Analysis and Comparison of Different Peer to Peer Lookup Protocols

Pragya Singh[1] Prof. Trupti Manik[2]
[1]M.E. Scholar [2]Assistant Professor
[1,2]Department of Information Technology
[1,2]L. D. College Of Engineering Ahmedabad, Gujarat, India

Abstract— This paper revolves around Peer-to-Peer (P2P) lookup systems which implements large scale lookup directory service that operate in dynamic peer-to-peer networks. P2P lookup provide many different Features i.e. Selection of nearby Peer, efficient search/location of data items, redundant storage ,data permanence or guarantees hierarchical naming, authentication and trust and anonymity. P2P also offer an efficient routing architecture i.e massively scalable, self-organizing, and robust in wide-area, combining fault tolerance, load balancing and explicit notion of locality. The main aim of this paper is to Analyze and compare various P2P lookup protocol. The review work includes the classification of various approaches or methods that could be selected for working with P2P look up protocol. It could then serve as the base to start with new research work by upcoming researchers.

Key words: Peer-to-Peer, Distributed Scalable Algorithms, Lookup Protocols

I. Introduction

PEER-TO-PEER (P2P) overlay networks are distributed system, without any hierarchical organization or centralized control. Peers are self-organizing overlay networks that over-layed on IP (Internet Protocol) networks, offering various features i.e efficient search of data items, selection of nearby peers, robust wide-area routing architecture, redundant storage, hierarchical naming, trust and authentication, massive scalability and fault tolerance**.**

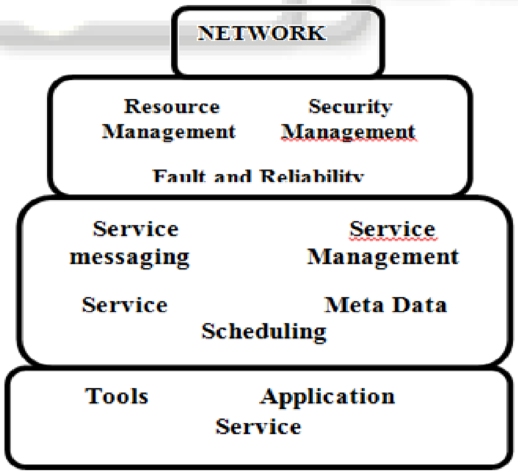

Fig. 1: Abstract P2P Overlay network Figure1 [1]

Figure 1 shows an abstract P2P overlay architecture, explaining all the components in overlay framework. Network Communications layer describes the network characteristics of desktop machines connected over the Internet or small wireless or sensor based devices that are connected in an ad-hoc manner, Nodes Management layer describes management of peers. Features Management layer deals with security, reliability, fault etc. Services Specific layer supports the underlying P2P infrastructure and the application-specific components through scheduling of parallel and computation intensive tasks, content and file management. The Application-level layer is concerned with tools, applications and services.

Basically there are two classes of P2P overlay networks: Structured and Unstructured. Structured P2P overlay network are tightly controlled and content are placed at specified location not at random peers for efficient use. Such Structured P2P systems use Distributed Hash Table (DHT) in which values or data object location information is placed deterministically, at the peers with identifiers corresponding to the data object's unique key. DHT-based systems have a property of consistently assigned uniform random Node IDs to the set of peers. Data objects are assigned unique identifiers called keys, Keys are mapped by overlay network protocol to a unique live peer in network.P2P overlay networks support scalable storage and retrieval of {key, value} pairs on the overlay network. While in Unstructured P2P system peers joining the network with some loose rules, without any prior knowledge of topology. The network uses flooding mechanism to send queries across the overlay with limited scope, as peer receives the flood query, it sends a list of all content matching the query to the originating peer. flooding-based techniques are effective for locating highly replicated items and are resilient to peers joining and leaving the system, they are poor in locating rare items. This approach is not scalable as the load on each peer grows linearly with the total number of queries and the system size. Thus, Unstructured P2P networks face basic problem: peers readily become overloaded, therefore, the system does not scale when handling a high rate of aggregate queries and sudden increase in system size.

II. Lookup Service as P2P System

The lookup services have been built to run on one or several highly available and powerful servers.

Two reasons to design P2P lookup service are:

1) Power of limited number of servers may be in sufficient for demand of certain application that store many objects and execute number of concurrent lookup.
2) Application may not have access to such servers all the may have large number of computing entities that together sufficient power, but individually very limited compared the demand of P2P application.

