (IPDPS '07), *IEEE International Parallel and Distributed Processing Symposium,* (pp.1-10). Long Beach, CA. doi: 10.1109/IPDPS.2007.370220

Scheuermann, P., Weikum, G., & Zabback, P. (1998). Data partitioning and load balancing in parallel disk systems. *VLDB Journal, 7*(1), 48-66. doi=10.1007/s007780050053.

Song, B., Hassan, M., & Huh, E. (2011), An IPTV service delivery model using novel virtual network topology (ICUIMC '11). *The 5th International Conference on Ubiquitous Information Management and Communication*. ACM, New York, NY, USA, Article 8. doi=10.1145/1968613. 1968623

Tang, K., Ko, K., Chan, S., & Wong, E. (2001). Optimal file placement in VOD system using genetic algorithm. IEEE Transaction. *Industrial Electronics, 48*(5), 891-897. doi: 10.1109/41.954552

Tang, W., Wong, E., Chan, S., & Ko, K. (2004).Optimal video placement scheme for batching VOD services. *IEEE Transaction on Broadcasting, 50*(1), 16–25. doi: 10.1109/TBC.2003.822983

Tsang, K., & Kwok, S. (2000). Video management in commercial distributed video on demand (VoD) systems. (PACIS'00), *Proceedings of the Pacific*

Asia Conference on Information Systems, (pp.214-224). Retrieved from http://aisel.aisnet.org/pacis2000/17.

Tsao, S., Chen, M., Ko, M., Ho, J., & Huang, Y. (1999). Data allocation and dynamic load balancing for distributed video storage server. *Journal of Visual Comm.& Image Rep. 10*(2), 197-218. doi: 10.1006/jvci.1999.0420

Wang J., & Guha R. (2001). Efficiently allocating video data in distributed multimedia applications. *Journal of Applied System Studies: Methodologies and Applications for Systems Approaches, Special issue on Distributed Multimedia System with Applications.* doi=10.1.1.93.9351

Wang, Y., Liu, J., Du, D., & Hsieh, J. (1997). Efficient video file allocation schemes for video-on-demand services. *Multimedia Systems, 5*(5), 283-296. doi:10.1007/s005300050061

Xie, C., Li, X., Wei, Q., & Cao, Q. (2007). EOP: an efficient object placement and location algorithm for OBS cluster. (ICA3PP'07), *The 7th International Conference on Algorithms and Architectures for Parallel Processing*, (pp. 222-230). Springer-Verlag, Berlin, Heidelberg.

Yarali, A., & Cherry, A. (2005). Internet Protocol Television (IPTV). (*TENCON 05), IEEE Region10*, (pp. 1-6). doi: 10.1109/TENCON.2005.300861