File Sharing in P2P Systems for the Expected High Churn

SHRIKANT P. BHOSALE
Department of Computer Science & Engineering
SVERI's College of Engineering
Pandharpur India

AMIT R. SARKAR
Department of Computer Science & Engineering
SVERI's College of Engineering
Pandharpur India

Abstract— **Most of the file sharing networks plays a major role in the current networking domain. On that the file sharing between the users is an important work takes place on those systems. When considering each of the system, the need is a performance improvement in the network. We address the problem of highly transient populations in unstructured and loosely structured peer-to-peer (P2P) systems. In existing system they consider the distributed system and causes collision while communication. But we consider centralized system and overcome the collision during file sharing and also the existing system drawbacks.**

Keywords-**File sharing, peer to peer, Distributed system, Centralized System.**

I. INTRODUCTION

While a peer is a computer that behaves as a client in the client/server model, it also contains an additional layer of software that allows it to perform server functions. The peer computer can respond to requests from other peers. The scope of the requests and responses, and how they are executed, are application-specific. Typically, there will be a request for access to resources that belong to the other peer. The request may be for information on content and files, or for a file to be read or copied, computations to be performed, or a message file to be passed on to others. When "computing" in P2P computing is used as a noun, it refers to a framework or computing model. This frame work provides the capabilities that allow peers to directly interact. An important characteristic of the direct-interaction capability is that the computing environment becomes decentralized. When the "computing" in P2P computing is used as a verb, it refers to what we do with P2P framework. Many end-user applications become possible through the P2P services, including storage, computations, messaging, security, distribution, and more. What unifies these application types is sharing of resources with some form of collaboration. Some P2P evangelists draw a distinction between so-called "pure P2P computing" and "hybrid P2P. " The term "'pureP2P computing" refers to a model, such as free net, where all participating computers are peers. No central server is used to control, coordinate, or manage the exchanges among the peers. In the "hybrid P2P" computing model, the application relies on a central server to perform some of the required functions. The degree of involvement varies with the application. For example, Napster requires the user to first connect to a control server, where the directory of all available files is stored.

When computers moved into mainstream use, PCs (personal computers) were connected together through LANs (local area networks) to central servers. These central servers were much more powerful than the PCs so any large data processing took place on these servers. Since then, PCs have become much more powerful, and they are able to handle the data processing locally rather than on central servers. Because of this, PC to PC, or peer-to-peer computing can now occur when individual computers bypass central servers to connect and collaborate directly with each other.

II. PEER TO PEER

A. A churn- Resistant Peer-to-Peer Web Caching System

Peer-to-peer applications may be subject to denial of service attacks from extreme stresses with origins typically considered non-malicious. They have studied one such source called churn, that arises from rapid arrival and failure (or departure) of a large number of participants in the system, and they have done so in the context of a peer-to-peer web caching application. Other malicious attacks are possible but they do not address these in this paper. This paper has shown how to design a churn-survivable peer-to-peer application. Their study has focused on the caching of web objects, and their solution has relied on the use of probabilistic techniques in the framework of the Kelips peer-to-peer overlay .Evaluation through micro bench marking on commodity clusters, as well as experiments done through a combination of web access logs, transit-stub topologies, and p2p host avail-ability traces, reveal significant advantages of locality and load balancing over previous designs for p2p web caching. Hit ratios and external bandwidth usage are both comparable to that in centralized web caching. In a system with a 1000 nodes, background communication costs as low as 3KBps per peer suffice to ensure favorable

and stable hit ratio, latency, external bandwidth use, and load balancing for access of web objects in the presence of system churn that causes 10%-25% of the total number of nodes to turn over within a few tens of seconds.

Demerits

In this paper they did not consider several interesting directions – [1] the hit ratio and latency behavior of Kelips web caching at other operation points than the \50% available - 50% churned" above, (2) the effect of churn on caching scenarios other than web page browsing, and [3] the feasibility of the Kelips constant-cost low-bandwidth solution to other applications and other stressful networking environments.

