iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** I **Month of publication:** January 2018

DOI: http://doi.org/10.22214/ijraset.2018.1375

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

Investigation of DHT Based P2P Resource Discovery Algorithms in Grid Environment

Dr. C.Jeyabharathi[1]
[1]PG Department of Computer Science, Arulmigu Palaniandavar Arts College for Women, Palani, Tamilnadu, India

Abstract: Grid Resource Discovery can be defined as searching and locating resources across multiple administrative domains. It returns a list of locations which satisfied the requirements and forms a resource service chain to complete the task. An efficient resource discovery mechanism is one of the fundamental requirements for grid computing systems, as it aids in resource management and scheduling of applications. Resource discovery activity involves searching for the appropriate resource types that match the user's application requirements. Various kinds of solutions to grid resource discovery have been suggested, including centralized and hierarchical information server approaches. This paper, focuses on DHT based Structured Peer-to-Peer Chord protocol and its variations in recent years. The four recently developed algorithms for quick Grid Resource Discovery using P2P Chord Protocol are concentrated in this paper. The results of the new implementations are tested in NSC-SE simulator and performance analysis is given at the end of this paper.
Keywords: Grid Computing, P2P, Resource Discovery, Chord, DHT.

I. INTRODUCTION

Grid computing is a model of distributed computing that uses geographically and administratively disparate resources. In Grid computing, individual users can access computers and data transparently, without considering location, operating system, account administration, and other details. In Grid computing, the details are abstracted, and the resources are virtualized. Resources in Grid system are heterogeneous, geographically distributed, belong to different administrative domains and apply different management policies [1]. The main goal of Grid computing is to enable collaborative and secure resource sharing over multiple organizations which are geographically distributed.

Resource Discovery can be defined as a directory service directed to the spontaneous network's environment [2]. In these networks, resources can enter or leave at any time and this mechanism's proposal is to provide information about the resources available in a specific moment. In a Grid environment, resource discovery is a very complex task basically for two reasons. First, the resources that are potentially interconnected to a Grid are not only computers, but also software, instruments and data, among others. This adds a very high degree of heterogeneity that should be taken into account. Secondly, a Grid can have a very huge number of resources, spread over multiple administrative domains that are geographically distributed. In this scenario, scalability issues must be taken into account. An efficient resource discovery mechanism is the fundamental requirement for Grid computing systems, as it supports resource management and scheduling of applications.

Grid Resource Discovery (GRD) can be classified into Centralized, Hierarchical, Peer-to-Peer (P2P)-based, and group-clustering models from the system architecture aspect. Among the models above, centralized[3],[4] and hierarchical models [5], [6], [7], [8] are easier to design and manage, whereas they prove to be inefficient as the scale of Grid system rapidly increases due to the bottleneck of servers and single point of failure. Therefore, the P2P-based model [9], [10], [11] which combines the P2P technology into Grid is applied widely to overcome those problems. Grid and P2P [12] are the two most common types of resource sharing systems currently in wide use. These two systems evolved from different communities and serve different needs.

In P2P system most resource location queries are not attribute-dependent as in Grids but they either specify a file name, they are keyword searches or range queries. Chord [13], CAN [14], Pastry [15] and Tapstry [16] are the original distributed hash table protocols.

Till recently, researchers have proposed many improved schemes to enhance resource discovery process by reducing the average number of hops, messages, memory consumed and average communication time. There are two main drawbacks of existing Chord Protocol in resource discovery process. One is Information given in node's finger table is not enough to locate the resources quickly and second is Variation between logical address and physical address .ie., node's logical ID is independent of its physical location bringing tremendous delay to network routing. In this paper, these two problems are concentrated and the entire Resource Discovery process is depicted in Fig. 1. This paper focuses on four recently proposed DHD based P2P resource discovery algorithms [17], [18], [19], [20] and eventually they are compared based on different parameters such as number of hops, number of messages per peer and average communication time. The results of the implementations are tested in NSC-SE simulator and performance analysis is given at the end of this paper.

