IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

A Survey of Peer- to-Peer Networks

Rajesh Kumar Maurya[1], Prof. Suman Pandey[2], Vinod Kumar[3]

M. Tech, Computer Science & Engineering, KNIT, Sultanpur, India[1,3]

Professor, Computer Science & Engineering, KNIT, Sultanpur, India[2]

Abstract: Peer-to-peer (P2P) networks are used to the consideration ofa worldwide with their excessive success in the file sharing in the networks (such as Napster, Gnutella, Kazaa, BitTorrent, JXTA and Freenet). Increase the popularity of (Peer to Peer) P2P networks has been witnessed by millions of Internet users. In this paper, an analysis of network architectures evolution, from client to server Peer to Peer (P2P) networking, will be given, underlining the benefits and the probable problems of existing approaches, which provides essential theoretical base to drive future generation of distributed systems. Peer to Peer (P2P) are important improvements on large scale DS (distributed systems) design and evolution of the Internet architectures. Widely used applications have a great pact of thepractical result has been followed in the research area for improvement. (P2P) Peer-to-Peer content/file sharing, mediastreaming, and telephony applications. There are a large range of other applications under improvement or being proposed. The underlying architecture shares the feature such as decentralization, sharing of system resources, virtualization, autonomy and self-organization. These features constitute the P2P paradigm. [1].

I. INTRODUCTION

In the recent years, Peer-to-Peer system research has grown significantly. Using a large scaledistributed networks of the machines has become an important element of the distributed computing due to the phenomenal popularity of P2P(Peer-to-Peer) services like **Napster, Gnutella, Kazaa, Bit Torrent, and JXTA**. As new design of P2P(Peer-to-Peer)network is being widely used in the design oflarge distributed applications. Ahigh increase popularity of P2P (Peer-to-Peer) file sharing, video streaming and telephony applications has been witnessed by millions of Internet users. As a highly emerging technology, P2P (Peer to Peer) network is attracting the attention of researchers worldwide, ranging from irregularly Internet users to undertaking capitalists. These days, the innovations of P2P (Peer-to-Peer) networks also offer the many interesting approach of research for scientific communities. Since few years, Peer-to-Peer resource sharing and data transfer have many implementations. Network architectures are starting to evolve from the centralised client-server architectures to the distributed Peer-to-Peer architectures or hybrid architectures between client-server and Peer-to-Peer.In this paper, the networks architecture developments are discussedas ofclient to server P2P. A summary of recent solutions for discovery of resource in Peer-to-Peer network is also given underlining the benefits and the prospective problems of these solutions.

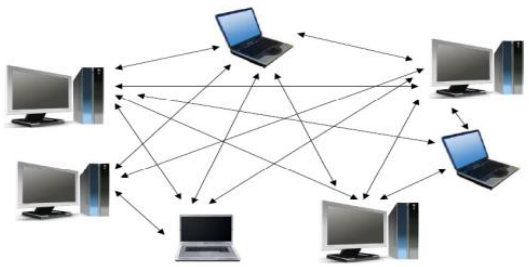

Fig1:-Pure P2P Architecture

II. NETWORK ARCHITECTURE EVOLUTIONS

2.1 Client Server Architecture:-The client to server architecture is the many number of clients request and receive services from server via Internet applications, i.e. LAN, WAN such as WWW, FTP, email etc.

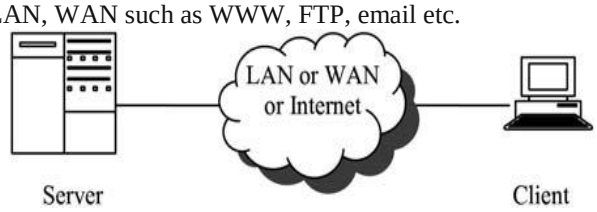

Fig2:-Client-server architecture

However, such a centralization of client to server architecture elevate a series of issues which are caused by the limitation of resources at the server side, such as network bandwidth, CPU capability, Input/output (I/O) speed and storage space. A server in the network could be overloaded traffic of the networks if too much number of requests is received. In order to manage with these limitations, the centralized server needs to bear the high costs of providing sufficient resources. For instance, of the Moreover, the centralization of client server architecture leads the single point of failure problem. If centralized server is removed or is not available for use, no alternative in the architecture can take its place and all services on the server will be lost.

