Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

An Adaptive Cloud Downloading Services

Vikas Mishra [B80544240][1],Lalit Patil [B80544247][2],Omkar Shinde [B80544261][3] ,Namrata [B80544212][4], Mrs.Deeksha Bhardwaj[5]

[1,2,3,4,5]*G . H . Raisoni Institute of Engineering and Technology*

Abstract - Video content downloading using the P2P approach is scalable, but does not always give good performance. In this service, the cloud storage and server caches user interested content, and updates the cache based on user downloading requests. We design an adaptive algorithm (AMS) to select the service mode automatically. Intuitively, AMS switches service mode from server mode to helper mode when too many peers request for blocked movies, and vice versa.
Keywords-cloud server, peer-to-peer, helper, file downloading, video.

I. INTRODUCTION

Video content distribution is a challenging research problem because of its high bandwidth requirement and the fast growing video population. In recent years, it is reported that Internet traffic is already dominated by video. In a P2P content distribution system, peers who create demand for videos also share their content with other peers.

The service capacity thus increases automatically with increasing peer population, making scalability an advantage of the P2P solution. Peers will get support from a server if they cannot obtain enough content for streaming from other peers.

In file sharing scenarios, however, dedicated server is not commonly deployed for service capacity. Peers requesting unpopular videos often suffer low downloading rate. To remedy this, a *cloud downloading* service is deployed in P2P video downloading systems to enhance the performance of downloading. The mechanisms to distribute video content include CDN (Cloud Download Network) and Peer-to-Peer (P2P). CDN is a traditional solution based on deploying servers at the edge of the network, near video access points. Scalability is a limitation of CDN because the server capacity becomes a bottleneck when there are a large number of concurrent peer requests.

Later P2P becomes a popular solution, and is adopted by many content distribution systems. In a P2P content distribution system, peers who create demand for videos also share their content with other peers. A cloud storage system is used to cache a large fraction of video content, and high bandwidth is provided to access this cache. Peers can get a big performance boost by connecting to cloud downloading.

There are two generic service modes for cloud servers. In the first mode, the cloud server is primarily focused on serving the content already cached at the cloud storage system. Requests for content not in the caches are blocked until such content becomes cached. The cloud storage system updates its cache periodically to replace content without requests by content with requests waiting. We call this the *server mode*.

II. LITERATURE SURVEY

1. Existing System:

CDN is a traditional solution based on deploying servers at the edge of the network, near video access points. Scalability is a limitation of CDN because the server capacity becomes a bottleneck when there is a large number of a concurrent peer request.

In file sharing scenarios, however, dedicated server is not commonly deployed for service capacity. Peers requesting unpopular videos often suffer low downloading rate. To remedy this, a *cloud downloading* service is deployed in P2P video downloading systems to enhance the

performance of downloading. A cloud storage system is used to cache a large fraction of video content, and high bandwidth is provided to access this cache.

Peers can get a big performance boost by connecting to cloud downloading. The architecture and implementation of such a cloud downloading system is introduced.

Disadvantages of Existing System:

a. Video content distribution is a challenging research problem because of its high bandwidth requirement and the fast growing video population. In recent years, it is reported that Internet traffic is already dominated by video.

b. In file sharing scenarios, however, dedicated server is not commonly deployed for service capacity. Peers requesting unpopular videos often suffer low downloading rate.

c. High bandwidth is required in case of file having large size to download.

2. Proposed System:

There are two generic service modes for cloud servers. In the first mode, the cloud server is primarily focused on serving the content already cached at the cloud storage system. Requests for content not in the caches are blocked until such content becomes cached.

The cloud storage system updates its cache periodically to replace content without requests by content with requests waiting. We call this the *server mode*.

An alternative mode is the *helper mode*, in which the cloud server does not block any requests. For videos that are not cached, the cloud servers simply relay chunks from some peers to other peers, acting as a helper peer. One contribution of our study is to compare these two modes analytically.

The results are interesting, in the sense that both modes can be advantageous for some operating regimes - the server mode when video population is large compared to cache size, and the helper mode when peer request rate is high compared to server bandwidth. We integrate these two modes into a single adaptive cloud downloading service

Advantages of Proposed System:

a. The benefit is that more peers can contribute their upload capacity by switching their state from waiting to downloading.

b. Server mode is most efficient for dealing with large video population relative to the cache size.

c. Peer doesn't have to wait for long time, as he get connected to helper peer as if contents are not serve by server.

d. Bandwidth required is moderately low as compare to existing traditional System.

III. STATIC MODEL

We begin analysis with static model. Let N denote the number of peers, hence downloading requests; by *static*, we mean the number requests remains unchanged at N.

Given the full mesh, and uplink limiting assumptions, the peers, server and cloud server can be thought forming a logical star network as shown in Fig. 1. Here, we add the analysis with the addition of helpers. Helpers are not interested in any movie and just help amplify the system capacity to improve user experience. In this work, the helper is a cloud server with storage to cache some movies to enhance service capacity. Given cloud servers as helpers, there are two intuitive strategies to serve each downloading request.

- Helper Mode: The cloud server begins to help the downloading request without caching the video in its cloud storage. The cloud server downloads video chunks from the P2P system and redistributes it to other peers who are without these chunks. Then, these chunks are discarded.