III. Different P2P Lookup Protocols

A. *Content Addressable Network (Can)* [3]:

The Content Addressable Network [3] is a decentralized distributed P2P infrastructure that shows result in form of hash table functionality on Internet-like scale. A CAN peer maintains a routing table that holds the IP address, virtual coordinate zone of each of its neighbors and the destination coordinates. A peer routes a message to the destination with the help of neighbor coordinates using the hash table, CAN has a routing performance of O (d.N $^{1/d}$) and its routing state is of 2.d bound.

A virtual coordinate space is used to store {K-key, V-value}. Using a uniform hash function, K is mapped to a point P in the coordinate space. Lookup protocol any peer can apply the same deterministic hash function to map K onto point P and then retrieve the corresponding value V from the point P. If the immediate neighbors do not own the point P or the requesting peer then the request should be routed through the CAN infrastructure until it reaches the destination where peer where P lays. A peer contains the IP addresses of peers that contain coordinate zones adjoining its zone. This set of immediate neighbors serves as a coordinate routing table that enables efficient routing between points in this space. A new peer that joins the system is allocated space by splitting existing peer's zone in half.

B. Chord [4]:

Consistent hashing is used by chord [4] to assign keys to its peers. Consistent hashing is designed so that the peers enter and leave the network with minimal interruption. This decentralized scheme helps to balance the overall load on the system. If there are N peers in the system, in a steady state each peer maintains routing state information for about only O (log N) other peers.

By hashing the peer's IP address and the data key is chosen, a key identifier is produced. The identifiers length (m) must be large enough to make the probability of keys hashing to the same identifier negligible. An identifier circle modulo 2m is used to order the Identifiers.

Key k is assigned to 1st peer whose identifier is equal or follows k in the identifier space. This peer is called the successor peer of key k, denoted by successor (k). If identifiers are represented as a circle of numbers from 0 to 2m − 1, then successor (k) is the first peer clockwise from k. The Chord ring is the term given to the identifier circle. When a peer (n) joins a network, to maintain consistence certain key assigned previously to the key n's successor need to be reassigned to n and all the assigned keys to n are reassigned to n's successor when the peer leaves the chord system i.e (log)^2 performance.

C. Tapestry [5]:

Tapestry [5] share similar properties to Pastry, it employs decentralized randomness to achieve both load distribution and routing locality. The keen difference between them is the way they handle network locality and data object replication. Tapestry's architecture uses variant of the Plaxton distributed search technique, with a mechanisms to provide availability, scalability, and adaptability in the presence of failures and attacks. Tapestry also uses multiple roots for each data object to avoid single point of failure. In the Plaxton mesh, peers can take on the roles of servers where data objects are stored, routers to forward messages, and clients' i.e entity of requests.

A peer's local routing map consists of multiple levels, each one of them represents a matching the suffix up to a digit position in the ID space. The nth peer where a message reaches, shares a suffix of length n with the destination ID. To locate the next router, the (n + 1)$_{th}$ level map is examined to locate the entry matching the value of the next digit in the destination ID. This routing method guarantees that any existing unique peer in the system can be located within at most logBN logical hops, in a system with N peers using Node IDs of base B. Since the peer's local routing map assumes that the preceding digits all match the current peer's suffix, the peer needs only to keep a small constant size (B) entry at each route level, yielding a routing map of fixed constant size: (entries/map) x no. of maps = B.logB.

D. Pastry [6]:

Pastry [6] is routing network as similar to chord and used for implementing Distributed hash table (DHT).The {K-key, v-Value} pair are stored in P2P network of connected hosts, and protocol bootstrapped by supplying it IP address of already existing peer in network via routing table that is been dynamically repaired and built, so there is no single point of failure due to its decentralized and redundant nature. Routing metric supplied be external program i.e IP address of target node, can switched to shortest hop count, high bandwidth, low latency. Any node can join and leave the network at any time without warning or with on data loss .To store {Key, Value} in routing table protocol use ping or trace-route etc to find best routes.

Similarly to chord protocol, Pastry hash table's key-space is also taken to be circular, containing different nodes that is positioned with 128-bit node IDs in the circular key-space. These 128-bit node IDs are chosen uniformly and randomly so peers who are adjacent in node ID are geographically diverse. The routing network is been formed on top of hash table by each peer exchanging and discovering state information consisting of a list of leaf nodes, a neighborhood list, and a routing table. The leaf node list consists of the L/2 closest peers by node ID in each direction around the circle, except the leaf nodes there also exist a neighborhood list, represents M closest peers in terms of routing metric. The neighborhood list is used for maintaining locality principals in routing table.

Routing table contains one entry for each address block assigned. To form address blocks, 128-bit key is divided into digits with each digit being b bits long, yielding a numbering system with base 2^b.