2.2Greed Architecture:-Greed Architecture is quickly emerging from the scientific and academic area to the industrial and commercial world. Current Grid computing systems are important implementations of client-server architecture for distributed computing.The Grid computing systems provide high performance computing and data infrastructure supporting to the flexibility, security and coordinated resource sharing between dynamic nodes and institutions known as “virtual organizations” [2, 3]. The main centre of Grid architecture is on interoperability

DOI 10.17148/IJARCCE.2016.5475

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

among resource providers and users in order to establish the sharing relationship, which needs common protocols at each layer of architecture. It is purporting to offer intact and uniform access to substantial resources without considering their geographical locations. Resources in the Grid can be high performance supercomputers, massive storage space, sensors, satellites, software applications, and data belonging to different institutions and connected through the Internet. The Grid provides the infrastructure that enables sparse institutions such as commercial, companies, universities, government institutions, etc. to form virtual organisations that share resources and collaborate for the purpose of solving common problems [2][3].

III. PEER-TO-PEER NETWORKS EVOLUTION

There are many interesting types of P2P applications, including file sharing, instant messaging, Streaming media, VoIP, High Performance Computing, search engine. Among them, file sharing, media streaming have become one of the most famous online activities [4], is the initial motivation behind many of successful P2P networks. P2P file sharing has become one of the most well-liked Internet activities. Today's popular P2P file sharing application, such as Kazaa and Gnutella, numbering more than one million users each at anypoint of time [5]. In this section, the history of the Peer-to-Peerfile sharing network is discussed alongside with the popular file sharing applications. Existing Peer-to-Peer file sharing networks can be divided into three categories [8] according to the degree of network centralisation: centralised P2P networks, decentralised P2P networks and hybrid P2P networks.

3.1. Centralised P2P networks:-

Although Peer to Peer (P2P) is commonly seen as an opposite model to the centralised client-servermodel, the first Peer to Peer(P2P) systems (e.g. Napster) started with the idea of centralisation. However, in thedifference to conventional client-server systems, the serverin centralised Peer to Peer(P2P) networks only conserve the meta-information about shared content (e.g. addresses or ID of peer nodes where the shared content is available) rather than storing content on its recognize.

2.3Napster:-Napster was the first widely-used P2P withmusic sharing service. Before Napster came along, Internet users only passively operated their connected computers, such as browsing news or checking email. With the increased popularity of Napster, ordinary Internet users started opening their PCs to actively contribute resources andplayed more important roles for the Internet.Compared to follow-up P2P applications, Napster utilises a simple but highly efficient mechanism to share and search files within the network. To participate in the Napster network, new users goes with registration to the Napster server and put out a list of files they are eager to share. To search for a shared file in the network, users request the Napster server and retrieve a list of service providers hosting the files which match the query. File transfer can takes place without the participation of Napster server. The requested file is transferred directly between the requester and the provider as shown in Fig:-3.[6]

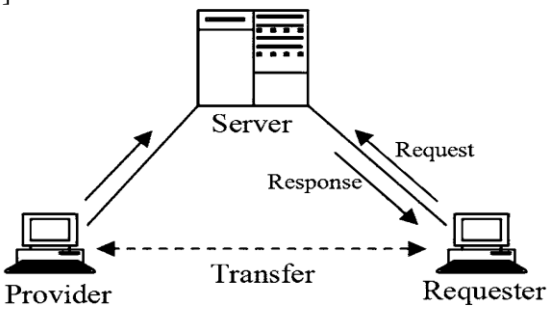

Fig:-3Example of Napster network

3.1.2Bit Torrent:- Bit Torrent are designed to share out large data amounts without realized the subsequent consumption in server and bandwidth devices. The original Bit- Torrent (before version 4.2.0) can be looked at as a Napster-like centralised Peer to Peer(P2P) system. To share single or multiple files, users need to create a small. torrent file that contains the address of the tracker machine that launches the file distribution. This .torrent files are published on well-known web-sites, so that users can search and download. Torrent file of interest using web search engines. The .torrent files are opened by the BitTorrent client software. The client software connects to the tracker machine and receives a peer node list that is participatingin transferring the file. For efficiently distribution of file, a file is broken into smaller data packet (typically 256 KB each) for transmission. The client, trying to download file, contact the peer node simultaneously that are participating in file transfer, and downloads different section of the file from different peer nodes. In the meantime, the client can also upload downloaded section to other participants[6].

