The effect on the streaming content delivery using p2p network with and without clustering

Rishabh Mehan

NYIT – Old Westbury, Old Westbury, NY – 11568

I-Abstract

P2P content streaming is gaining popularity these days due to fast penetration of the high-speed Internet, high number of users and their demands. The new generation P2P live streaming systems not only attract a large number of viewers, but also support better video quality by streaming the content at higher bit-rate.

In this we shall look into the average download times in a network, which has a peer-peer or a super peer, structure whenever there is a multicast or data streaming (such as video stream). The study is based on a comparison between a clustered structure and a non-clustered architecture. Through k-means clustering algorithm the nodes will be clustered and based upon the request the data will be sent across the network from a particular cluster. The work will help us determine the factors which will enhance the efficiency of content delivery in a P2P network with clustering and also with this we will be able to serve more users without amending the infrastructure.

Keywords – Peer-to-peer, K-means, content streaming, clustering.

1. Introduction

P2P based file sharing, voice-over-IP, and video streaming services all achieve admirable success, attracting a large number of users and changing the way digital goods are delivered over the Internet. Even for companies dealing in business with limited number of clients, there is a wide range of applications of such systems, specially to the ones who are globally spread and needs to exchange data in real time and with efficient mechanisms.

The work related to this study would be to first compute the average download time of a non-clustered regular P2P structure irrespective of its architecture. The computation will be based on the transmission of packets of various sizes and then recording the time in which till they are successfully received. The clustering part

shall give us insights on how the distributed content delivery can be optimized and how better is it to use a infrastructure that is divided according to the requirements. The algorithm that will be used in the clustering will be the K-means clustering. Using that, the nodes of the P2P network will be clustered and will be evaluated for its average download time for the same data set that will be used in the non-clustered testing. The evaluation will be based upon the time taken for data to be requested and received by the client. It is simple to understand what we are doing, take an example of a company which has its servers all around globe and share the same data, and now if a user wants to extract that data then what will be better, fetching smaller chunks of it from all the servers spread around the globe or by fetching larger chunks it from the nearest servers. Based on the results we would be able to draw a perfect conclusion on which is the better method and in which cases.

2. Significance and Motivation

We are doing clustering because For Example – If a user wants to download a file shared in a network, all the available peers around the globe will start transmitting bits of that file; but instead of that, if the peers in his proximity, based on their availability and activity in the peer group, could transfer the same file but in larger chunks. Now when this happens not all the nodes of the network are used, and they can be used for another operation of serving other client.

Today in the heterogeneous set of users and every network having its own capacities and limitations, we cannot afford to lose time and also can't have cost-increasing solutions. Previous studies [4] have talked about the hybrid mechanisms that only reshape the architecture of the network and create a content delivery network. Also a theory about using the Multiple Descriptive Coding [2] was stated which distributes a single data stream into *n* sub streams that can be transmitted parallel. A Content Delivery Network has also been a success but the use of such an infrastructure is not feasible in every application. It doesn't assure you a real-time data, as there is always a cache involved.

The major impact which this paper can provide is knowledge of using alternate mechanisms, as per the requirement. That is, to know when a P2P network can be used as clustered to provide efficiency and when can it serve normally.

What we have proposed signifies to serve a basic user in any area, to share and stream data online in a peer without having to look for seeders around the globe with

different network configurations. It will not only reduce the cost but also provide an alternative to changing the complete infrastructure thus reducing the costs as well.

3. Related Work

Multimedia/data streaming over P2P is becoming popular these days as it provides a seamless interconnectivity between the client and the server. In some studies [1], solutions have been proposed which alters the BitTorrent content-distribution protocol that will allow the P2P network to deliver data on time. Besides all those studies [1-4], there was no precise study that could provide us the answers to what happens if we would just use the strength of the peers by clustering them on a real-time basis in terms of data hold, nearest nodes to client, availability and bandwidth.