- Server Mode: The downloading requests are not served until the requested video is cached by the cloud storage. The requests for videos not in the cache are blocked.

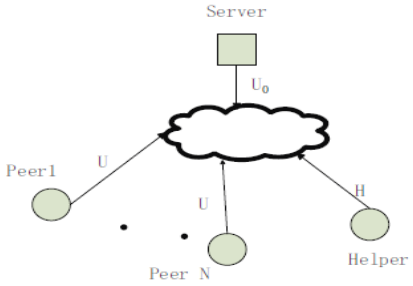


Fig. 1

The cloud storage is updated periodically in order to serve blocked requests.

Now, we analyze the static case by considering a single movie. The performance objective of minimizing downloading time is equivalent to maximizing throughput rate. The maximum throughput is also referred to as the capacity of the P2P system, denoted *C*, and has been derived by previous studies.

$$C = min\{U_0, U + \frac{U_0}{N}\}. \quad ..(1)$$

Intuitively speaking, there are two possible bottlenecks. One is the capacity of source, i.e. server. The other one is the total capacity. By constructing a set of 1-hop and 2-hop spanning trees and derives the maximum throughput and finds a centralized scheduling strategy to achieve the maximum throughput.

If the cloud server adopts helper mode, the cloud server needs to download video content from P2P or the source, which is a waste of bandwidth compared with server mode. To maximize the throughput, the cloud server should distribute any particular chunk to all peers. Thus, the efficiency to utilize cloud server's upload capacity is $\frac{N-1}{N}$.

1. **Theorem *1*:** The maximum throughput in helper mode is:

$$C = min\{U_0, U + \frac{U_0 + \frac{N-1}{N}H}{N}\} \quad ..(2)$$

The server mode case is a straightforward extension of the basic single-file downloading result in Eq. 1:

2. **Theorem *2*:** If the cloud server already cached this video, the maximum throughput is:

$$C = min\{U_0 + H, U + \frac{U_0 + H}{N}\} \quad ..(3)$$

In the static model, the server mode is always better than the helper mode with the cost of additional storage to cache the particular video. Since we only consider a single video in the static model, the storage cost can be ignored. However, in practical systems, the cloud server is unable to cache all videos. In that case the helper mode may surpass the server model.

IV. DYNAMIC MODEL

In this section, we extend our discussion to a multi-video system. The peer downloading requests arrive as a Poisson process. Once downloading is complete, a peer leaves the system immediately. For benchmarking, we also analyze the unlimited-cache case, in addition to the cases of helper mode and server mode. In the unlimited-cache case, we assume all videos are stored in the cloud storage.

A. *The Unlimited-cache Case*

In this case we assume the cloud storage is able to store all videos, i.e. $K = M$.

For video j, the arrival rate of requests is $\lambda \times \eta j$. We can write the following equation for each video:

$$\frac{dN_j}{dt} = \lambda \times \eta_j - \frac{\beta_j N_j U + H \times \frac{N_j}{N}}{F}, j = 1, \ldots, M,$$
$$s.t. \quad \sum_{j=1}^{M} N_j = N, \quad \sum_{j=1}^{M} \eta_j = 1. \qquad ..(4)$$

β_j is the effectiveness of file sharing, which is the efficiency for peers to help each other download content.

$\beta_j \approx 1$ when the video can be divided into many small chunks. For simplicity,

let $\beta_j = 1$ in our analysis. For EQ. 4, on the right hand side, the first term is the arrival rate of requests and the second term is the departure rate. The departure rate is determined by the system capacity.

B. *Helper Mode*

It is reasonable to consider the regime that only a fraction of all the videos are stored in the cloud. In helper mode, the cloud server serves all downloading requests whether the video is cached or not. If the video is not cached, the cloud server will relay and amplify the video content downloaded from other peers. The cloud server has no interest in replicating any video.

As implied by EQ. 2, about $\frac{1}{N_j}$ of the cloud server's bandwidth (allocated to video j) will be wasted.

Although some paper claims that full helper bandwidth utilization can be achieved using distributed P2P scheduling, they still require centralized coordination. Then, we analyze the average downloading time based on distributed scheduling.

C. *Server Mode*

In server mode, any request for a video not cached is blocked until the cloud storage update. Therefore, it is necessary to differentiate peers as *downloading peers* and *waiting peers*.

The hit rate, denoted by γ, is the probability that the requested video is cached by the cloud storage. The average time interval between updates of cloud storage is denoted by 1/μ. A new video is brought into the cloud storage by removing a video without requests. The population of waiting peers and downloading peers are denoted by N_W and N_D respectively. Usually, however, the waiting time does matter and must be considered. In the general case, it is complicated to analyze the waiting time. We study the worst case in this work.

V. ADAPTIVE ALGORITHM

From the above analysis, there are both strengths and drawbacks for both the helper mode and server mode. The helper mode wastes P2P resource because the cloud server needs to keep downloading new content to help peers; while the server mode wastes the bandwidth resource of blocked peers. In this section, we design an adaptive algorithm to determine the service mode for

each movie. The cloud server adjusts its strategy periodically, by running the following Automatic Mode Selection (AMS) algorithm to determine the mode for each movie.