3.2 Decentralized P2P Architecture-

Addressing the problems of the centralised P2Pnetworks (scalability, single point of failure and legal issues), decentralised P2P networks become widely used, which do not trust on central server[6].

3.2.1 Gnutella:-Gnutella networksare decentralised file-sharing Peer to Peer(P2P) networks, which is built on an open protocol developed to potentially peer node discovery, search, and file transfer. Each Gnutella user needs Gnutellaclient software(GCS) to connect Gnutella network. The GCS on initial use bootstrap to find the totalnumber of possible active peer nodes in networks and try to link to them. If some attempts succeed, these working peer nodes will then become the newly node's neighbours and give the newly node their own lists of working nodes. The new node continues connect to these working peer nodes, until itreaches a certain portion (usually userspecified).The new node keeps the peer nodes it has not still tried as backup. When a peer node leaves the P2P network and wants to re-connect to the networks,

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

peer node tries to connect to the nodes whose addresses it has stored. Oncethe peer node re-connects into the network, it will periodically ping the network connections and update its list of node addresses. In opposite to Napster, Gnutella networksare decentralised P2P file or multiple files sharing networks not only for file storage, but also for content lookup and query routing. Gnutella nodes take over routing functionalities initially performed by the Napster server. Figure gives an example of query propagation over Gnutella network. In the Gnutella networks, each peer node uses a Breadth-First Search (BFS) mechanism to search the network by broadcasting the query with a (TTL) Time-to- Live to all connected peer nodes. TTL represents the number of nodes in route a message can be forwarded before it is discarded. All peer nodes receiving query will process it, and check the local file storage, and respond to processed query if any one matched file is found. Each peer node then decreases TTL by one andremits that query to all neighbours. This process continues until TTL decreases to zero.Gnutella network are not based on central server to index files, which avoids the single-point-of-failure issue and the performance bottleneck at server side. Instead, several peer nodes are visited by flooding queries to see whether they have a requested file. The drawback of Gnutella i.e., generates potentially enormous network traffic by flooding queries[6].

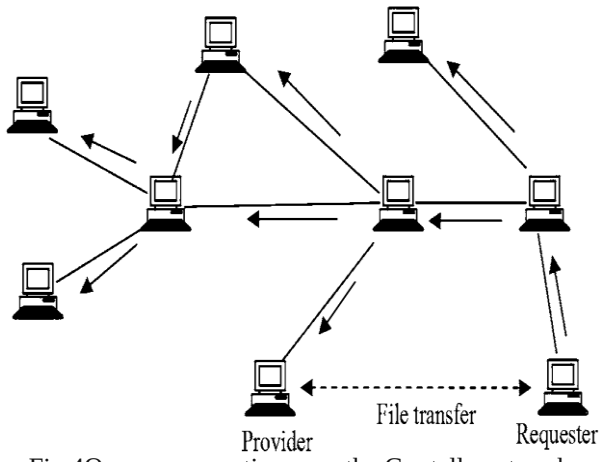

Fig.4Query propagation over the Gnutella network

3.2.2 Freenet:-Freenet is a decentralised P2P data storage system designed to provide electronicdocument exchange through strong anonymity. In contrast to Gnutella, Freenet acts as a P2P storage system by enabling users to share unused local storage space for popular file replication and caching. The stored information is encrypted and replicated across the participating computers. In Freenet, a file is shared with an ID generated from the hash value of the name and description of the file. Each peer node forms a dynamic routing table to avoid network flooding. A routing table includes a set of other peers associated with the keys they are expected to hold. To search a required file, the query is forwarded to the peer node holding the nearest key to the key requested. If the query is successful, the reply is passed back along the route the query comes in through. Each peer node that forwards the request will cache the reply and update the routing table by a new entry associating the data source with the requested key.

3.3. Hybrid Peer-to-Peer Networks:-

To avoid the detect problems of the centralised and decentralised Peer to Peer(P2P) networks discussed in the above. The hybrid Peer to Peer(P2P) networks are emerging recently to provide trade-off solutions with a hierarchical architecture[6].