This partitions the addresses into different levels from the viewpoint of the client, with level 0 representing a zero-digit common prefix between two addresses, level 1 a one-digit common prefix, and so on. Routing table contains the address of closest known peer for each possible digit at each address level, except for the digit that belongs to the peer itself at that particular level. This results in the storage of 2^b-1 contacts per level, with the number of levels scaling as ($\log_2$N)/b.

E. Kademlia [7]:

Kademlia [7] is decentralized peer-to-peer overlay network. It takes a basic approach to assign each peer a Node ID amongst the 160-bit key space and key, value pairs are stored on peers with IDs close to the key.

A Node ID-based routing algorithm is used to locate peers near the destination key. Kademlia's key architecture is used of the novel XOR metric for distance between points in the key space.

XOR is symmetric and allows the peers to receive lookup queries precisely from the same distribution of peers contained in their routing tables. Kademlia can send queries to any peer within an interval, allowing it to select routes based on latency or send parallel asynchronous queries.

Single routing algorithm is used by it throughout the process to locate peers near a particular ID. Every message to be transmitted by a peer includes its peer ID, which helps the recipient to record the sender peer's existence. Data keys are 160-bit identifiers. It has an unidirectional approach which makes sure that all lookups for the same key converge along the same path, regardless of the originating peer.

Therefore, caching {k, v} pairs along the lookup path. The peer in the network stores a list of {IP address, UDP port, Node ID} triples for peers of distance between 2i and 2i+1 from itself. These lists are called k-buckets. Each k-bucket is kept sorted by least recently accessed peer at the head, most-recently accessed at the tail.

The Kademlia routing protocol consists of:

– PING - check whether Peer active or not.
– STORE - Instructs peer to store {key, value} pair for retrival.
– FIND NODE – take 160 bits ID and return (IP address, UDP port, Node-ID} for K peer knows that are closest to target ID.
– FIND VALUE – it is similar to FIND NODE, it also return { IP address, UDP port, Node-ID} except for the case when a peer received a STORE for the key, it just return the stored value.

Kademlia's peer should locate the k closest peers of some given Node ID.

This lookup initiator firstly picks X peers from its neighboring non-empty k-bucket, then sends parallel asynchronous FIND NODE to the chosen X peers. Now, if the FIND NODE fails to return a peer that is any closer than the closest peers already noticed, the initiator resends the FIND NODE to all of the k closest peers it has not yet queried. It can also route for lower latency as it has the flexibility to choose any one of k peers to forward a request.

F. Viceroy [8]:

Viceroy [8] is again a decentralized P2P overlay network. It is designed to handle the discovery and location of data and resources in a dynamic butterfly fashion. Viceroy employs consistent hashing, to distribute data so that a balance across the set of servers can be maintained and resilient servers to join and leave the network. It utilizes DHT to manage the distribution of data among a changing set of servers and allows peers to contact any server in the network to locate any stored resource by name. It also maintains an architecture that is an approximation as butterfly network.

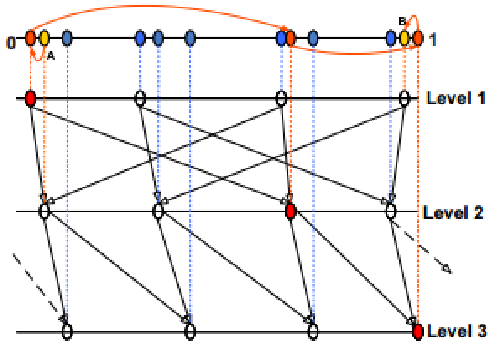

Fig. 2: Simplified Viceroy Network Figure 2[1]

It uses links between successors and predecessors on the rings for short distances. Its overlay diameter better than CAN and its degree is better than Chord, Tapestry and Pastry. When N peers are operational, one of logN levels is selected with near equal probability. Level l peer's two edges are connected to peers at level l + 1. A down-right edge is added to a long-range contact at level l +1 at a distance about 1 2 l away, and a down-left edge at a close distance on the ring to the level l + 1. The up edge to a nearby peer at level l – 1 is included if l > 1. Then, level-ring links are added to the next and previous peers of the same level l. Routing is done by climbing using up connections to a level l − 1 peer. Then proceeds down the levels of the tree using the down links, and moving from level l to level l + 1. It follows either the edge to the nearby down link or the further down link, depending on distance > 1 2 l. This is recursively continues until a peer is reached with no down links, and it is in the vicinity of the target peer.

Therefore, a vicinity lookup is performed using the ring and level-ring links for reliability and fault resiliency and when a peer leaves the overlay network it hands over its key pairs to the successor from the ring pointers and notifies other peers to find a replacement. It is formalized and proved that the routing process requires only O (log N), where N is the number of peers in the network.