We assume the value of N' is known. The movies in helper mode have higher priority to be included into cloud storage. Then, we consider the other movies in the order of decreasing peer population.

Automatic Mode Selection (AMS) Algorithm:

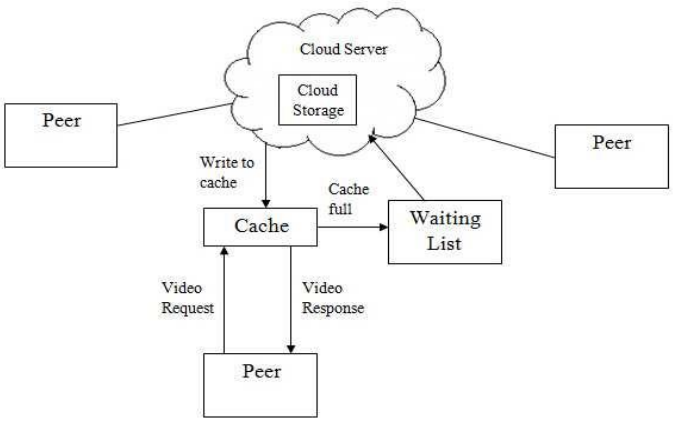
1: for each movie j not in K. do
2: if The active movie is less than K then,
3: Update cloud storage to add movie j by replacing any movie without request.
4: $N' = N' + N_j$
5: else
6: if $(H/(N' + \alpha_j N_j)) < N_j U$ then
7: Use Helper Mode for movie j.
8: $N' = N' + \alpha_j N_j$
9: else
10: keep blocking peers requesting for movie j
11: end if
12: end if
13: end for

Based on our analysis, the weakness of helper mode is the additional bandwidth cost to download the requested video by cloud server. The benefit is that more peers can contribute their upload capacity by switching their state from waiting to downloading. Thus Alg. 1 compares the cost and the benefit and start helper mode once the benefit is larger than cost.

Adaptive Cloud Downloading Architecture(Main architecture):-

In order to implement Cloud Downloading we need :

a) Helper Mode
b) Server Mode



VI. MATHEMATICAL MODEL

Let 'Cd' be the Cloud Downloading System whereas,
Cd= {U, L, Se, He ,Ad, Q,M,Um} where,
U = {u1, u2, u3, …., un| 'U' is the set of Users}
L = {L1 |'L1' is the Login}
Ad = {ad1|'Ad1' is the Administrator}
Se = {Se |'Se' is the Server mode}

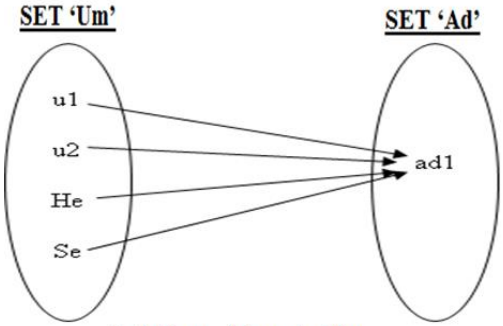
He = {He| 'He' is the Helper mode}
Q = {q1, q2, q3, …., qn| 'Q' is the set of Queries}
M= {Se, He| 'M' is the Modes to Select}
Um= {u1, u2….un, Se, He| 'Um' is the Users and Modes}

A. Activities:

- **Activity A1: Connection**
  First, the connection is established between User, Server and Helper.
- **Activity A2: Register**
  Administrator will register to a System.
- **Activity A3: Entry**
  Enter the Username and password.
- **Activity A4: Login**
  Check whether details are correct or not in the system.
- **Activity A5: Request for a File**
  For downloading a File sends the request to a Server.
- **Activity A6: Downloading**
  Download the file from Server Mode or From Helper Mode.
- **Activity A7: Uploading**
  Upload the file to Server Mode.
- **Activity A8: Asking for cache Store**
  If downloading is completed, then it gets stored in cache of server

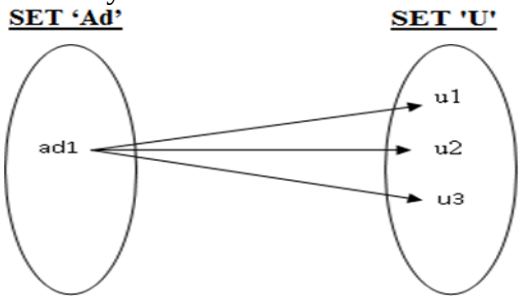
B. Venn Diagrams for Events:

- **Activity A1: Connection**
  First, the connection is established between User, Server and Helper.



Relation – Many to One

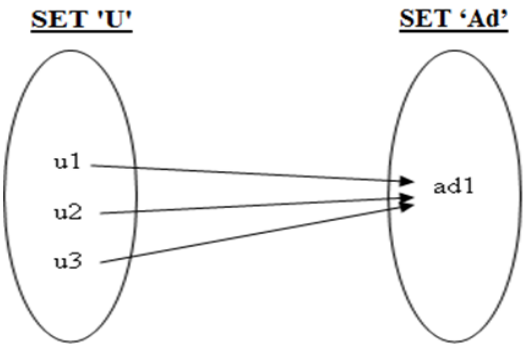
- **Activity A2: Register**
  Administrator will register to a System.