The network assumed will be a basic architecture as described in the Fig3.1 (a) and will be clustered based on the factors discussed above, illustrated in Fig3.1 (b)

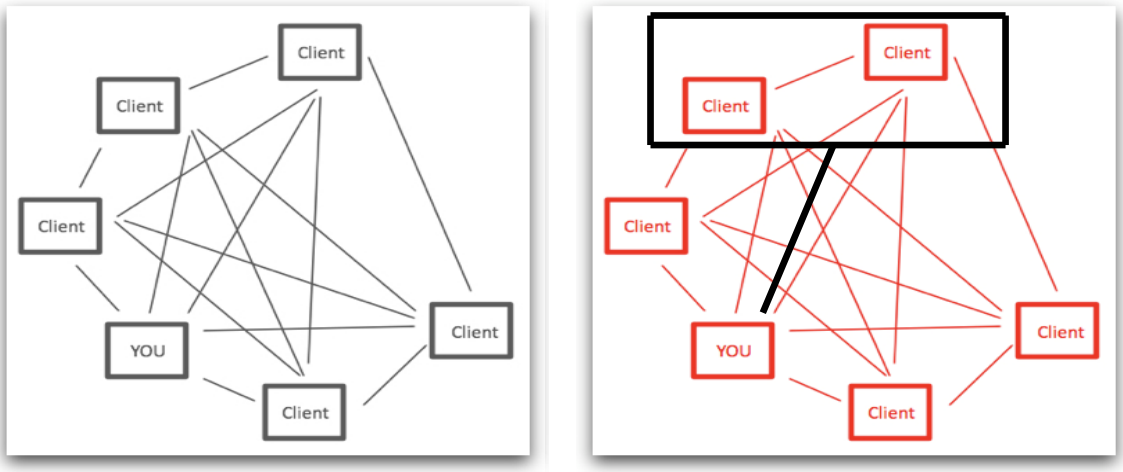


Fig3.1 (a) Non-Clustered

Fig3.1 (b) Clustered

Fig 3.1 - A diagrammatic view of clustered and Non-clustered Peer-to-peer architecture

A client when requests a data, all the online peers shall transmit the data in a simple non-clustered architecture. Considering a BitTorrent protocol, the packet size on each node will vary and depending on their bandwidths and the number of connected clients to a particular node will affect the average download time.

The Algorithm to be used will be the K-means algorithm which will distribute the nodes and cluster them. However, there are other proposed algorithms [5] which have successfully outperformed traditional k-means but we will try to keep this paper to k-

means as it gives us a standard view for our clustering. The number of observations required for the functioning of algorithm will be chosen based upon the number of nodes and the nodes which are near, available, and has available unused bandwidth. Once the clusters are formed we can easily assign one cluster to the client requesting the data. In meanwhile, all the other nodes that are not part of that cluster or that are not serving the client are available for the next client.

By measuring the average time for transmission of data of same size for both types of network we shall be able to plot a graph. That graph will help in the deduction and explanation of the result we shall receive.

4. Problem Formulation

The problem will consider *n* peers in a peer-to-peer network, holding various data. Queries requesting data may arise from any node, and will be served by the available nodes having the data. In figure 4(a) and 4(b) is shown how the network will appear after applying this filter to disregard unwanted peers, that is, the peers that don't hold the required data. Another thing about this approach is that it helps save time, as for example in the set of node (*a,b,c,d,e,f),* if node (*a)* requests a data and nodes (*b,c)* don't have it the filter provides a new set which consists of only (*d,e,f)* as the data provider nodes. This saves the time between the handshake between two extra nodes and getting a response of unavailability.

