Reliable File Sharing in Distributed Operating System using Web RTC

Rajesh Dukiya
School of Information Technology & Engineering
VIT University
Vellore, India

Shubhankar Sharma
School of Information Technology & Engineering
VIT University
Vellore, India

Prof. Sriniwas Koppu
School of Information Technology & Engineering
VIT University
Vellore, India

Madusudhana Rao
Department of Information Technology
Higher College of Technology
Muscat, Oman

***Abstract*— since, the evolution of distributed operating system, distributed file system is come out to be important part in operating system. P2P is a reliable way in Distributed Operating System for file sharing. It was introduced in 1999, later it became a high research interest topic. Peer to Peer network is a type of network, where peers share network workload and other load related tasks. A P2P network can be a period of time connection, where a bunch of computers connected by a USB (Universal Serial Bus) port to transfer or enable disk sharing i.e. file sharing. Currently P2P requires special network that should be designed in P2P way. Nowadays, there is a big influence of browsers in our life. In this project we are going to study of file sharing mechanism in distributed operating system in web browsers, where we will try to find performance bottlenecks which our research will going to be an improvement in file sharing by performance and scalability in distributed file systems. Additionally, we will discuss the scope of Web Torrent file sharing and free-riding in peer to peer networks.**

Keywords— WebTorrent, P2P File Sharing, Distributed operating system, network workload sharing

I. INTRODUCTION

Distributed operating system is a software used for running distributed system which refers to bunch of computers interconnected via LAN/WAN each computer having their own computational resources used by each other. Distributed operating system is an advancement of network operating system differing that it is not a centralized operating system. The rule of thumb in distributed operating system is that no user is in interconnected system should know which machine they are using where their programs been storing from whom they are downloading the files, if you can tell which computer you are using with how much resources you are, it is not a part of the distributed operating system. After having the distributed system the interconnected user should take the advantages of using vast and variety of resources sharing like CPU processing power, file sharing, printers etc. Into the system but not knowing from whom let the user feel like they are part of a centralized system in context they actually not so in this project we are going to analyze web torrent file sharing system which generally, follows P2P method for sharing between computers. P2P is very resource efficient way in distributed operating for file sharing. Even though there are many P2P systems for file sharing systems are out in the market but very few areimplemented and succeeded like Napster and supernova.org but they are long gone by now for sharing the files with illegal and copyright issues, Therefore, we will propose a implementation of P2P file sharing in a different way where first share a file on a virtually created network and later we analyze the performance issues and give a solution.

II. P2P NETWORK STRUCTURE

P2P computing concept and architecture is purely an application of distributed systems. The main motive of Peer to Peer is sharing the tasks and workload among networks connected peers. In P2P system all peers have equal weightage and they participate equally. These peers act as the network nodes. These nodes join resources like computing power CPU, hard disk and other storage media, network bandwidth and other shareable components, the output remains in a single unit without need of a server or central operating model. Peers can play client and server roles simultaneously for resources, while in case of generalized client-server architecture it use fixed resources allocation. We can create a unique functionality with general resources using P2P systems, it cannot possible by a single peer. Additionally, these results are beneficial for all connected peers.

Peer to Peer networks follow two type architecture, we can classify the architecture as structured and unstructured. Structured networks are implemented virtually over traditionally physical network topology. We can separate like a subset of the traditional network topology. In structured network data exchanged over TCP/IP protocol using an overlay, however all peer are able to communicate directly. The overlay helps in peer indexing and preserve peer metadata for communication. Unstructured networks do not follow any special connection or architecture rule to establish connection with other peers. They establish random connections to others. Additionally, we can define a new network architecture using above tow model, Hybrid model. In this model some partition of network follows structured architecture and other part follows unstructured. Hybrid model can helpful to achieve optimal results. We can understand architecture by these diagrams.