Relation – One to Many

- **Activity A3: Entry**

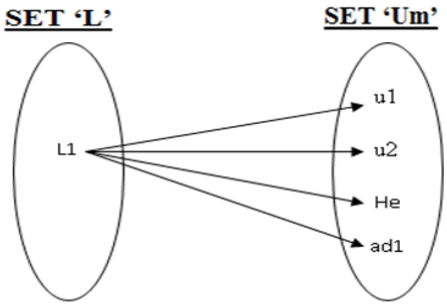
Enter the Username and password.



Relation – Many to One

Activity A4: Login

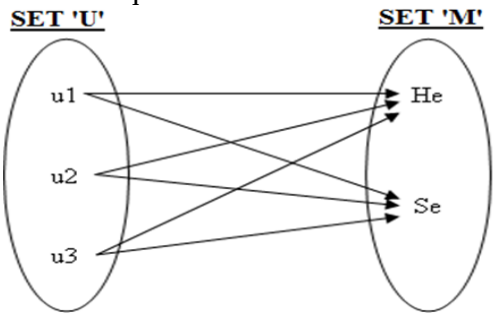
Check whether details are correct or not in the system.



Relation – One to Many

- **Activity A5: Request for a File**

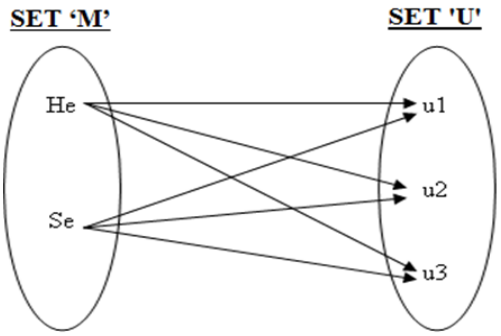
For downloading a File sends the request to a Server.



Relation – Many to Many

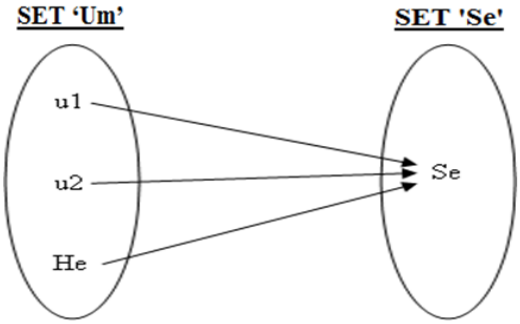
- **Activity A6: Downloading**

Download the file from Server Mode or From Helper Mode.



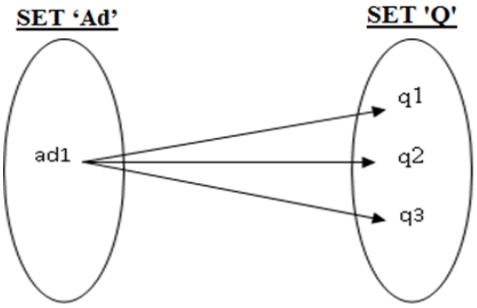
Relation – Many to Many

- **Activity A7: Uploading**
  Upload the file to Server Mode.



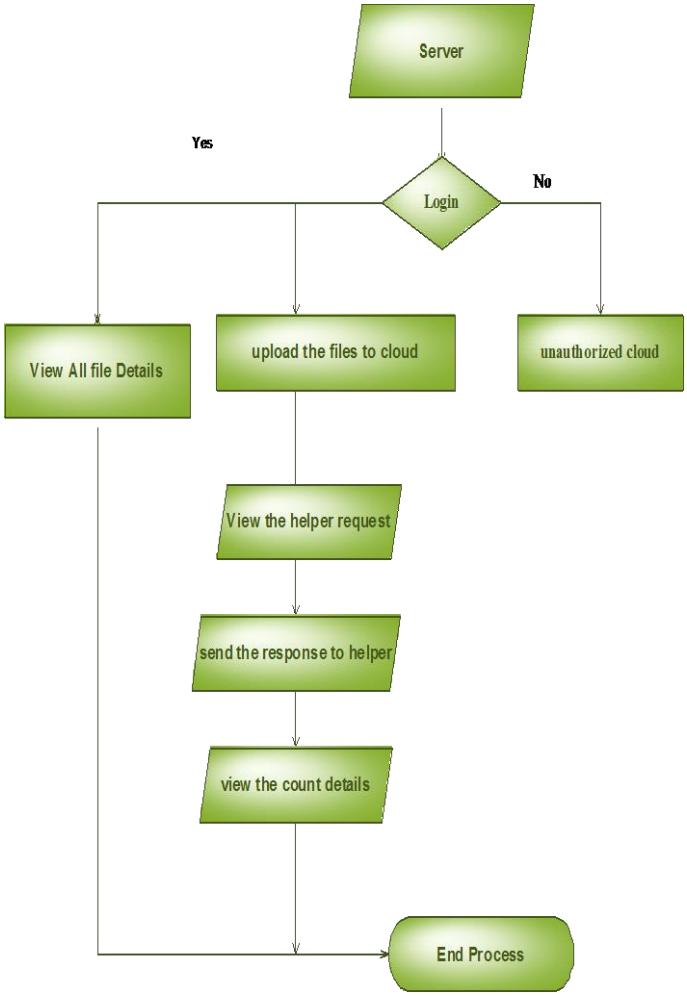
Relation – Many to One

- **Activity A8: Asking for cache Store**
  If downloading is completed, then it gets stored in cache of server.