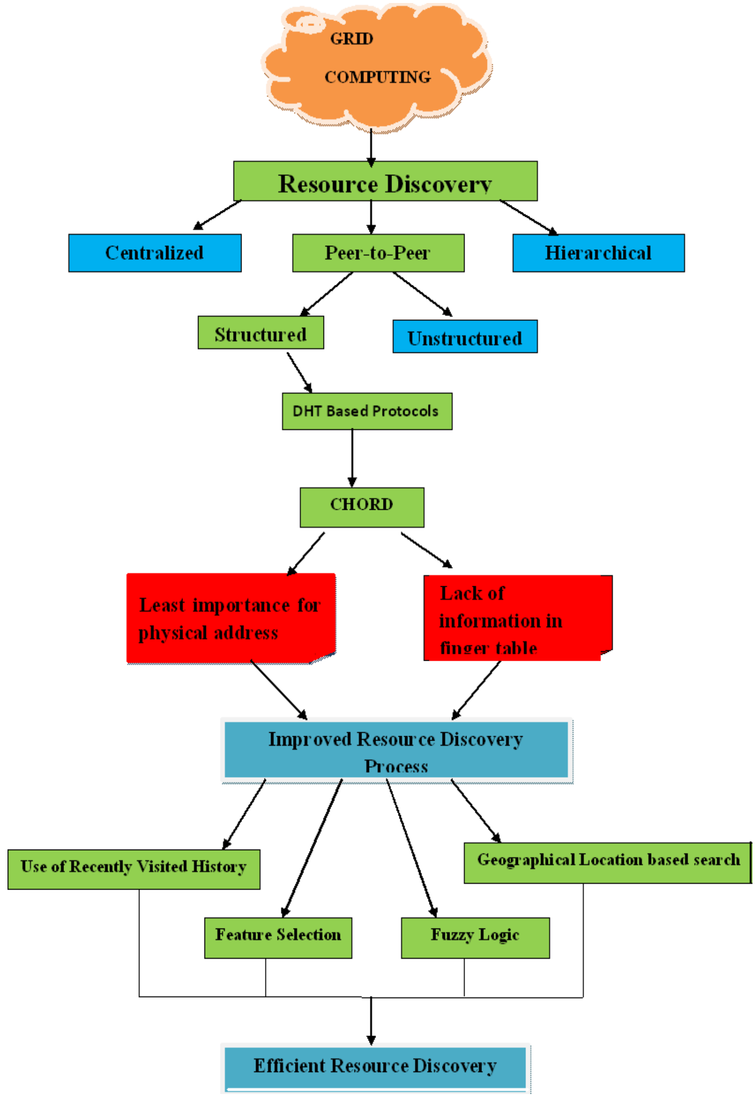


Fig. 1 Work flow of the Proposed Grid Resource Discovery Process

II. DHT BASED P2P RESOURCE DISCOVERY

A peer-to-peer (P2P) [21], [22] computer network relies primarily on the computing power and bandwidth of the network participants rather than concentrating on these in a relatively few servers. P2P networks typically link nodes via largely ad hoc connections. An important fact of peer-to-peer networks is that all clients provide bandwidth, storage space, and computing resources. P2P means a network of peer nodes acting both as servers and clients. The P2P paradigm is based on the principle that every component of the system has the same responsibilities acting simultaneously as a client and as a server, as opposed to the traditional client-server model. P2P systems are divided into two main categories based on the connection protocol they employ and the way peers are organized-structured and unstructured. Structured P2P systems employ a rigid structure to interconnect the peers and to organize the file indices, while in unstructured systems each peer is randomly connected to a fixed number of other peers and there is no information about the location of files. For structured P2P systems, distributed hash tables (DHTs) are widely used. DHT-based systems [23],[13] arrange < attribute; attribute-value >, that is, < key; value > pairs in multiple locations across the network. A query message is forwarded towards the node that is responsible for the key in a limited number of hops. The result is guaranteed, if such a key exists in the system. As compared to the flooding technique, however, DHT-based approaches need intensive maintenance on hash table updates. Chord, CAN, Tapstry and Pastry are the original distributed hash table protocols. This paper discuss about structured P2P approaches in Grid environment.

III.CHORD

All Chord protocol [13] is proposed by Ion Stocia, Robert Morris, David Karger, Frans Kaashoek and Hari Balakrishnan, and was developed at MIT (Massachusetts Institute of Technology), Cambridge. In Chord ring, each machine acting as a Chord server has a unique 160-bit Chord node identifier, produced by a SHA hash of the node's IP address. Chord is a structured P2P routing protocol based on DHT, with the features of fully distributed, scalability, load balancing and stability. A distributed hash table stores key-value pairs by assigning keys to different nodes. A node's identifier is calculated by hashing the node's IP address while a key identifier is produced by hashing the key. In the Chord ring, the keys and the node identifier are arranged on an identifier circle of 2^m where m is the number of bits in the identifiers. The identifiers on the Chord ring are ranged from 0 to 2^m-1. Identifiers are ordered in an identifier circle modulo 2^m.

Each node has a successor and a predecessor. The successor to a node (or key) is denoted by successor (k) and is the next node in the identifier circle in a clock-wise direction. The predecessor is counter-clockwise. In the steady state, in an N-node system, each node maintains information only about O (log N) other nodes, and resolves all lookups via O (log N) messages to other nodes. Key k is assigned to the first node whose identifier is equal to or follows in the identifier space. This node is the successor node of key k, denoted by successor (k). The Chord protocol supports just one operation: given a key, it will store the key's value. Queries are transferred across the Ring via successor pointers and half of the circle is searched every time. Each node maintains a routing table which contains addresses of m successor nodes. When a node joins and leaves the system, maximum number of messages needed is O (log2 N) in most cases. Fingers are arranged in such a way that the distance to the queried ID can at least be halved with every hop. A node's i^{th} finger is the first node succeeding the node ID plus 2i. The i^{th} finger = successor (*node ID* +2i) Lookups are routed clockwise through the network using finger table. The lookup starts with the closest finger and the process is continued till it gets the result.

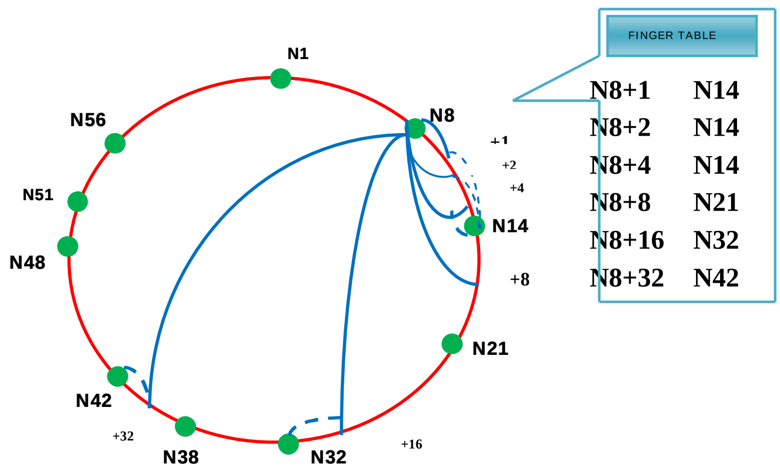


Fig. 2 Chord Ring with 2^6 Nodes

Fig. 2 shows the Chord Ring with 64 Nodes. N1, N8, N14, N21, N32, N38, N42, N48, N51 and N56 are active nodes. Finger table of node N8 is given along with the Chord ring.