978-1-5386-1716-8/17/$31.00 ©2017 IEEE

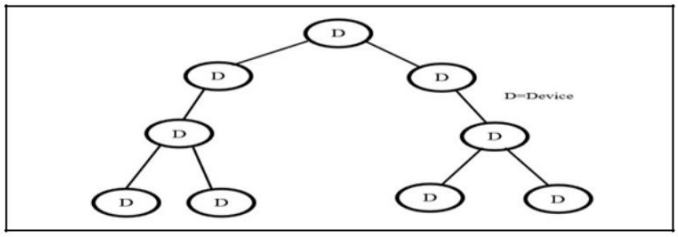

Fig. 1 Structured peer-to-peer networks: organized into a specific topology

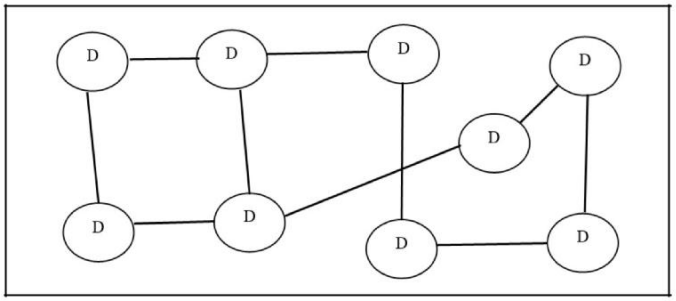

Fig. 2. Unstructured peer-to-peer networks, random node connections

A. The Architecture Components

i. Broadcasting - We can send the messages with two ways in Peer 2 Peer commination. Either it can be broadcast message or it may be for a particular peer.

ii. Dropped Packets - The small fragments of data which has to be sent, sometimes drop due to a failure in network. We can classify these packets as dropped packets.

iii. GUID - Global Unique Identifier, the GUID act as a node index on p2p networks. It is a randomized string, which is helpful to locate a peer or deliver a message to the peer.

iv. Hops - Every packet follows a path to reach its destination, number of peers in a path called Hopes.

v. Host Catcher - During first join of a P2P network, user send a ping request to others. The P2P client's host catcher maintains other hosts list. Using this list software establish the automatic connection also.

vi. Peer - We can consider two network connected systems as peers if they are playing similar role at same time during sending. For instance, a laptop in a college might connected with the college's library server but it is playing as a client so we cannot say peer to that computer.

vii. Ping - When a new peer tries to connect with network. It broadcast a message, it called as a ping.

viii. Pong - If a user ping than a peer receive it, receiving a ping by older user and replying in that network called as Pong. This pong have replier peer IP with port and data details. Port - Communication applications has specific port to communicate. On most of P2P servant software has default port 6346.

ix. Push request - When user is using software behind the firewall, software send a request to other users for communication.

Identify applicable funding agency here. If none, delete this text box.

x. DHT: It is short of Distributed Hash Table, a table that provides host lookup like service to connect with other peers.

B. P2P Architecture

File sharing in distributed system using p2p is a distribution of content and file with peers using optimal load balancing. General internet user can download various type of day to day useful software, music, videos, eBooks and data archives. In client server system to distribute these type of content, we need a very powerful server but in case of p2p we can do these networks, such as eDonkey network, TBP are well known.things without a host also. There are many p2p file distribution.

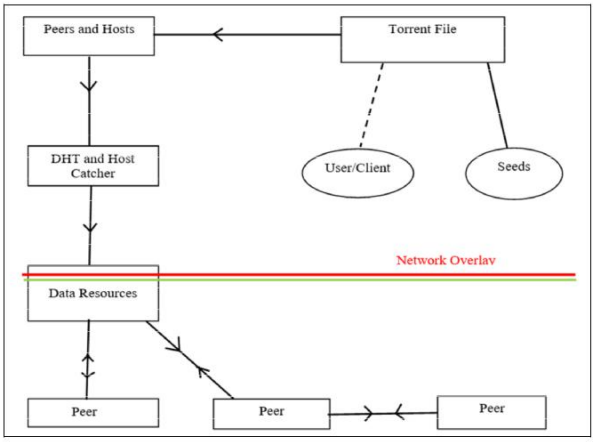

Fig. 3. Data Workflow and Distribution in P2P Networks

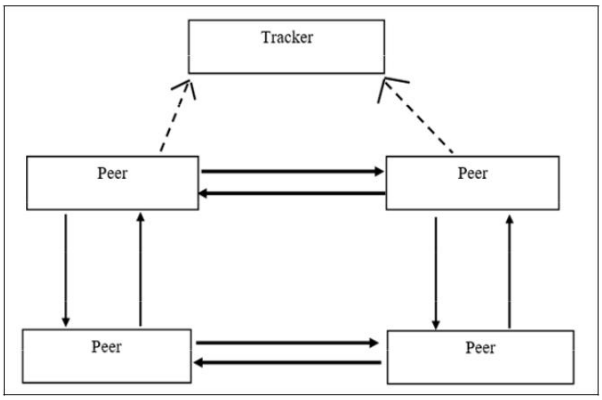

Fig. 4. Peer workflow in the network

III. Web Torrent Concept

Web torrent concept is relatively new in technology terms. It is dependent on peer's browser. In past some unreliable methods were available for browser to browser communication so its implementation was not possible. Now all popular browsers have WebRTC. WebRTC is a protocol that allows browser to browser communication, additionally it eliminate the centralize server system requirements. WebRTC is an open source project, additionally it is specially designed for light weight communication.