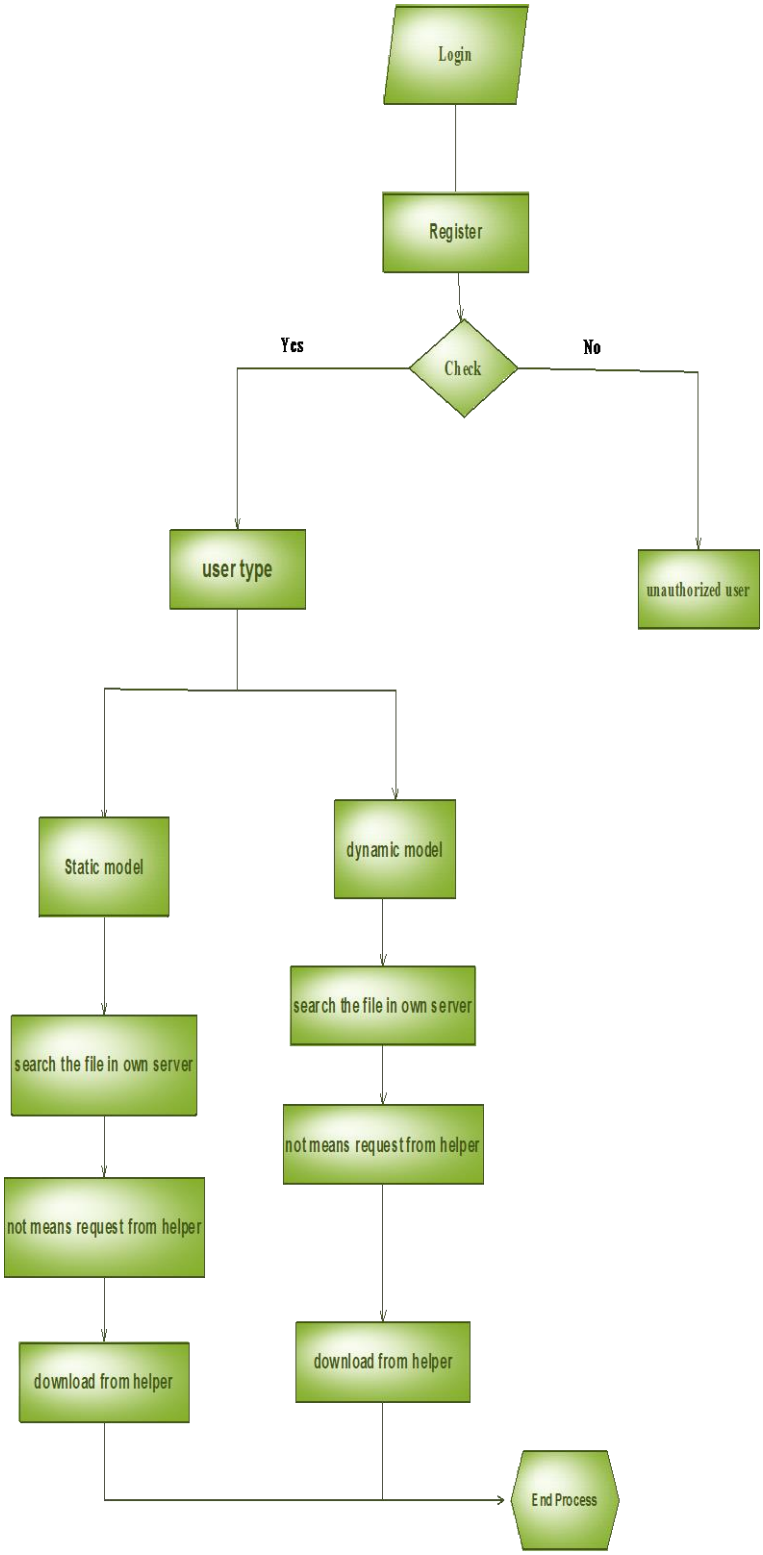


Relation – One to Many

VII. System Design (Admin):

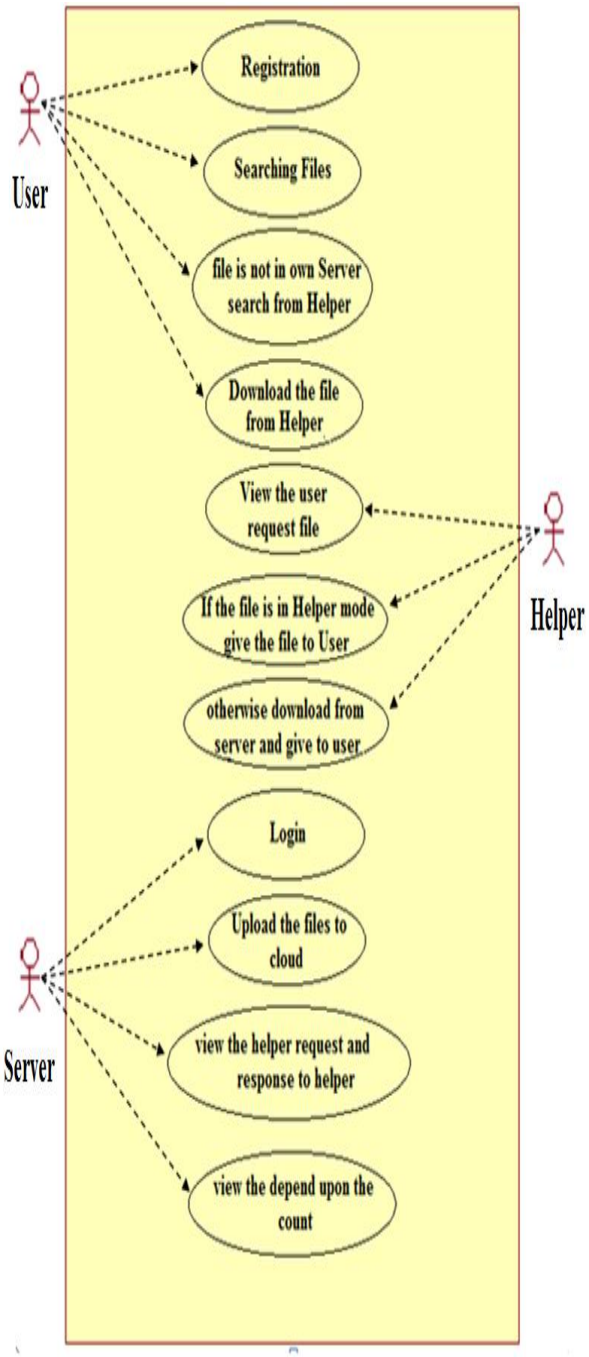


System Design (User):