One of the most important features of structured P2P protocols is keeping the logical overlay structure. The basic Chord protocol has a simple stabilization mechanism in which each node periodically sends a stabilization message to its successor. If any node leaves the network, it just has to inform both its successor and predecessor to update its neighbour list. Similarly node joining also informs the corresponding neighbors and stabilization is processed.

A. Limitations Of The Existing Methods

Traditionally, the data to be gathered by the resource discovery mechanism is statically defined. In a highly heterogeneous and dynamic environment such as a Grid, statically defined searches are usually inappropriate. Peer-to-Peer approach is a suitable technique for Grid resource discovery because it is a term used to describe the current trend towards utilizing the full resource available within a large distributed network. Drawbacks of existing Chord Protocol in resource discovery processes are inefficient information in node's finger table and variation between logical address and physical address. The next coming sections introduced four approaches on Chord protocol to minimize the limitations of existing methods.

IV. PROPOSED APPROACHES

The This section includes the new approaches related to structured P2P Chord protocol based on Recently Visited Node (RVN-Chord) [17], Feature Selection (FS-Chord) [18], Fuzzy classification (FZ-Chord) [19] and Geographical based location (Geo-Chord) [20].

A. RVN -Chord

In this section, a new form of Chord Protocol, RVN-Chord (Recently Visited Node) is designed which modifies the original Chord's finger table by adding a new column which stores the ID of Recently visited node. Every new lookup uses that ID to find the successor of the key and the search is minimal if the key matches with RVN.ID**.** The primary aim of any protocol is the efficient and fast lookup of nodes containing keys. Generally the performance of Chord like algorithm can be analyzed in terms of three metrics: the size of the finger table of every node, the number of hops a request needs to travel in the worst case, and average number of hops. In the original Chord, when a node requests a key, it has to search its own finger table first and it may find the successor of the key in that table if the match is found. Otherwise, the node sends messages to other nodes whose node id is less than or equal to the searched key and it may need few hops to locate it. After locating the successor of the key, the result is returned to the node that started the search and the lookup process is successfully done. This approach modifies the finger table of the original Chord to add a new entry in the finger table which stores the recently visited node's id.

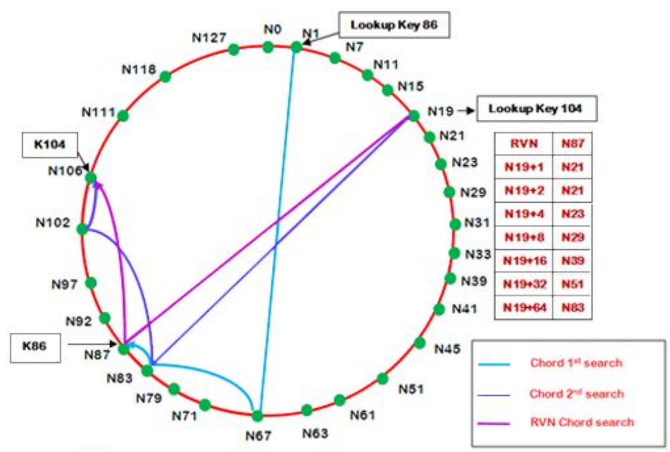


Fig. 3 Lookup comparison between Base Chord and RVN-Chord

If the RVN.id matches the query, the process will succeed and it can directly locate the destination. If it is not exactly matched with the node, next it is checked if the key is greater than the RVN.id. If it is true then the recent route id is in its destination path. Then it can jump to that id and continue its search from that location. Otherwise, the normal Chord algorithm is invoked to find the successor of key. This type of lookup will reduce the number of hops, messages and communication time. Since Memory consumed by the finger table to store the RVN.id is very less in terms of bytes, it will not be a big issue.

Fig. 3 depicts a chord ring whose m is equal to 7 (m is the bit size of the keywords or identifiers) and this Chord network can have a maximum of 128 nodes. The identifier ranges from 0 to 127(2^7-1). Here, RVN Chord is compared with the base Chord algorithm. The number of hops and messages are compared between both models for searching a few keys. Consider node N1 searches key 86 in the Chord ring. In normal Chord, the process of N1 is looking up key K86 is as follows:

N1 ⟶ N67 ⟶ N83 ⟶ N87

It needs 3 hops and 14 messages to find the object node reserving the key N86. The next search can be started by any node and let us consider, the node N19 starts the lookup of key 104. The process looks like

N19 ⟶ N83 ⟶ N102 ⟶ N106

and it needs 3 hops and 13 messages.

Now RVN-Chord is directly applied to find the successor of the key. Assume that the node N19 look for the key 104. When the process is started RVN-Chord verifies its Recently Visited Node id for quick search, because it stored the last visited node in its finger table's first column. As per the previous example, the RVN.id is 87 (as in base chord). Node N19 first checks RVN.id for equality (if 104 = 87) and the match is not found. It then checks condition (N104 > N87) and it is found to be true. So it jumps to N87. Next it searches the finger table of N87 and finds that N104 is the 5^{th} finger of N87 which reserves the key 106. The lookup looks like

N19 ⟶ N87 ⟶ N106

This process requires only 2 hops and 6 messages to locate the key whereas base Chord locates the key in 3 hops and 13 messages**.**

The RVN-Chord algorithm performs well for the lookup of keys whose successor is equal to RVN.id or greater than RVN.id.

This algorithm will work efficiently if the new search needs the recent route. Our simulation result shows that the average number of hops is less than original Chord. This algorithm performs well when the number of nodes is increased greatly.