3.3.1 Kazaa:-Kazaa reorganises peer nodes into a two-level hierarchy with supernodes and leaves. Supernodes are reliable and capable peer nodes that take more tasks for providing services in the network. A supernode is a temporary index server for all other peer nodes. The peer nodes having high computing power and fast network connection automatically become supernodes. Similarly to the bootstrapping method used in the Gnutella network, a newlyjoined node will attempt to contact an active supernode from a record of super nodes offered by Kazaa client software. The newly joined node will send a list of files it shares to the connected supernodes and further retrieve more active supernodes from the connected supernodes for future connection attempts. In Kazaa, each leave node begins a lookup by sending a lookup request to its connected supernode as shown in Fig.3. Supernode not only checks the local index for the file requested, but also communicates with other supernodes for a list of addresses of peer nodes sharing the files. When a supernode discovers the requested file from its local index, it will act in response to the original supernode. The file is transferred directly among the query originator and the target peer node that shares the file [6]as shown in Fig.5

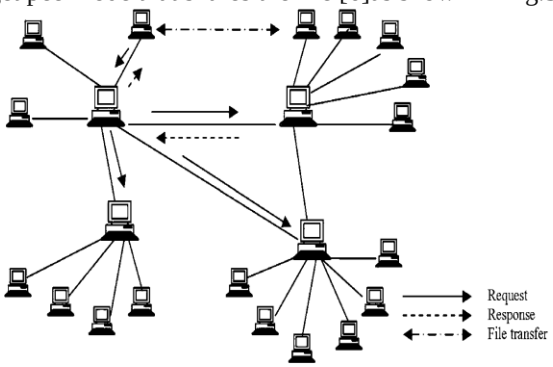

Fig. 5Example of Kazaa network

3.3.2JXTA:JXTA is type of open source Peer to Peer platform developed by Sun Micro-systems. The JXTA API(Application Programming Interface) hides many programming details, which makes a JXTA application writing much easier than developing a Peer-Peer application from scrape. JXTA is similar to the Gnutella and Kazaa and it is maintains to the hierarchical network structure with assemble peers and edge peers. Different from Kazaa, the assemble peers in the JXTA networks call the SRDI (Shared Resource Distributed Index)serviceto distribute indices to other assemble peers within the network. When a peer node start searching for a file, it will send the query to the connected assemble peer and also

DOI 10.17148/IJARCCE.2016.5475

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

multicast the query to other peers on the same subnet. If the assemble peer finds the information about the requested resources on its local cache, it will notify the peers that publish the resources and these peers will respond directly to the query originator. If the rendezvous peer cannot locate the requested information locally, a default algorithm is used to go through a list of assemble peers for a assemblepeer that caches the demanded information [7].

As discussed above, hybrid Peer-Peer networks combine the techniques of both the centralised Napster and the decentralised Gnutella. However, since only a limited number of nodes (peer) are responsible for the query processing and routing, existing hybrid Peer-Peer networks still have the capability bottlenecks of the super nodes, which are also vulnerable to planned attacks [10].

IV. STRUCTURE OF PEER TO PEER NETWORKS

Chord:-Chord [34] is a well-known DHT-based distributed protocol aimed to efficientlylocate the peer node that stores a particular data item. Peer nodes are arranged in aring that keeps the keys ranging from zero to 2m −1. A consistent hashing is usedto assign items to nodes, which provides load balancing and only requires a smallnumber of keys to move when nodes join or leave the network [34]. The consistenthash function assigns each node and each key an ID using SHA-1.In Chord, each peer node maintains a finger table pointing to O (logN) other nodes on the ring. Given a ring with 2m peer nodes, a finger table has a maximum ofm entries. The Chord routing algorithm utilizes the information stored in the fingertable of each node to direct query propagation. For example, a node sends a queryfor a given key k to the closest predecessor of k on the Chord ring according to itsfinger table, and then asks the predecessor for the node it knows whose ID is theclosest to k. By repeating this process, the algorithm can find the peer nodes withIDs closer and closer to k. A lookup only requires O (logN) messages in a N-nodeChord network and 12log2N hops on average [34].Unlike some other P2P models (e.g. Gnutella and JXTA) that provide a set of protocolsto support P2P applications, Chord provides support for just one operation:given a key, it maps the key onto a node. In Chord, peer nodes are automaticallyallowed to participate in the network using the standard Chord protocol, no matterwhether the nodes are useful and capable or not. Chord needs monitoring andselection functions in order to support and optimise its deployment over the realnetworks.