Transience of Peers & Streaming Media

In this paper, they argued that the migration of the multicast functionality from the network layer to the application layer results in a fundamental shift in the underlying infrastructure. The infrastructure units are now participating clients. That are known to be more whimsical and transient that network routers. The observation is especially true for P2P networks, where the participating peers are the providers of an end-service. They urge that the end-host multicast schemes be evaluated on end-system performance metrics of relative delay-penalty, response time, packet losses, and lost-block widths. They indicated that making peer transience is a primary challenge in such a domain. They introduced the design philosophy of a peering layer at each peer that isolates policies for maintaining the topology from end-application functionality. The applications at a peer now need not be aware of a change in the server providing its data feed.

B. Napster

Napster[2] is a name given to two music-focused online services. It was originally founded as a pioneering peer-to-peer file sharing Internet service that emphasized sharing audio files, typically music, encoded in MP3 format. The original company ran into legal difficulties over copyright infringement, ceased operations and was eventually acquired by Roxio. In its second incarnation Napster became an online music store until it was acquired by Rhapsody from Best Buyon December 1, 2011. Later companies and projects successfully followed its P2P filesharing example such as Gnutella, Freenet and many others. Some services, like LimeWire, Grokster, Madster and the original eDonkey network, were brought down or changed due to similar circumstances.

III. METHODOLOGY

In the proposed system, they consider the centralized system, in that we have number systems and a super-peer. Form those requesting peer user can download the files as much as possible fast and efficiently. Before that we form the network environment to show the process. The user can view the files available in the super-peer if that peer gave the permission after that only the requesting peer can download the file from this the authentication is providing there. Then the normal peers are act as the sub super-peers to the requesting peer who have the files and also nearby user to the requesting peer. The requesting peer who acts as the sub super-peer information is also given to the end user by the super-peer. Then the requesting peer have to send request to the client who going to act as the sub super-peer. After getting the permission to the requested file from the sub super-peer then the requesting peer can download the file what the user requested to download.

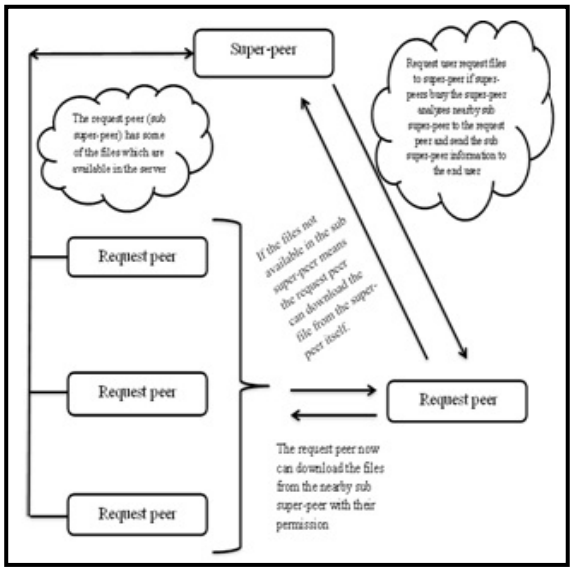


Figure 1: The architecture diagram of File sharing in P2P System

C. Peer Module

In this module, Peers have to register and then they can authenticate the peer to enter into the further process. After registering they can login and they upload the file and they can upload and download the file. And the super peer can upload the file to their data base.

D. File Upload

Each login process is checked by the super peer. The login peer can upload the file into the super peer database. Then they can send query to the super peer about the files which was present into the super peer. Then they can get reply from the super peer based on that they can download the file with the super peer permission. The uploaded files are maintained in the separate storage area.

E. Query Requesting

In this module, the login user can search the files by using searching process by giving the keyword

related to the files which was available in the super peer database file list. The query requesting is first received by the super peer. Then the super peer first receives and then the super peer check the file availabilities in the other available peers and also check whether they are the neighbors to the requesting peer.