B. FS-Chord

Till recently, various works have been carried out to improve the lookup performance of Resource discovery process of Chord protocol. Since Chord protocol searches only a single keyword at a time, SHA-1 function is suitable for hashing IP address of nodes and keys maintained by each node. In this section, it is proposed a novel approach that uses feature selection mechanism in DHT based P2P Chord protocol for efficient Grid resource discovery.

The aim of this method is to consider the valuable features of resources of each node based on the individual resource's strength. To extract meaningful features of nodes, frequency of queries posted on a particular resource is calculated. Based on the strength and resource count of each node, nodes are taken for comparison. Apriori algorithm is applied and dominant resource sets are selected among the nodes. Apriori is a classic algorithm for frequent itemset mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger itemsets as long as those itemsets appear sufficiently often in the database. The frequent itemsets determined by Apriori can be used to create association rules which highlight general trends in the database. This has applications in domains such as market basket analysis.

This analysis is made on all the resources of nodes in the Chord ring. Nodes with greater number of weighted resources and nodes with unique resource are finally selected for searching of resources for current discovery process. Based on the selected nodes, the Chord ring looks in new format with minimum active nodes for current search. Initially Chord ring was constructed with 2^m nodes as in original Chord. After that, only selected resources and the nodes which store these resources will be present in the network. The remaining nodes are temporarily not considered for the current searching process. Periodically a new set of nodes are selected to form the active nodes. In the Fig. 4, nodes represented as * are dominant nodes which have dominant attributes. The remaining nodes are non-dominant nodes. Frequently dominant nodes and non-dominant nodes are updated as per the strength of the resources it maintains. The most important thing in this method is that Resource management table (RMT) is built from finger table and along with finger table each node maintains RMT for quick discovery. Since for every query only RMT is referred and dominant nodes are looked for resources, this method can easily locate the resources within minimum hops.

Since the majority of searching is concentrated on a small amount of dominant resources, FS-Chord algorithm is proposed to increase the searching efficiency of dominant resources. The computational method of keyword's searching rate is based on the weighted calculation of resources. To deal with searching frequency of a keyword, Apriori algorithm is applied and frequent item sets and large frequent itemsets are calculated. Node which stores Keywords with more searching frequency will become active nodes in the Ring. Non-dominant nodes are nodes with same type of resources as in dominant nodes with lower capabilities and less in resource count. Nodes with unique resource play an important role and they are identified. Unique nodes will also participate as an active node in the Chord ring.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

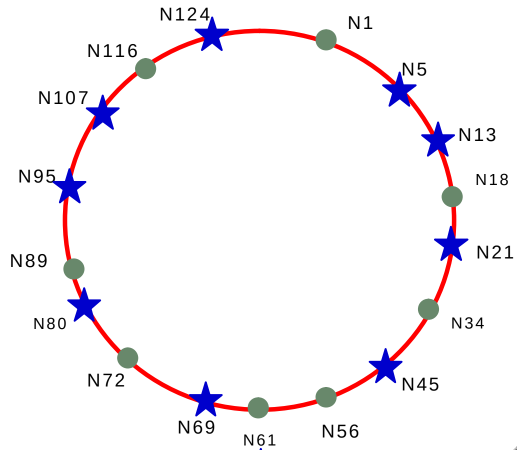


Fig. 4: Chord Ring with dominant ( ) and non-dominant nodes ( )

is maintained by each node which stores meaningful information about resources of dominant nodes. RMT is constructed by including columns such as Node-id, Resource-type, Qty and Node-status. Resources are maintained in a separate array with corresponding resource weight. Periodically Resource Node Weight (RNW) is updated and node with greater RNW is stored at the top of the RMT. Table 1 given below is the structure of Resource Management Table.

TABLE 1
RESOURCE MANAGEMENT TABLE

Node ID	RNW (in order)	No. of Resources	Resource 1	Resource 2	………	Resource N	Node Reference ID

Resources are prioritized based on the requirement. The typical priority structure and node value are calculated as in Fig.5. Herein the following lists of sample resources are considered and prioritized.

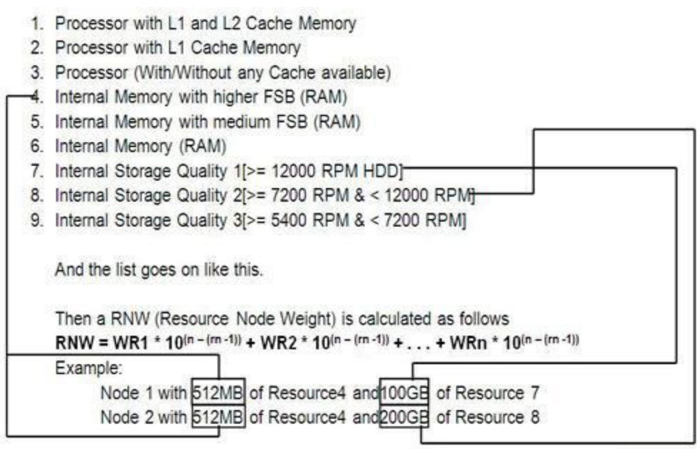


Fig. 5 An example for Resource Selection Process

The formula given in Fig. 5 helps to calculate Resource Node Weight, where n is the total number of resources and *rn* is the particular resource number.

In a single cycle all nodes are checked with its resource status and the nodes with more than 60% of resources are extracted and the details are stored in RMT. For simulation purpose, nodes with more than 60% of resources are considered.