Two-tier Distributed Hash Table (2T-DHT)-
Two-tier DHT overlay is based on chord. It contain of two tier, one is upper tier containingall super peers and other is lower tier containing all normal peers with one super peer. Nodes in the upper tier are more stable nodes means have more uptime and shared resource and the nodes of lower tier are referredas less stable and depict the system architecture of two-tier DHT. In this scheme all ids of normal peers between the range of super node id and super node's predecessor id. This property ensures that the cost for the maintenance of overlay structure is manageable in the event of node join, node leave and node-migration. Note that all super peers maintains two finger table, two successor lists and two predecessor lists for upper tier and lower tier respectively.

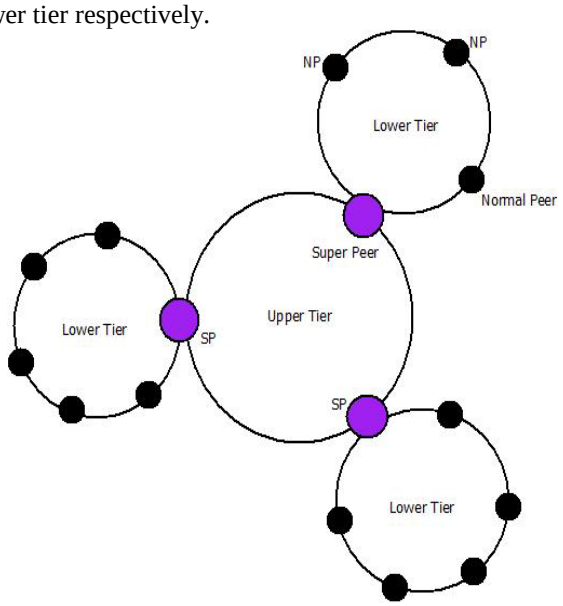

Figure 6: Architecture of 2T-DHT overlay

V. P2P APPLICATIONS

The Peer to Peer applications can be classified into four types-

1. File sharing-Content storage and exchange of the areas where Peer to Peerequipment has been most successful. File sharing applications [9], [11], [11] focus on storing information on and retrieving information from various peers in the networks. The popular example of Peer to Peer system is Napster, it became famous as a music exchange system. Other instances are Gnutella, Freenet, Kazaa, Chord, etc.[13].

2. Collaboration-Collaborative Peer to Peer applications aim to allow application level collaboration between users. These applications range from immediate messaging and chat, to on line games, to shared applications that can be used in business, educational, and home environments. Such as Groove, Jabber. [14] is a traditional of streaming XML(Extensible Mark-up Language) protocol and technology that enable to the entities of Internet to the exchange messages, presences, and other structure information in to the close real time. Groove [14] provides a variety of applications for communication, content sharing (files, images and contact data), and collaboration (i.e. group calendaring, collaborative editing and drawing, and collaborative Web browsing).[13]

3 Distributed computing-These applications use resources from the number of networked computers. The general knowledge behind these applications is that idle

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

cycles from any computer connect to the network can be used for solving the problems of the other computers that require extra computation. SETI@home is one example of the such systems. SETI (Search for Extraterrestrial Intelligence) [16] is a scientific search project aimed at building a huge virtual computer based on theaggregation of the computer power offered from internetconnected computers during their idle periods. The project uses two major components: the database server and the client. Clients can help with search for excess terrestrial life by running the search program for a specified portion of the universe. This project strongly relies on its server to distribute jobs to each participating peer and to collect results after processing is done.[13]

4 Platforms-P2P platforms provide infrastructure to support distributed applications using p2p mechanisms. P2P components used in this context are for instance naming, communication,discovery,resource aggregation andsecurity. JXTA [20] is p2p platform that provides a general-purpose of the network is being programming and distributed computing infrastructure. It creates a Peer to Peer system by identifying a small set of basic functions necessary to support p2p applications and providing them as building blocks for higher-level functions. it includes three layer: core, services and applications. JXTA core provides core support for peer-to-peer services and applications. At the core, capabilities must exist to create and delete peer groups, to advertise them to potential members, to enable others to find them, and tojoin and leave them. At the next layer, the core capabilities can be used to create a established of peer services, including indexing, searching, and file sharing. In the third layer peer applications can be built usingthese facilities[16] [13].