G. Georoy [9]:

Georoy algorithm[9] is a location-aware variant of Viceroy algorithm[8]Georoy is used to build overlay network that can provide accurate- efficient resource lookup in an ad hoc wireless network, support either node mobility and resource adding or removing. Using a geographic aware hash function, Georoy is able to obtain small *stretch factor* i.e the ratio between the hop distance of the path traversed by query in order to find the node, number of hops traversed in the physical network from the searching node to the searched one. Stretch factor responsible for measuring the discrepancy between the physical hops traversed during resource lookup and that would have been traveled going directly to the final destination using minimum hop count routing.

As a main difference with Chord and others, Georoy is that it doesn't use a flat node topology, but employs two level hierarchy with 2 different kinds of nodes: *Leaf Peers (LP)* which share and request resources by querying their associated super peers and *Super Peers (SP)* which provide the distributed resource catalog and are used by LPs to publish and request resources.

IV. CONCLUSION

This paper describes the Analysis and Comparison of various schemes in Peer-to-Peer overlay network proposed by different researchers. The comparative study could make researchers use the best possible lookup protocol that could provide efficient routing between peers, efficient search/location of data items etc. in decentralized manner.

COMPARISON OF VARIOUS P2P OVERLAY NETWORK PROTOCOLS							
ALGORIT-HMS	**Architecture**	**Lookup Protocol**	**Routing Performance**	**Peer joins and Leave**	**Routing state**	**System Parameters**	**Fault /reliability**
Viceroy	Butterfly network with rings connected of predecessor and successor links, data is managed by servers	Routing through levels of tree until a peer is reached with no downlinks; vicinity search performed using ring and level ring links.	O(logN)	logN	logN	N -number of peers in network.	Failure of peers will not cause network wide failure. Load incurred by lookups routing evenly distributed among participating lookup servers
CAN	Node ID co-ordinate space, and Multi Dimensional in nature.	key, value pairs to map a point P in the co-ordinate space using uniform hash function	$O(d.N^{1/d})$s	2d	2d	D is the number of Dimension N is the number of peers in network.	On failures, application retries. Peers failure not cause network failure because multiple peers are responsible for each data item.
Pastry	Global mesh network of Plaxton - style.	Match Key and prefix in Node IDs.	$O(\log_B N)$	$\log_B N$	$B\log_B N$+ $B\log_B N$	N- Number of peers are there in network and b-number of bits ($B=2^b$) are used of the chosen peer identifier.	Failure of peers will not cause network failure. Data is replicated on multiple peers. Keep track of multiple paths to each peer.
Tapestry	Global mesh network of Plaxton - style.	Match suffix in Node IDs	$O(\log_B N)$	$\log_B N$	$\log_B N$	N- Number of peers is there in network and B-base of the chosen peer identifier.	Failure of peers will not cause network failure. Data is replicated on multiple peers Keep track of multiple paths to each peer
Chord	Circular Node ID space and unidirectional in nature.	Match node IDs and Key.	O(logN)	$(\log N)^2$	logN	N- Number of peers are there in network.	On failures, application retries. Failure of peers will not cause network failure. Data is replicated on multiple peers
Kademlia	XOR metric for distance between points in the key space.	Matching Key and Node-ID based routing	$O(\log_B N)$ + c where c=small constant	$\log_B N$ + c where c= small constant	$B\log_B N$ +B	N- Number of peers are there in network and b-number of bits($B=2^b$) of nodeIDs.	Failure of peers will not cause network failure. Replicate data across multiple peers.

Table 1: Comparison of Various P2p Overlay Network Protocols

REFERENCES

[1] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma and Steven Lim, a Survey and Comparison of Peer-to-Peer Overlay Network Schemes, IEEE Communications Survey and Tutorial, March 2004 1.

[2] Roger Khazan, Peer-to-Peer Lookup Algorithms CAN, Pastry, and Tapestry Otober23, 2001.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," in Processings of the ACM SIGCOMM, 2001, pp. 161–172, USA, ACIRI AT&T Center for Internet Research at ICSI.

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 17– 32, 2003.

[5] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, pp. 41–53, January 2004.

[6] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in Proceedings of the Middleware, 2001.

[7] [7] P. Maymounkov and D. Mazi` eres, "Kademlia: A peer-to-peer information system based on the xor metric," in Processings of the IPTPS, Cambridge, MA, USA, February 2002, pp. 53–65.

[8] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: a scalable and dynamic emulation of the butterfly," in Processing's of the ACM PODC'02, Monterey, CA, USA, July 2002, pp. 183–192.

[9] Laura Galluccio, Giacomo Morabito,Sergio Palazzo, Marco Pellegrinib, M. Elena Renda, Paolo Santi Georoy: A location-aware enhancement to Viceroy peer-to-peer algorithm.