A. Program Design

1. Creating/collecting a .torrent file - During design our first objective is creating or collecting a valid torrent file. This file contains a map of our target download in metadata format. The data is file is encoded in bencode, which is an encoding format.
2. Analysis of a .torrent file - In this step we will decode the file and start analysis of metadata. After decoding we can see tracker information and data map.
3. Connecting to a tracker - We can get tracker url with 'announce' key in that metafile. The tracker is a just HTTP which contains sharable data and peers info. It do not have desired data but keeps record that how many users are connected, what they are downloading, how much they finished. It response on GET request.
4. Connect to peers - Peer connections are made through TCP to the appropriate host IP and port. Now might be a good time to consider how or if you want to deal with connecting to multiple peers at the same time, as this will influence how you connect to your peers.
5. Handshake with peers - After made once you have a connection to your peer(s), the first contact step is your responsibility. The first message you send should be a Handshake.

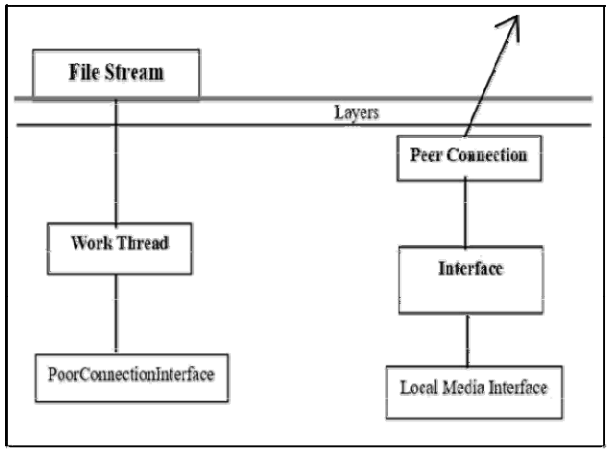

Fig. 5. Web Torrent Architecture

B. Difference between Ordinary and Browser based Torrent

Theoretically, both type clients are working on same topology, but technically the problem with web torrent is conversion of protocol for web browser and p2p node. First condition is the WebRTC supported browser. Let we have a WebRTC enabled client still we need a bridge between WebRTC and P2P. The main advantage of WebRTC is its lightweight resources consumption during P2P communication and there is no requirement of any plugins

.

IV. CASE STUDY

Peer-to-Peer (P2P) sharing method [1] has risen as a critical worldview for providing distributed services, specifically find a node and sharing of information. Right now P2P systems are developed and maintained by members taking after their own particular ungraceful protocols. They therefore experience the bad effects of continuous network overload, partitioning and transfer speed. In this paper, researcher's tires to implement P2P networks using a different distributed method. Author give a distributed protocol to build systems with great topological properties— to be specific, steady degree, network, what's more, low width. An appealing element of the protocol is that it is easy to actualize. They break down the protocol under a practical element setting and demonstrate thoroughly that it brings about the above properties with huge probability. We likewise demonstrated that our protocol is normally robust to disappointments and that it has decent self-correcting properties, for example, rapid recovery from network error. In this paper[2], the author shows that ISP can collaborate with P2P system to make or earn profits and improve performance. See that the p2p has emerged as a very agnostic in the recent years so to overcome this problem they propose a solution that ISP should collaborate or P2P systems so that ISP can earn much profits and users can enjoy improved performance. So the idea behind if an ISP can issue an Oracle to its user to keep track of neighbors p2p nodes or detail to whom to download the content and this technique tells that whosoever have the greatest potential of giving the most official proper content and it rank to querying nodes bestowing to user's locality. For ranking they proposed a metrics evaluates the impact of using the Oracle method which shows the entities like small mean path lengths and small diameter and node degree the result will be calculated as graph based simulation as well as experimented method can put changes in p2p clients. In this paper [3] here author describes how to make a super-peer Networks so first of all super-peer network is entity of p2p networks that can play the role of both as a server to multiple clients and can be one in a network of super-peers. The authors describes in this paper how they examine the super- peer networks in brief, and showing the performance benchmark tradeoffs and the fundamental characteristics of super node. They also teach the process of design of efficient super-peer networks. As a result they gave 4 rule of thumb.

Redundancy, it is good in super peer networks. Second, increase the cluster size will drops aggregate load, but escalates individual load. Third, maximum out degree, and last minimize Time to live.