5. Peer to Peer(P2P) Challenges:-
P2P system is an offer the many number of advantages over conventional client-server systems such as fault Tolerance, scalability, performance. However, there are some challenges are:-

Security:-Distributed implementations create additional challenges for security compared to client-server architecture. Since in P2P systems the set of active peers is dynamic and also peers don't trust each other, achievement a high level of security in peer-to-peer systems is more difficult than non-peer-to-peer systems.

Traditional security mechanisms to protect data and systems from intruders and attacks such as firewall can't protect peer-to-peer systems since they are essentially globally distributed and also these mechanisms can inhibit peer-to-peer communication. Therefore new security concepts are required that allows interaction and distributed processing in peer-to- peer systems. [19] [13].
Reliability:- A reliable system is a system that can be recovered whena failure occurs. The factors should be occupied into account for reliability are data replication, node failure detection and recovery, existence of multiple guarantees for location information to avoid a single POF(point of failure) and the availability of multiple paths to data. Data replication increases reliability by increasing redundancy and locality. These are two strategies for replication, owner replication and path replication. In owner replication, when the searchis successfully of the data stored on theclient node only. In path replication, when the search succeeds, data is stored in all nodes besidethe route from requester node to provider node [20]. P2P communities can also replace and replicate the data to achieve adequate performance [21]. In structured P2P overlay networks the messagesis routedin minimum number of nodes. The overlays should modified routing states are automatically when nodes arejoin and leave It should routemessages are correctly even a hugesegmentof nodes the network partitions or crash. To achieve reliability in such systems, nodes essential consume networksbandwidth to maintain routing state, so to reduce this cost the techniques should be employed that adapt to operating condition [17]. For increasing fault-tolerance and reliability in unstructured Peer to Peer systems, dynamically adding terminatedlinks to the systems have been addressed [22] [13].

Flexibility:-Flexibility is the important aspects in Peer to Peer system are the autonomy of peers so that they can join or leave at their will. Recent P2P(Peer to Peer) systems can bedistinguished bytheir decentralized control, extreme and large dynamism in the network. To deal with the scale and dynamism the properties of adaptation and self-organization are required to be considered in building p2p systems. More recent unstructured Peer to Peer systems, like KaZaA and GIA [23] address the dynamic environment. Queries In Kazaa are send only to super nodes, which maintain a list with the file names of their connected peers, avoiding overloading all peers of the system. GIA is a Gnutella like system which aims to respond to highly aggregated query rates. In GIA each peer calculates the maximum number of queries it can handle per second and based on the metric to number of neighbours to which the peer can connect or forward a request is computed [24].In standard structured Peer to Peersystems, Peers are assigned static identifiers and distributed data structures are constructed based on these identifiers, so the overlay network structures are determined throughthe choice of these identifiers and in turn any self-organization of the systems are prevented. Structured systems based on DHTs should perform lookups quickly and consistently while nodes arrive and depart from the system [25], [26]. For instance Chord [27] adapts as nodes join or leavesystem, and respond answer of queries although the system is changing continuously. Self-stabilization protocol run by every node periodically is used to discover joined nodes [28]. Complex Adaptive Systems which areused to describesocialsystems and certain biological behaviour can be used as a model to build adaptive P2P networks [29] [13]

Load Balance:-Data distribution to be warehoused or computations to be carried out by the nodes are critical