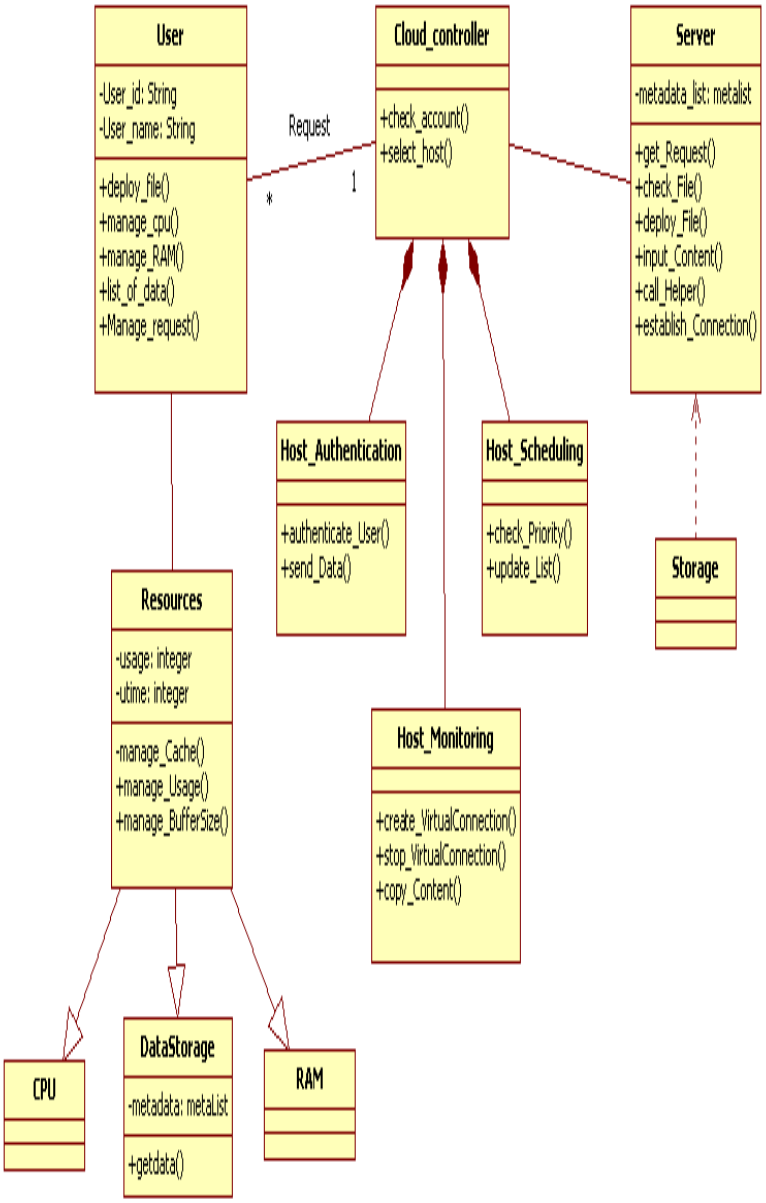


VIII. UML Diagrams

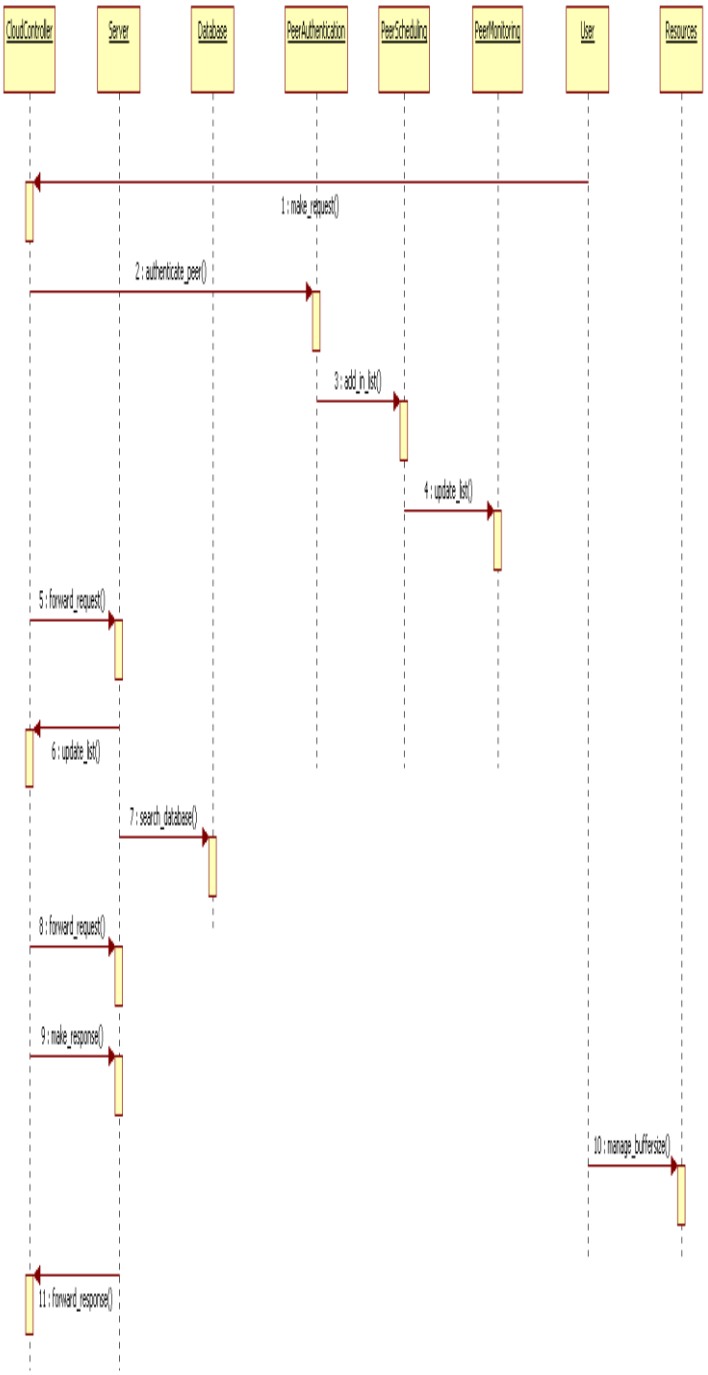
A. Use Case Diagram



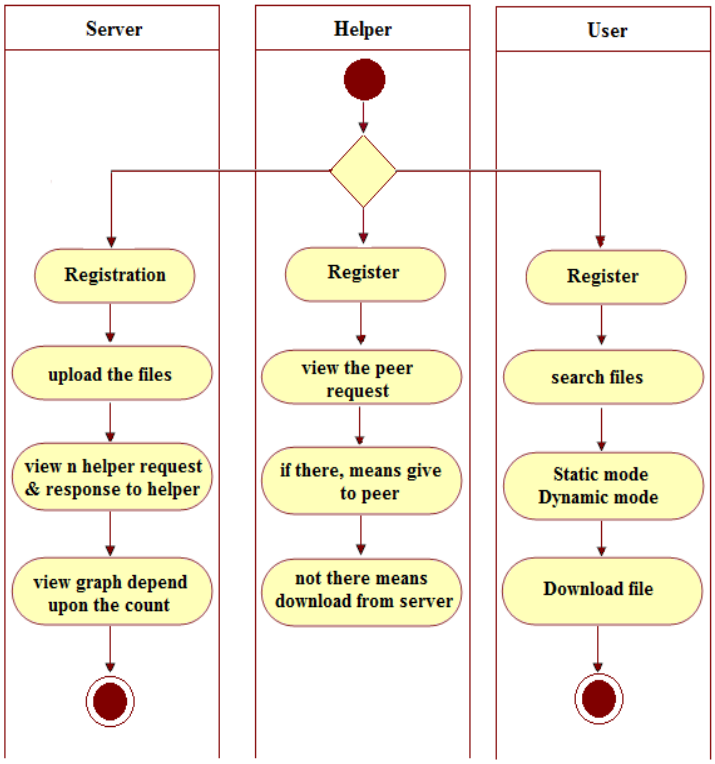
B. Class Diagram



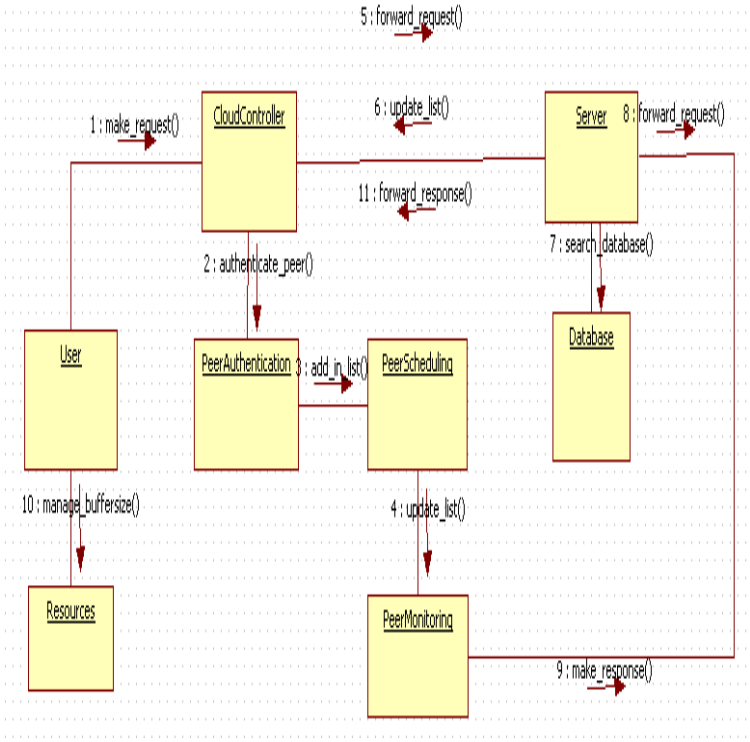
C. Sequence Diagram



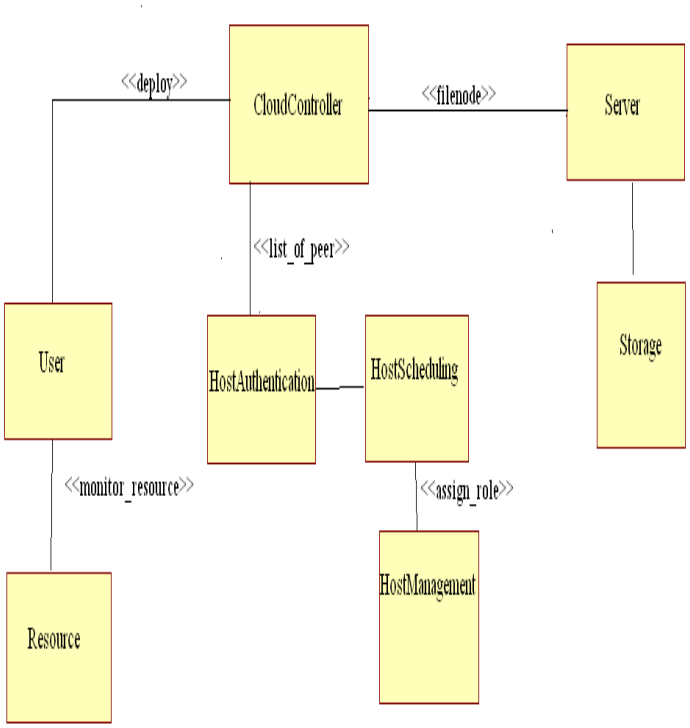
D. Activity Diagram



E. Collaboration Diagram



F. Object Diagram



IX. CONCLUSION

In this work, we build a theoretical model to analyze different strategies for a cloud downloading system. In particular, helper mode and server mode are used as abstraction of two different design philosophies - using the cloud as peer or as server. Our analysis reveals that each strategy can be advantages, for certain operating scenarios. Helper mode wastes some server bandwidth, but is best at leveraging P2P capacity when request load is high.