The paper[4] presented an extent study on the two most popular new and old file sharing system based on P2P Gnutella and Napster this study is conducted on several population of peer that choose to join in both the file system. This paper first shows the introductory part of both systems benchmark performance, rate distortion, speed etc. this study also include the measurement of bottleneck of bandwidth, file sharing, latency, availability patterns of these peers. There are several results emerged from carrying

the problem statement give evidence of the heterogeneity in both Napster and Gnutella. This implies that it's important to know the other peer a little when providing the delegate information, second that both have different responsibility in as server like and client like behavior, Third Peers tend to deliberately misreport whenever they want to, because information is much more important because the future machines shall be built on incentives for peers to tell the truth or system must be able to directly measure and verify the information report.

The author [5] is describing about the most popular P2P architecture Gnutella network which is open architecture based on application level like most of the P2P. To extract the topology at the application layer the author built a crawler technique in this they analyze the graph and network traffic of the topology. The purpose here to make changes in the P2P coding to improve performance and scalability improvement. As result shows that Gnutella node connectivity follows the multi model distribution with the combination of quasi-constant distribution and power law, this property keeps the reliability in network whenever a random node malfunctions, also the Gnutella gives protection to few malicious attack. The two mostly seen conclusion is that the firstly we observe the application level topology defines the generated traffic of volume, the application reliability and application search success rate furthermore the nodes can understand themselves about the physical network and can create virtual application topology automatically secondly the smart routing feature added whenever there is a chance of flooding and improved group communication mechanism. The paper[6] proposes a HCDN Hybrid content distribution network integrating the compliment advantages of CDN and P2P that is used for content distribution to improve the efficiency to achieve the in depth effectiveness of HCDN we evaluate a detailed performance changes in deterministic model of fluid they also show the results of P2P ,CDN and HCDN individually and performance metrics are carried out the analysis such as no. of seeds and downloader the service capacity of system and marking the average downloading time and the service capacity. As a result they have finished an in depth performance analysis on HCDN comparing with content delivery network (CDN) and pure peer to peer Network. After the long analysis and numerical output they have concluded that HCDN has a lot of merits when talking about average downloading time, system scalability and service capacity. The paper [7] starts with a discussion starting with the concept of an event handled before another event occur in distributed system is studied which displays the partial ranking of events. An algorithm in distributed system is proposed for syncing logical clocks of system which can be used to order the events happened. The total ordering is solves the synchronization problem. The paper created by author unfolding the algorithm for extending the partial ordering to subjective total ordering this shows that a simple ordering can be helpful in solving simple synchronization of clocks. The idea of total ordering of events is somewhat arbitrary it is shown that the algorithm closely synchronizes the clocks. It should be clear that in distributed architecture event occur in partial ordering. Their idea is valuable in understanding multiprocessors structure, it helped us in understanding the problems in multiprocessing.

CONCLUSION

In this paper, we have presented a review and study distributed file sharing with Web based p2p sharing method. The result was interesting, we can apply this method with many existing systems and modules. There will be a boom in upcoming days in P2P cum WebRTC applications. We analyzed that goodness of both protocols will became an innovative foot step in field of IOT internet of things, smartphones, cloud and many higher bandwidth based sectors like space science. However there is still communication gap between RTC and P2P. The problem can be solved using NPM library so by taking advantage of platform independent npm libraries we can change direction of file sharing in distributed operating system.

REFERENCES

[1] Pandurangan, G., Raghavan, P., & Upfal, E. (2003). "Building low-diameter peer-to-peer networks." IEEE Journal on selected areas in communications, 21(6), 995-1002

[2] Aggarwal, V., Feldmann, A., & Scheideler, C. (2007)." Can ISPs and P2P users cooperate for improved performance?." ACM SIGCOMM Computer Communication Review, 37(3), 29-40.

[3] Yang, B. B., & Garcia-Molina, H. (2003, March). "Designing a super-peer network. In Data Engineering," 2003. Proceedings. 19th International Conference on (pp. 49-60). IEEE.

[4] Saroiu, S., Gummadi, P. K., & Gribble, S. D. (2001, December). "Measurement study of peer-to-peer file sharing systems. In Electronic Imaging" 2002 (pp. 156-170). International Society for Optics and Photonics.

[5] Ripeanu, M. (2001, August). "Peer-to-peer architecture case study: Gnutella network. In Peer-to-Peer Computing," 2001. Proceedings. First International Conference on (pp. 99-100). IEEE.

[6] Jiang, H., Li, J., Li, Z., & Bai, X. (2009). "Efficient large-scale content distribution with combination of CDN and P2P networks." International Journal of Hybrid Information Technology, 2(2), 4.

[7] Lamport, L. (1978). "Time, clocks, and the ordering of events in a distributed system." Communications of the ACM, 21(7), 558-565.