DOI 10.17148/IJARCCE.2016.5475

issues for the efficient operation of P2P networks. A particular method for such distribution in P2P systems are the DHT (distributed hash table), in which each data item that is stored is mapped to unique identifier ID. The identifier space is divided among nodes and nodes have the responsibility of storing the data mapped toidentifiers in its portion of the space. In such approaches load balancing should be considered in both address-space balancing key address-space distribution among nodes and item balancing in the case that distribution of data in address-space can't be randomized. In this method, nodes are free to migrate anywhere and it has no restriction to be in a certain number of virtual node locations (it means the items can migrate among the nodes) [30], [31], [32]. Load balancing among the computing nodes in Peer to Peer systems can be implemented by agent-based self organization models. Messor [33] is a Anthill load balancing algorithm. In Messor, their behaviour is adopted by ants onthe load conditions, wandering about randomly when the loads are uniformly balanced, moving rapidly to regions of network with high unbalanced loads. Thereare high tendency of failures if jobs are assigned to crashed nodes are simply reinserted in network by the nest that generated them and they are self-organized as new nests or nodes may join to a system and the computing power is rapidly exploited to carry on the computation, as soon as ants discover the nest and start to assign it jobs transferred from other nests.

VI. CONCLUSION

In this paper, the development of network architecture has been investigated from the client-server to Peer to Peer networking. P2P is helpful when removing the centralised server. On the other hand, new mechanisms are required to compensate for the server. The main advantage of Peer to Peer architecture lies in its good scalability, agility, resilience and availability. On the contrary, its major challenges lie in its efficiency, dependability and security. To address these challenges, hybrid systems combining the techniques of both Grid and P2P computing could be potential solutions for the design of next generation distributed systems. By introducing P2P techniques into Grid computing, Grid systems could be more scalable and resilient, removing the sing-point-of-failure. With the cooperation of Grid computing environments, the usage of Peer to Peer networks could be also broadened from simple file provision to more advanced services, such as sharing redundant computing power for complicated scientific calculation and sharing extra bandwidth for real-time video transmission. Considering different architectures of peer-to-peer systems, system designers should evaluate the requirements for their particular applications and choose a topology for the platform that matches their needs. On comparing, we can say that in pure peer-to-peer networks every peer is given equal responsibility irrespective of its computing/network capabilities, this can lead to reduction of performance as less capable nodes are added. Pure peer to peer(P2P) systems lack manageability since every peer is its own controller. Unstructured pure peer to peer(P2P) systems in which blind flooding search is used are not scalable since in large scale systems the large number of exchange messages limits the scalability. Using structured systems or intelligent search approaches can solve scalability limitation.. It should be the taken that some structured system like Chord have overcome to this problem and they can adapt efficiently as a nodes join or leave the system.Pure P2P systems are fault tolerant, since failure of the any particular node does not impact the rest of the system. Hybrid P2P systems solve the manageability problem of pure Peer to Peer systems, so that the control server/servers act as a monitoring agent for all the other peers and ensures information coherence regarding distributed indexing and centralized indexing systems, drawbacks associated to centralized indexing systems are single point of failures when central server goes down and also not being scalable because of capacity of server to maintain database and to respond to queries. Distributed indexing systems alleviate these shortcomings by using super-peers. Although SP (super peer) clusters are capable, scalable and manageable, in order toavoidof a single point of the failures for the clients in a cluster, some policies of super-peer redundancy should be taken into account. As in the instance of fail over super-peer, these strategies should able to take over the job of the primary super-peer.

REFERENCES

[1]. Waterloo, Canada Princeton,USA Princeton, USA Waterloo, Canada and Xuemin (Sherman) Shen Heather Yu John F. Buford Mursalin Akon March 2009
[2]. Foster, I., Kesselman, C. (eds.): The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann (1999)
[3]. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications 15 (2001) 200–222
[4]. Oberholzer, F., Strumpf, K.: The Effect of File Sharing on Record Sales: An Empirical Analysis. Journal of Political Economy 115 (2006) 1–42
[5]. Risson, J., Moors, T.: Survey of Research towards Robust Peer-to-Peer Networks: Search Methods. Computer Networks 50 (2006) 3485–3521
[6]. XueminShen · Heather Yu · John Buford ·Mursalin Akon.
[7]. Traversat, B., Abdelaziz, M., Pouyoul, E.: Project JXTA: A Loosely-Consistent DHT Rendezvous Walker. Sun Microsystems, Inc (2003)
[8]. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications, Proc ACM SIGCOMM, pp 149-160, 2001, doi: 10.1145/383059.383071
[9]. S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-topeer file sharing technologies, 2002.
[10]. Lo, V., Zhou, D., Liu, Y., Gauthier Dickey, C., Li, J.: Scalable Supernode Selection in Peerto- Peer Overlay Networks. Second International Workshop on Hot Topics in Peer-to-Peer Systems, La Jolla, California (2005)
[11]. Ulrike Lechner. Peer-to-peer beyond file sharing. In IICS, pages 229–249, 2002.
[12]. Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble.A measurement study of peer-to-peer file sharing systems. In Proceedings of Multimedia Computing and Networking 2002(MMCN '02), San Jose, CA, USA, January 2002.
[13]. B. Pourebrahimi K. Bertels S. Vassiliadis Computer Engineering Laboratory, ITS, TU Delft, The Netherland sfbehnaz, koen, stamatisg@ce.et.tudelft.nl