Simulation results show that this method discovers resources with a small number of hops and messages within a short time. This method is compared with the base Chord protocol and our previously proposed method, RVN-Chord. The results of the methods are compared in terms of Number of messages, Number of hops, average communication time and memory consumed.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

C. FZ-Chord

In this method we apply Fuzzy classification which classifies elements into a fuzzy set and its membership function is defined by the truth value of a fuzzy propositional function. The Chord is constructed initially with 2^m nodes. The ring is divided into three rings according to the basic Fuzzy- rule. Nodes with more than 66% of resources are grouped in HOTTEST RING. Nodes with resources between 34 to 65% are grouped into HOTTER RING and nodes with less than 34% resources are allocated to the HOT RING. The goal of this method is to create a model which includes all the necessary conditions to clearly differentiate each ring. Whenever a new node enters, the algorithm correctly predicts the corresponding ring and the node will join into the desired place. In this place Divide-and-Conquer learning concept is applied to split the nodes into subsets and this process is recursively executed for the subset of nodes.

Fuzzy numbers are fuzzy subsets of the real line. They have a peak or plateau with membership grade 1, over which the members of the universe are completely in the set. As in Fig. 6 the membership function is increasing towards the peak and decreasing away from it. The following figure depicts the fuzzy membership function for three groups of nodes.

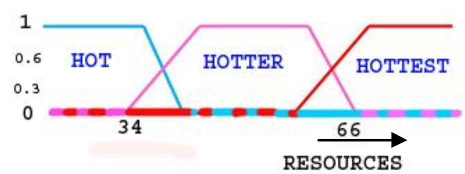


Fig. 6 Fuzzy Membership Function for Three Rings

Now the 2^m nodes are grouped into three rings according to the classification procedure of decision tree algorithm with the help of basic fuzzy logic properties. The splitting is based on the number of resources each node has. In each ring, Ring-Head (marked as * in Fig.7) is selected based on the strength of the resources. Ring-Head is the entry point to each ring. If a query is started by the node of any ring, the Ring-Head checks its own ring and simultaneously sends the query to the Ring-Head of the remaining two Ring-Head. So the resources are simultaneously searched in all the three rings. The lookup process may successfully end in the ring which originates the query or any one of the two rings may find the resource. If the resource is located in the originated ring, the process is similar to the normal lookup process. Otherwise, the node which stores the key will return the result to its Ring-Head and in-turn the Ring-Head submits the result to the corresponding node via the head of the corresponding ring. As far as the lookup process is concerned, this method will find the key through parallel search and reduces the communication time dramatically. We named this method as FZ-Chord.

After splitting the rings, Resource Table is constructed by getting details of nodes from the finger table of each node. The entries of the table are node.id, number of resources and node status. As shown in Table 2, Node status stores the status of resources of each node.id.

TABLE 2
THE STRUCTURE OF RESOURCE TABLE

Node.id	No. of Resources	Node Status

Entropy is used at the time of creating Resource Table to avoid duplicate entries of resources. If more than one node maintains the same set of resources, it will be handled by entropy method to put single entry about those resources. For example, if nodes N2, N4, N9 have RAM with 1GHZ speed, Resource Table will have a single entry for these three nodes. Entropy finds this redundancy and makes the Resource Table meaningful.

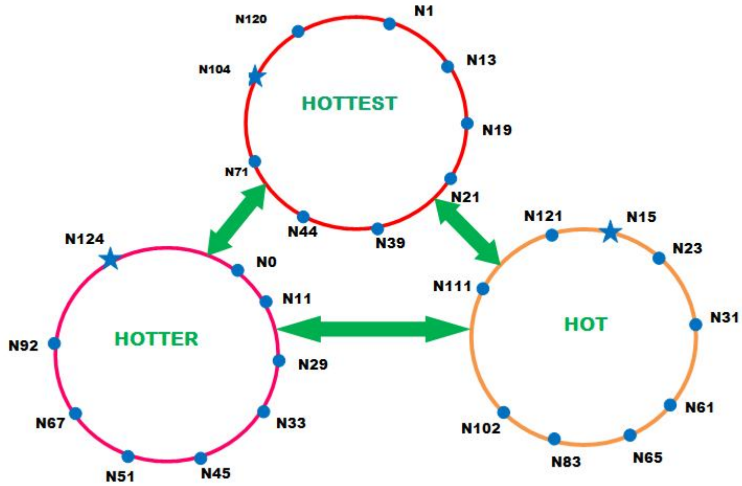


Fig. 7 Three Chord Rings after Fuzzy classification

Chord Ring with 2^7 (128) nodes is divided into three Rings as shown in Fig. 7. An example is given below for clear understanding of this logic.

Node 31 searches the key 75 which is in the HOT RING, the lookup process follows the following steps.

Step 1: Node 31 searches its own Resource Table and simultaneously it sends the request to its Ring-Head.

Step 2: Now Ring-Head of HOTTEST RING sends request to the Ring-Heads of Hotter Ring and Hot Ring.

Step 3: The search process is simultaneously going on with the help of Ring-Heads.

Step 4: Key 75 is located in the HOT RING and the corresponding Ring-Head sends the result to Node 31 through the Ring-Head of HOTTEST RING.

Chord is a good choice for single keyword search, we experiment Chord by applying fuzzy classification to split it into three rings and search simultaneously all the rings for quick discovery. This method reduces number of hops, number of message and communication time to search a resource.

D. Geo - CHORD

Among structured P2P system Chord is a well-known protocol due to its simplicity, dynamicity, scalability and flexibility in node join and departure. Despite these favourable characteristics, the Chord protocol has the disadvantages such as that nodes in the network do not consider physical location of nodes that are in their geographic range; i.e., each node directs its lookup requests based on routing table (or finger table) rather than first searching for the desired data in the nodes that are in its geographic range. This property increases the lookup time and wastes the network bandwidth. Hence, this problem is taken into account and a new functionality is added to the Chord protocol that considers physical location of nodes present in the network. This new functionality increases the performance of the search procedure and this form of Chord is named as Geo-Chord.