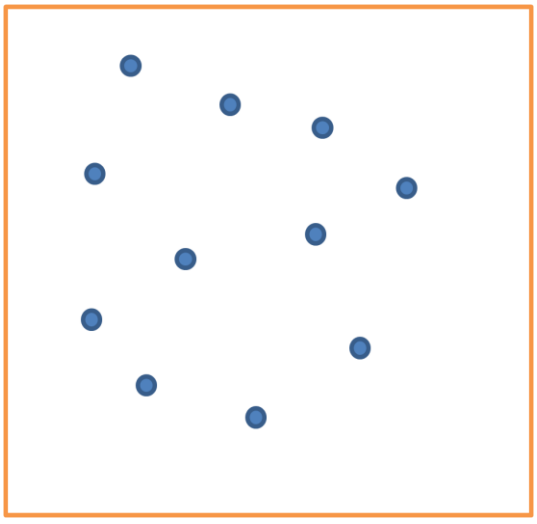


Fig4 (a) P2PNetwork's peers

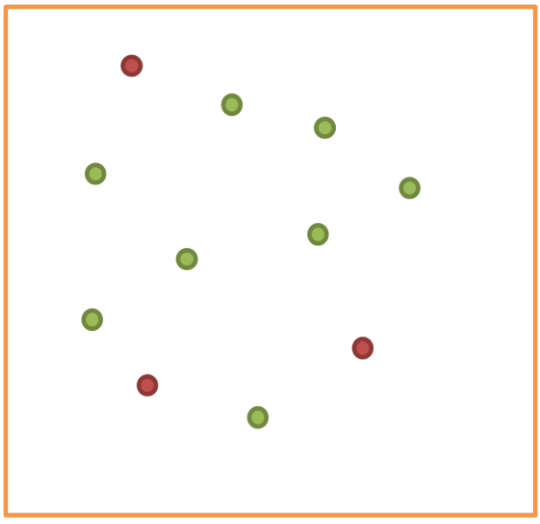


Fig4 (b) Green Showing the peers having the requested data and red showing the blanks

The nodes gathered will be clustered using K-Means Algorithm, which is described below.

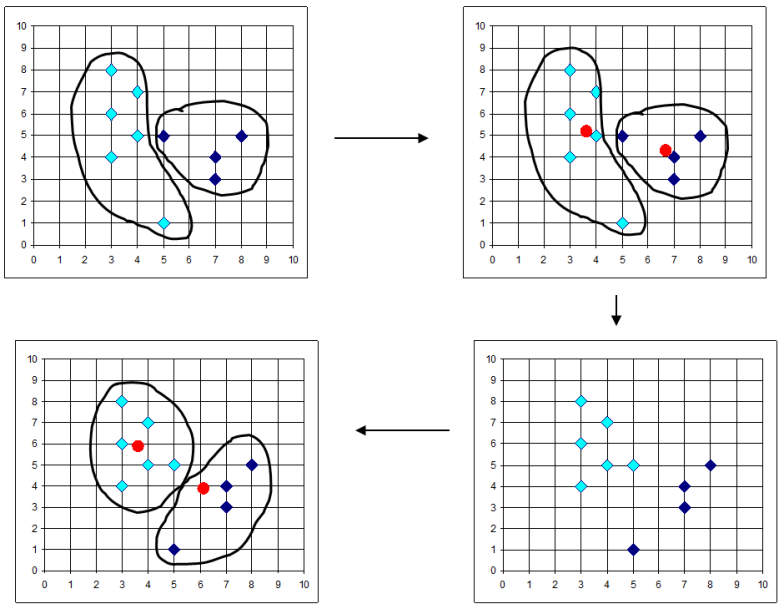
4.1 K-Means Algorithm

In simple words, K-Means clustering partitions data set into *k* segments, where *k* is a user-defined input, in our case it is the number of nodes requesting the data at a moment. The goal for k-means is to achieve clustering which minimizes the sum of each peer from the centroid (peer requesting the data).

The algorithm works as follows:

- Initially, the number of clusters must be known, or chosen, to be K say.
- The initial step is to choose a set of K instances as centres of the clusters.
- Next, the algorithm considers each instance and assigns it to the cluster which is closest.
- The cluster centroids are recalculated either after each instance assignment, or after the whole cycle of re-assignments.
- This process is iterated.

Example:



The K-Means algorithm attempts to minimize the squared error for all elements in all clusters.

The error equation is:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left| p - m_i \right|^2$$

Where E is the sum of the square error for all elements in the data set; p is a given element; and m_i is the mean of cluster C_i

4.2 Empirical Analysis

We discussed how the k-means algorithm will work to cluster the nodes in to *k* segments, where *k* is the number of peers requesting a data at a particular moment. We now shall do an experimental analysis to compare our proposed solution with the other studies [1-4].

The network we shall be looking at consists of six nodes (*a,b,c,d,e,f).* We assume the queue size to be zero for all the connections and a bandwidth of 100 Mbps. And int his example we shall consider the data to be sent to be 1000Mb. We shall calculate the latency based on a simple formula

Latency = Propogation + transmit +queue

Propogation = distance/speed of light *(speed of light = 3X10^8 m/s)*

Transmit = size/bandwidth

Below is the distance matrix for (a,b,c,d,e,f). Distance is in Kilometers

	A	**B**	**C**	**D**	**E**	**F**
A	0	4500	4500	6000	1500	1800
B	4500	0	1800	2000	3000	6500
C	4500	1800	0	1500	6000	6000
D	6000	2000	1500	0	6500	5500
E	1500	3000	6000	6500	0	2000
F	1800	6500	6000	5500	2000	0

Table 4.2(a)- Distance Matrix for the nodes (a,b,c,d,e,f)