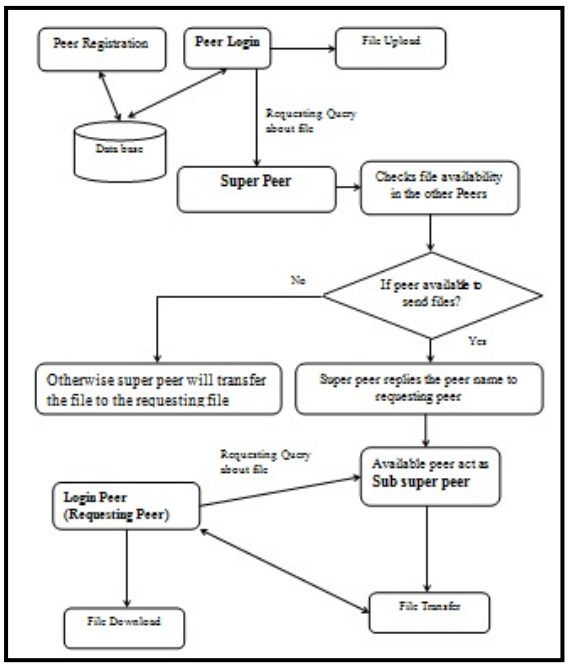


Figure 2: The data flow diagram of file sharing in P2P systems

F. Monitoring

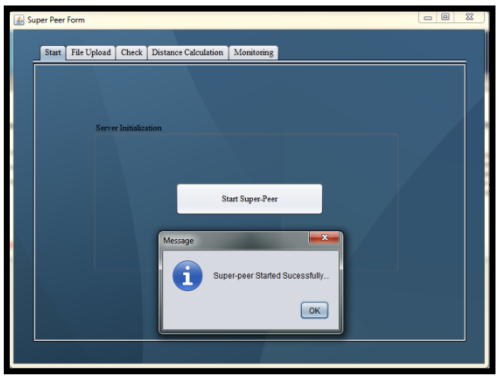
The requesting queries are processed by the super peer and then after getting the check result whether the peer are available to send the requesting file to the requesting user. The requesting user will get the sub super peer detail which was send by the super peer to the requesting user. After receiving the sub super peer details to the requesting peer. The requesting peer will request the sub super peer related to file download as like query requesting to the sub super peer.

G. File Download

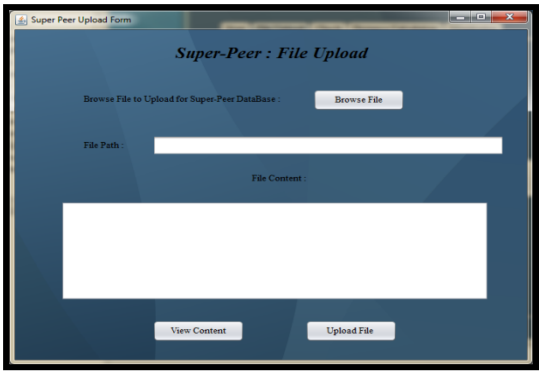
In this module, after giving request to the sub super peer the sub super peer will transfer the requested file to the requesting user. Then the requesting user will receives the file and there is an option to save the received file. If there is no available user to act as the sub super peer means the super peer will transfer the file to the requesting user. After saving we can evaluate the performance in the terms of time consumption.

IV. RESULTS AND CONCLUSIONS

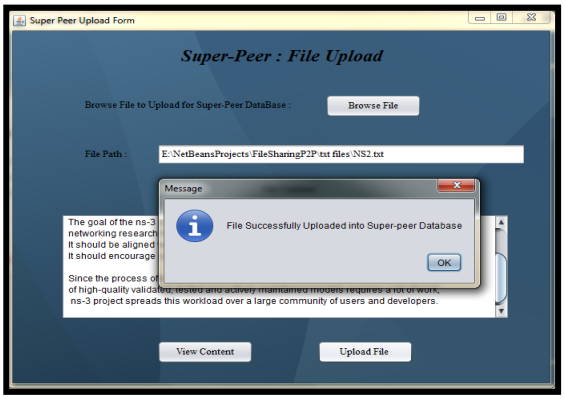
A. Step 1: Start Super Peer.