[14]. Jabber, http://www.jabber.org.
[15]. Groove, http://www.groove.net.
[16]. David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: an experiment in public-resource computing. Commun. ACM, 45(11):56–61, 2002.
[17]. RatulMahajan, Miguel Castro, and Antony Rowstron. Controlling the cost of reliability in peer-to-peer overlays.InIPTPS'03, February 2003.
[18]. Ra Ti On. Project jxta: An open, innovative collaboration.
[19]. Dan S.Wallach. A survey of peer-to-peer security issues. In ISSS, pages 42–57, 2002.
[20]. Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. In ICS '02: Proceedings of the 16th international conference on Supercomputing, pages 84–95. ACM Press, 2002.
[21]. JussiKangasharju, Keith W. Ross, and et al. Adaptive content management in structured p2p communities.
[22]. Leonardo Mariani. Fault-tolerant routing for p2p systems with unstructured topolog. Inproceedings of the 2005 International Symposium on Applications and the Internet (SAINT 2005), IEEE Computer Society, February 2005.
[23]. YatinChawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. Makinggnutella-like p2p systems scalable. In SIGCOMM '03: Proceedings of the 2003 conference on Applications,technologies, architectures, and protocols for computercommunications, pages 407–418. ACM Press, 2003.
[24]. EvangeliaKalyvianaki and Ian A. Pratt. Building adaptive peerto-peer systems.InPeer-to-Peer Computing, pages 268 269, 2004.
[25]. Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz.Handling churn in a DHT. In Proceedings of the 2004USENIX Annual Technical Conference (USENIX '04), Boston, Massachusetts, June 2004.
[26]. David Liben-Nowell, Hari Balakrishnan, and David Karger.Analysis of the evolution of peer-to-peer systems. In PODC '02: Proceedings of the twenty-first annual symposium on Principles of distributed computing, pages 233–242. ACM Press, 2002.
[27]. Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In Proceedings of the 2001 ACMSIGCOMM Conference, pages 149–160, 2001.
[28]. C. Tryfonopoulos and M. Koubarakis. Distributed resource sharing using self-organized peer-to-peer networks and languages from information retrieval. In Proceedings of International Workshop on Self-* Properties in Complex Information Systems SELF-STAR), Bertinoro, Italy, 2004.
[29]. [29] Alberto Montresor, Hein Meling, and O¨ zalpBabaog˘lu.Towards adaptive, resilient and self-organizing peer-to-peer systems. Lecture Notes in Computer Science, 2376:300–??, 2002.
[30]. David R. Karger and Matthias Ruhl.Simple efficient load balancing algorithms for peer-to-peer systems. In SPAA '04: Proceedingsof the sixteenth annual ACM symposium on Parallelism in algorithms and architectures, pages 36–43. ACM Press, 2004.
[31]. Yingwu Zhu and Yiming Hu. Efficient, proximity-aware load balancing for structured p2p systems. In P2P '03: Proceedings of the3rd International Conference on Peer-to-Peer Computing, page 220. IEEE Computer Society, 2003.
[32]. AnanthRao, Karthik Lakshminarayanan, Sonesh Surana, Richard M. Karp, and Ion Stoica. Load balancing in structured p2p systems. In IPTPS, pages 68–79, 2003.
[33]. A. Montresor, H. Meling, and A. Montresor. Messor: Load balancing through a swarm of autonomous agents, 2002.
[34]. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A Scalable Peerto- Peer Lookup Service for Internet Applications. ACM SIGCOMM, San Diego, CA (2001)

DOI 10.17148/IJARCCE.2016.5475