Geo-Chord is based on geographical location of nodes. Based on the Euclidian distance among nodes, neighbors are identified and many Region-Rings are formed. Ring heads of Regions form the main Chord ring. Geo-Chord makes use of the physical network topology of the overlay network to demonstrate more efficient key lookup.

In Geo-Chord, each node is given a unique position ID which contains the data about longitude and latitude. In real-time grid environment, the node position is sent to the node by recorded location from the network service provider. In case of combination of wired and wireless network grids, position of a node is calculated by service-providers using tri-tower location finding or tri-satellite location finding procedures. This procedure is the same as used in the mobile GPS calculations. Euclidean distance is the distance measurement calculated for distance among nodes in the Region. Nodes are grouped according to their physical Region. After grouping the nodes according to their physical location, sub-rings are formed based on Euclidean distance among nodes as Fig. 8. When a node joins the overlay, Geo-Chord introduces other nodes within the same Region to this node by binding it to the Region ring. Moreover, each node is responsible for maintaining another finger table which is related to the Region ring nodes. This table, which is named as Region finger table, is identical to the original finger table in terms of formatting and the completion procedure, except that this table deals only with Region ring nodes. The aim of this table is to maintain the information of proximate nodes (the nodes within same Region) and therefore lets the lookup process to be done more efficiently.

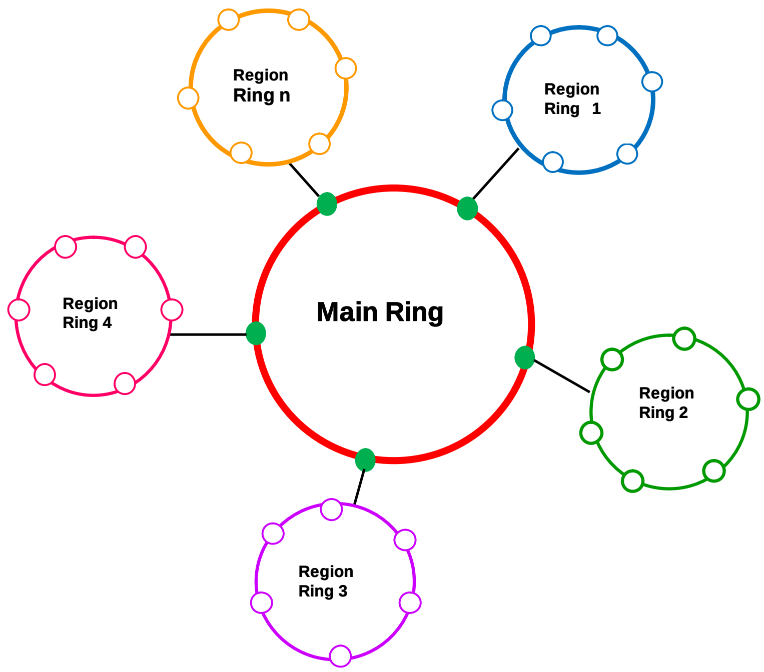


Fig. 8 Chord with Multiple Region Ring

Region-length finger table is used in this application for memory optimization. The finger table has the basic information along with additional physical address details. Nodes are grouped based on physical location, so that nodes which are physically nearer are grouped in the same Region. Region heads form the chord and have links with other nodes through heads.

Ant Colony Optimization (ACO) procedure is applied for search process which quickly locates the keys within a shortest distance. I the Region finger table, Region code is stored along with the IP address. When the entire grid is free, the key is searched in the same Region where the originator node is present. If the Cluster-head of the super ring is busy, the successor of the node will receive the query. This query is forwarded to the neighbouring Region. If the entire grid is busy, the query has to wait till any of the nodes in grid is free. Search process by any node can first check the Region code and locate the corresponding Region head. Then the process will continue until the desired key is located.

From the simulation results, it is identified that the Geo-Chord algorithm effectively reduces the required number of hops, messages and communication time.

V. PERFORMANCE EVALUATION

Figures 9, 10 and 11 respectively show the average number of messages, average number of hops and average communication time for base Chord, RVN-Chord, FS-Chord, FZ-Chord and Geo-Chord with changes of total number of nodes in P2P systems. The average number of messages and hops taken by the Geo-Chord is the same as the FS-Chord and FZ-Chord but Geo-Chord reduces the communication time dramatically than Chord and all previously proposed algorithms by us. All the algorithms are developed in VC++ and tested in simulation environment. From the simulation result, it is concluded that when compared to the previously stated algorithms, Geo-Chord performs well in terms of communication time.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