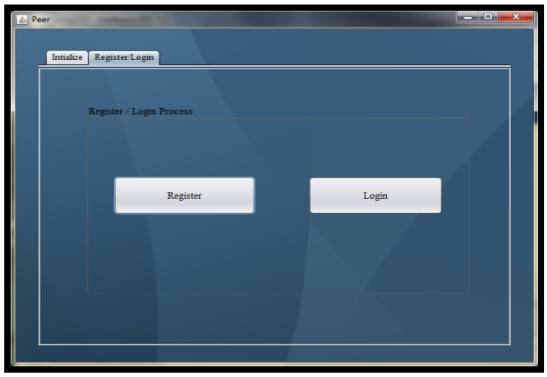
B. Step 2: File upload at Super Peer.



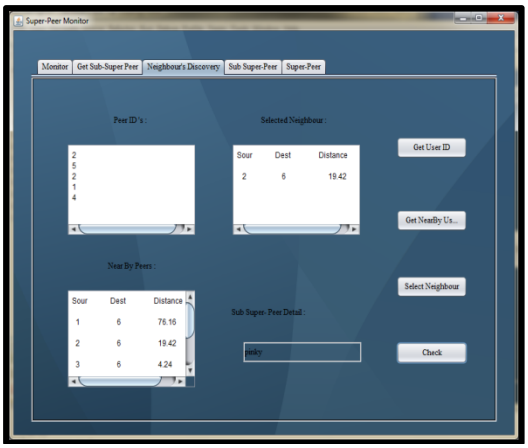
C. Step 3 : Successfully upload file at Super Peer



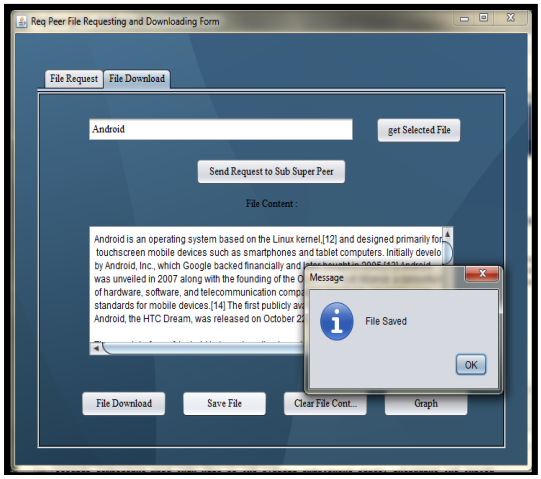
D. Step 4: Peer Login and Registration



E. Step 5 : Neighbor's Discovery



F. Step 6 : File Download and view



V. CONCLUSION

We conclude that the proposed system have the more efficient and high performance providing process. This process is takes place between the super-peer and n number of sub super-peer and the request peer. Those the main issue to improve the file sharing process in highly transient populations in unstructured and loosely structured peer-to-peer (P2P) systems

VI. ACKNOWLEDGMENT

WE WOULD LIKE TO ACKNOWLEDGE THE STUDENTS OF COMPUTER SCIENCE & ENGINEERING DEPARTMENT, SVERI'S COLLEGE OF ENGINEERING, PANDHARPUR FOR THEIR SUPPORT. I MR. SHRIKANT BHOSALE SPECIALLY WANT TO THANK MY GUIDE PROF. AMIT R. SARKAR FOR THEIR GUIDANCE.

VII. REFERENCES

[1] Y. Qiao and F. E. Bustamante, "Elders know best—Handling Churn in less structured P2P systems," in Proc. IEEE Int. Conf. Peer-to-Peer Computing(P2P), Konstanz, Germany, Aug.–Sep. 2005.

[2] Napster.2003[Online].Available:http://www.napster.com.

[3] M. Bawa, H. Deshpande, and H. Garcia-Molina, "Transience of peers and streaming media," in Proc. HetNets-I, 2002, pp. 107–112.

[4] P. Linga, I. Gupta, and K. Birman, "A Churn-resistant peer-to-peer web caching system," in Proc. ACM Workshop on Survivable and Self-Re-negerative Syst., Oct. 2003, pp. 1–10.

[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic.