Now lets see the two cases:

CASE I – In this, we calculate the latency for the network where *a,c* requests the same data at the same instant. And as a regular peer-to-peer network, the data is partitioned into equal number of chunks having same size and then transmitted by every node. Here, only *b,d,e,f* will be serving and we shall only look into the average download time of the transfer at C. Thus the numbers of partitions are 4 and each node shall transmit 250 Mb of data. (As 1000Mb/4 = 250Mb)

Using the formula described above for latency we get:

E=>C: 2.52 sec

F=>C: 2.52 sec

B=>C: 2.506 sec

D=>C: 2.505 sec

The total download time = (2.52+2.52+2.506+2.505) = 10.051s

Note – We have not yet considered the delay occurred in A=>C, since A didn't have the data that C required, it was unnecessary to even query it on A.

CASE II - In this, we calculate the latency for the network where *a,c* requests the same data at the same instant. Here, we will make use of our clustering and get the nearest peers to transmit the data chunks having same size and then transmitted by every node of the cluster. Here, only *b,d,e,f* will be serving and the clusters formed can is shown in Fig 4.2.1. Thus the numbers of partitions are 2 for the data transfer at C and each node shall transmit 500 Mb of data. (As 1000Mb/2 = 500Mb).

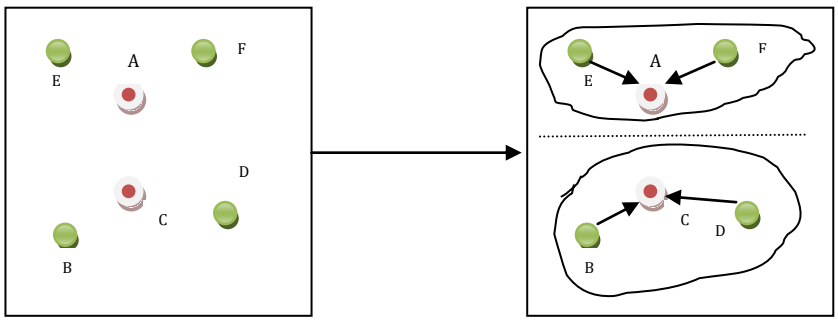


Fig 4.2.1: Formation of clusters using A and C as the centeroids.

For the data transfer at C, latency is calculated –

B=>C: 5.006

D=>C: 5.005

The Average Download Time is = (5.006+5.005) =10.011s

5. Results and Conclusions

In this paper, we successfully show that the clustering on a peer-to-peer network affects its total transfer time. As we progressed with the analysis, we were able to deduce that the total transfer time in clustering mechanism is lesser than the regular transfer. However, we have assumed multiple ideal conditions and have also not taken in count the time loss in CASE I of the analysis in section 4.2 from A to C.

Also the other factors proposed, such as the filter method described in section 4, and shown in Fig.4(a) and Fig. 4(b), helps in saving significant time delays.

With this method we hope to cut down costs, involving network architecture costs, to a significant level. Also try the approach with a new clustering algorithm as described in other study [5].

6. Future Work

Since, this was just a beginning stage, in future we can look into different possibilities and network configurations. Such factors include variable bandwidth, larger data, more number of peers, and more number of requests per second. By working on these factors and analyzing the effects we can take this study further.

An Adaptive neural network clustering [5] can also be done to enhance the performance of the clustered network.

7. References

[1] Purvi Shah, Jehan-Francois Paris, "Peer–to-Peer Multimedia Streaming using BitTorrent" in Department of Computer Science, University of Houston, Houston, TX.

[2] M.Saranya, G.Sophia Reena "K-means based Error resilient Video Streaming in Peer to Peer networks" in International Journal of Emerging Trends & Technology in Computer Science, India

[3] Souptik Datta, Chris Giannella, Hillol Kargupta "K-means clustering over Peer to Peer networks".

[4] D. Tran, K. Hua and T. Do, "Zigzag: an efficient peer-to-peer scheme for media streaming," Proc. 22nd IEEE INFOCOM Conference, San Francisco, CA, 2003.

[5] Santosh K. Rangarajan, Vir V. Phoha, Kiran S. Balagani, Rastko R.Selmic, S.S. Iyengar, "Adaptive Neural Network Clustering of Web Users," Computer, vol. 37, no. 4, pp. 34-40, April 2004, doi:10.1109/MC.2004.1297299