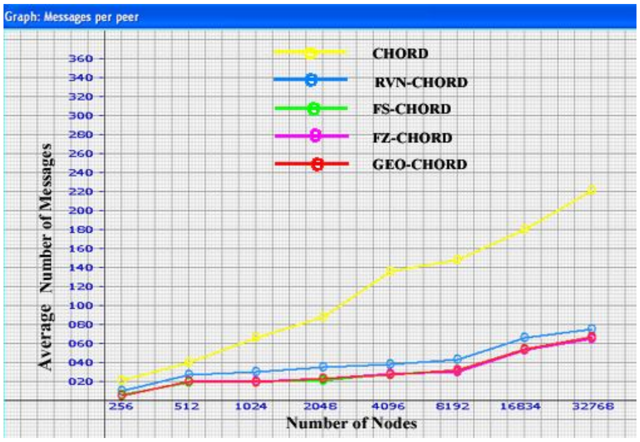


Fig. 9 Number of Messages per peer

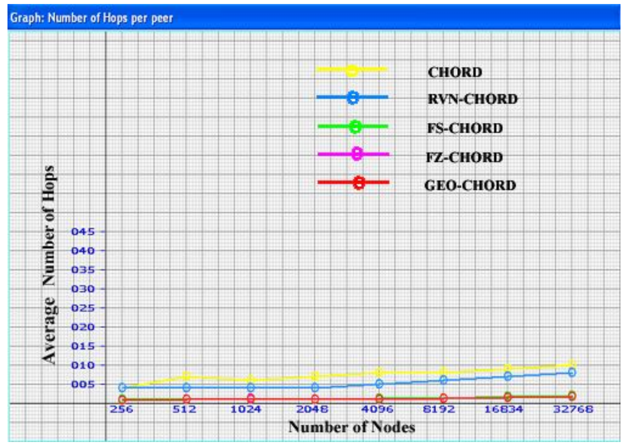


Fig. 10 Number of hops per peer

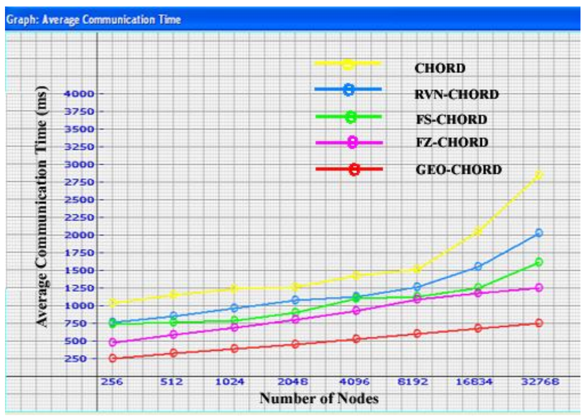


Fig. 11 Average Communication time

VI. CONCLUSIONS

The hierarchical and centralized resource discovery schemes are suitable for small size grids, while P2P schemes are suitable for dynamic and large scale grid systems in which reliability of queries is not important. This paper focuses on the Chord P2P Structured protocol and various improvements of the Chord over its finger table as well as it incorporate new technologies on Chord. It mainly compares and analyse the four recently proposed DHD based P2P resource discovery algorithms based on different parameters such as number of hops, number of messages per peer and average communication time. The results of the implementations are tested in NSC-SE simulator and from performance analysis we concluded that Geo-Chord performs best in terms of communication time.

REFERENCES

[1] Foster, I. and C. Kesselman, "The grid: blueprint for a new computing infrastructure", Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1999
[2] G. Coulouris, J. Dollimore, T. Kindberg, "Distributed Systems: Concepts and Design", Addison-Wesley, 772p, 2001.
[3] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke, "A directory service for configuring high-performance distributed computations", preprint, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1997.
[4] M.Riedel, B.Schuller, D.Mallmann, R.Menday, A.Streit, and B. Tweddell, et al., "Web services interfaces and open standards integration into the European unicore 6 Grid middleware", in the Eleventh IEEE International EDOC Conference Workshop, Annapolis, Maryland, USA, pp.57-60,October 2007.
[5] R.Chang and M.Hu, "A resource discovery tree using bitmap for Grids", Future Genertion Computer System, vol.26, issue 1, pp.29-37, January 2010.
[6] L.M.Khanli and S.Ebadi, "A new distributed and hierarchical mechanism for service discovery in Grid environment", in the Fifth International Conference on Advances in Grid and Pervasive Computing, Hualien, Taiwan, pp.174-82, May 2010.
[7] L.M.Khanli and S.Kargar, "FRDT:Footprint resource discovery tree for Grids", Future Generation Computer System, vol. 27, issue 1, pp.29-37, January 2010.
[8] L.M.Khanli, A.K.Niari and S.Kargar, "Efficient method for multiple resource discoveries in Grid environment", In Proceedings of the International Conference on High Performance Computing and Simulation, HPCS 2011, Washington, DC, USA, pp.128-34, August 2011.
[9] Y.Fu, R.Hu, J.Chen, Z.Wang and G.Tian, "An improved lookup algorithm on over- DHT paradigm based P2P network", Lecture Notes in Computer Science, vol 6987, pp 200-7, September 2011.

[10] Carlo Mastroianni, Domenico Talia, and Oreste Verta, "A super-peer model for resource discovery services in large-scale Grids", Future Generation Computer Systems, 2005.

[11] Stefan Schmid and Roger Wattenhofer, "Structuring Unstructured Peer-to-Peer Networks", Eds: HiPC 2007, LNCS 4873, pp. 432–442, Springer-Verlag Berlin Heidelberg 2007.

[12] Z.Xiong and X.Zhang, "Resource management architecture and genetic ant algorithm for P2P Grid system", Journal of Computational Information Systems, vol 6, issue 1, pp 79-88, 2011.

[13] Ion Stoica, Robert Morris, I.Stoica, D.L.Nowell, D.R.Kargar, M.F.Kaashoek,, H.Balakrishnan "Chord: A Scalable Peer-to-Peer lookup service for internet applications," IEEE/ACM Transactions on Networking, vol 11, issue 1, pp 17-32, Feb 2003. (First appeared in Proc. ACM SIGCOMM conference on Applications, Technologies, Architectures, and protocols for Computer Communication, 2001, pp. 149-160).

[14] S.Ratnasamy, P.Francis, M.Handley, R.Karp, S.Shenker, "A scalable content-addressable network," in Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, California, pp 161-72, Aug 2001.