On the other hand, server mode is most efficient for dealing with large video population relative to the cache size. We design an automatic mode selection (AMS) algorithm to choose the suitable service mode for different scenarios. Our analysis helps a cloud downloading system to optimize its design. We also discuss the potential benefit to apply our result for the mobile P2P case.

REFERENCES

[1] Yipeng Zhou, *Member, IEEE,* Tom Z. J. Fu, Dah Ming Chiu, *Fellow, IEEE,* and Yan Huang, "An Adaptive Cloud Downloading Service", **IEEE TRANSACTIONS ON MULTIMEDIA, 2013.**

[2] H.-C. Hsiao and C.-T. King, "Bristle: a mobile structured peer-to-peer architecture," in *Proc. of Parallel and Distributed Processing Symposium*, 2013.

[3] Niu, C. Feng, and B. Li. Pricing Cloud Bandwidth Reservations under Demand Uncertainty. Technical report, University of Toronto, 2012

[4] Y. Huang, Z. Li, G. Liu, and Y. Dai, "Cloud download: Using cloud utilities to achieve high-quality content distribution for unpopular videos," in *ACM Multimedia*, 2011.

[5] B. Li, and S. Zhao. Understanding Demand Volatility in Large VoD Systems. In *Proc. of the 21st International workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 2011.

[6] Raymond sweha, vatche Ishakian, Azer Bestavros, **Angels in the Cloud**, 2011 IEEE 4th International conference.

[7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A View of Cloud Computing. *Communications of the ACM*, 53(4):50–58, 2010