[15] A.Rowstron, P.Druschel, "Pastry: Scalable, decentralized object location and routing for large scale peer-to-peer systems," in Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, London, pp 329-50, Nov 2001.

[16] B.Y.Zhao, J.D.Kubiatowicz, and A.D.Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", Technical Report. UCB//CSD-01-1141,UC Berkeley, April 2001.

[17] A.Pethalakshmi and C.Jeyabharathi, "RVN-Chord: A Novel P2P Lookup Algorithm for Dynamic Grid", International Journal of Computational Intelligence and Informatics, Volume 2 Number 4 March, 2013.

[18] C.Jeyabharathi, **"**An Improved Mechanism for Resource Discovery in A Large Scale Grid through DHT Based P2P Protocol**",** International Journal of New Technologies in Science and Engineering Vol. 4, Issue. 9, September 2017.

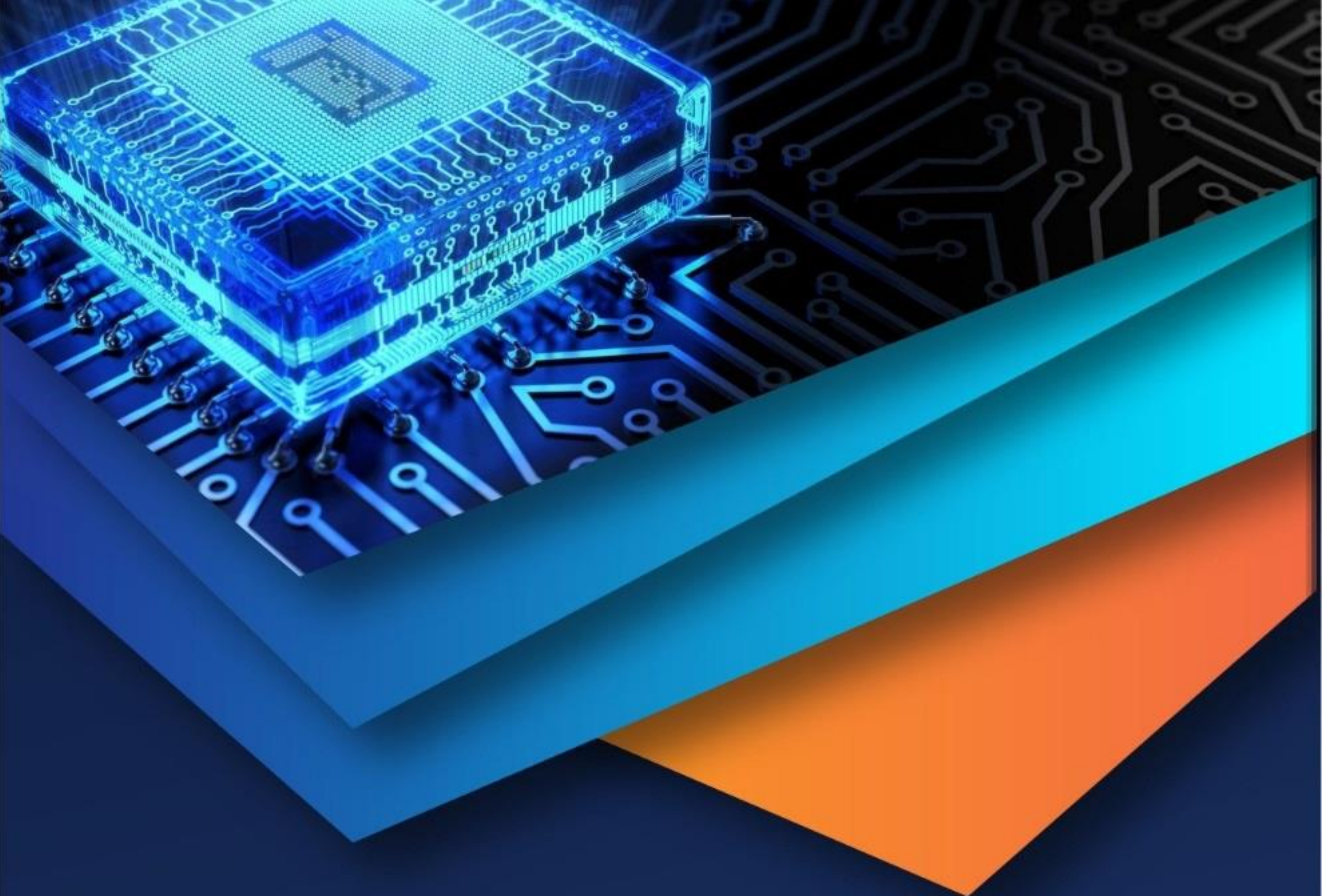
[19] A.Pethalakshmi and C.Jeyabharathi, "Parallel Search in Structured Chord Protocol for Quick Resource Discovery in Grid Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 12, December 2013.

[20] A.Pethalakshmi and C.Jeyabharathi, "GEO-Chord: Geographical Location Based Chord Protocol in Grid Computing", International Journal of Computer Applications, Volume 94 – No 3, May 2014.

[21] M. Marzolla, M. Mordacchini and S. Orlando, "Peer-to-peer systems for discovering resources in a dynamic Grid", parallel Computing 33, 339–358, 2007.

[22] H. Stockinger, Asad Samar, Bill Allcock, Ian Foster, Koen Holtman, and Brain Tierney, "File and Object Replication in Data Grids", Journal of Cluster Computing, 5(3), pp. 305-314, 2002.

[23] M. Frans Kasshoek and D. R. Karger, "Koorde: A Simple Degree-optimal Distributed Hash Table", In Proceedings of second International Workshop on Peer-to-Peer Systems (IPTPS'03), Berkeley, USA, February 2003.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY