

Geo-Chord: Geographical Location based Chord Protocol in Grid Computing

A.Pethalakshmi
Associate Professor and Head
Dept. of Computer Science
M.V.M Govt. Arts College (W)
Dindigul

C. Jeyabharathi, Ph.D
Research Scholar Manonmaniam Sundaranar
University,
Tirunelveli

ABSTRACT

Grid computing is a service for sharing computer resources and data storage capacity over the internet. As resource requirements of recent applications increased greatly, grid systems have gained importance in the last decade. Resource discovery is the essential job in Grid computing which provides searching and identifying necessary resources for given processes. There are various Resource Discovery mechanisms in which Peer-to-Peer (P2P) technology witnessed a rapid development. Current Peer-to-Peer system is based on DHTs struggle with routing locality because of random node ID assignment. Locality awareness is one of the important properties of P2P Systems, which builds and operates their topology independently in the underlying network topology. Among structured P2P system Chord is well-known protocol due to its simplicity, dynamicity, scalability and flexibility in node join and departure. Due to the use of logical overlay network, Chord suffered from high routing latency and low efficiency in data lookup. In this paper, Geo-Chord is introduced, which 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. Simulation results show that our proposed algorithm performs better in terms of average number of messages, hops and average communication time.

Keywords

Grid Resource Discovery, Peer-to-Peer, Physical-location of node, Region-Ring, Geographical location based Chord, Ant-Colony Optimization.

1. INTRODUCTION

Complex and large-scale problems in science, engineering and business need more powerful computing machines because they are more computing-intensive and data-intensive. Grids are geographically distributed platforms with heterogeneous resources in which users can access via a single interface. Grid provides common resource-access technology and operational services across widely distributed and dynamic virtual organizations. The nature of Grid infrastructure is to integrate large computational and storage resources, data,

services and applications from different disciplines [1]. The ability to manage Grid systems depends on the accuracy and availability of the resources like computational power, storage and networking [2]. Locations of resources in Grid environment is the primary necessity which permits the discovery of services across multiple administrative domains. Resource discovery is the process of locating proper resource candidates which are suitable for executing jobs within a reasonable time. The characteristics of the Grid systems make the resource discovery a time consuming process which can decrease the performance of the whole system. Resource discovery in Grid is a challenging issue because characteristics of its resources are heterogeneous, dynamic, various and autonomous.

Grid middleware provides basic services for resource discovery, data management, security and communication. Globus toolkit can be considered as the most extended core middleware for Grid computing environments [3], [4]. Globus defines and implements a set of protocols, APIs and services used by hundreds of Grid applications all over the world. Moreover, Globus worked as a pioneer in interoperable Grid systems development. To be an attractive choice, Grid computing must provide adequate mechanisms for resource management to huge heterogeneous environments spread over multiple organizations that are geographically distributed [5]. One of the most important components of a resource management system for Grid is resource discovery. Various methods have been proposed to solve the resource discovery problem in Grid systems. Grid resource discovery process uses different classes of systems like centralized, hierarchical systems, agent-based systems and Peer-to-Peer systems. Due to central points of administration, Centralized systems and hierarchical systems can make the whole system unusable. Peer-to-Peer (P2P) techniques have been recently exploited to achieve efficient Resource discovery. Peer-to-Peer (P2P) systems are distributed systems in which there are neither central control points, nor a hierarchical structure among its members. In a P2P system, all nodes in the system have equal role and are interconnected through internet. The benefits of P2P systems are adaptation, self-organization, fault-tolerance and load balancing. P2P systems, initially designed for file sharing (Napstar [6], Gnutella [7]) are gaining popularity in

industries. According to [8], [9] we can classify P2P systems into two basic types of overlays: Structured and Unstructured.

The technical meaning of Structured is that the P2P overlay network topology is tightly controlled and contents are placed not at random peers but at specified locations that will make subsequent queries more efficient. Such structured P2P systems use the Distributed Hash Table (DHT) as a substrate, in which data object (or value) location information is placed deterministically. DHTs have become a common mechanism to safely share information in P2P systems. DHT-based systems [10]-[12] arrange $\langle \text{key}; \text{value} \rangle$ 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 unstructured systems, DHT-based approaches need intensive maintenance on hash table updates. Unstructured architecture is attractive because of their simplicity and their high robustness.

In unstructured systems, there is neither a centralized directory nor any control over the network topology or resource placement. When a new peer joins the P2P network, it forms connections with other peers freely, e.g., it selects arbitrary peers as neighbours. In order to publish its resources, a peer usually just stores them locally or places them on randomly chosen peers. Generally, unstructured overlays have loose guarantees for resource discovery, and it is possible that a file is not found although it exists in the network. There are many improved schemes to enhance routing efficiency of Chord Protocol based on geographical location such as TB-Chord [13], ML-Chord [14], CLT-Chord [15] and Quasi-Chord [16]. In this paper, a new mechanism of locating resources in DHT based P2P Chord protocol is proposed with the help of Geographical location of nodes in the P2P network. The main contribution of this paper is introduction of Geo-Chord (Geographical location based Chord) algorithm. Geo-Chord is a modified version of Chord which considers physical network topology through dividing the geographical space in which the nodes are distributed, into smaller Regions based on their physical location. These smaller Regions form a sub-ring (Region-Ring). This small ring of a Region lets the nodes inside that Region to become aware of each others existence. In order to keep more information, Geo-Chord allows every node to be familiar with other proximate nodes and do a better routing by selecting the next hop more appropriately. ACO algorithm is applied in the Region rings to quickly find the resources by using the shortest path. As a result, the average distance that must be traversed by each message is significantly reduced, so that the communication time to locate a key is highly reduced. Moreover, less bandwidth is required in average to route a query from its source to the destination. In other words, Geo-Chord exploits underlying network topology information to perform better message routing. Euclidean distance is the distance measurement calculated for distance among nodes in the Region.

The rest of this paper is organized as follows. Section 2 discusses the various Resource Discovery models, section 3

explains the common characteristics of Grid and P2P, section 4 describes base Chord Protocol and its Working process, section 5 lists and discusses various existing physical location based resource discovery methods using Chord protocol, section 6 discusses the basics of Ant Colony Optimization algorithm, section 7 proposes our new implementation for meaningful resource discovery, section 8 and 9 analysis the performance of the simulation results and section 10 concludes our work.

2. GRID RESOURCE DISCOVERY

A Grid Resource Discovery can be defined as searching and locating resources across multiple administrative domains. A Grid resource discovery mechanism returns a list of locations which satisfied the requirements and forms a resource service chain to complete the task. Grid resources are characterized into two types namely static and dynamic. Static attributes refer to resource characteristics that do not change frequently, such as Operating System, CPU speed of a computing resource. Dynamic attributes are associated with fast changing characteristics, such as CPU load and free memory.

Grid resource discovery process uses different classes of systems like centralized, hierarchical and agent based systems. Even though the first two methods have the advantages of Open Grid Service Architecture (OGSA), they suffer from scalability, reliability and false-positive problems. But agent based systems are attractive in Grid systems because of their autonomy properties. They have capabilities to determine new migration sites according to their migration policies for the distribution of resource discovery queries, so that researchers adopted Peer-to-Peer (P2P) technology in Grid environment to solve these problems.

P2P systems are mainly divided into Structured P2P networks, Unstructured P2P networks and Super-Peer systems. In Structured P2P system, the network is formed with a rigid structure and each peer in network can be linked with a fixed number of other peers [17], [18]. Unstructured network is formed by randomly joining peers and it has no fixed strategy. Unstructured P2P resource discovery approaches handle the dynamicity of resources. In super-peer based methods, flooding [19] mechanism is used which leads to single point of failure. New techniques like Genetic algorithm [20] and Fuzzy logic [21] are currently implemented in resource discovery process for efficient resource discovery.

3. GRID AND P2P

Nowadays P2P systems have been the hottest research topic in a large distributed system. Since P2P based network approach may overcome the limitations of hierarchical and centralized methods, P2P techniques are especially used in resource discovery process. The self-organization, scalability and dynamicity are the inspiring features of P2P systems. These features help P2P networks for achieving High throughput computing (HTC) [22]. An HTC system can be defined as a platform to execute a large number of jobs per unit of time. These jobs are independent, meaning that they can be submitted by different users, and the jobs submitted to the system can be executed in any order.

Grid and P2P are both resources sharing systems and harnessing of resources across multiple administrative domains is their main goal. The two main characteristics of these two systems are dynamic behavior and heterogeneity of the involved components. Even though the goals of Grid and P2P are the same, they use different approaches towards resource discovery. The first thing is that P2P systems are designed to share files among peers but Grids deal with complicated set of resources. Another difference is that the dynamism of P2P systems comes from both nodes and resources but in Grids, nodes are connected to a network in a relatively static manner and the dynamism comes from the fast-changing status of resources. Due to the techniques used in Grid and P2P systems, both are mutually benefited. The researchers argued that Grid and P2P systems will eventually converge [23].

The complementary nature of the strengths and weaknesses of the two approaches suggests that the interests of the two communities are likely to grow closer over time [24]. Because of the scalability nature of Grid systems, centralized management will become inefficient and other mechanisms are considered. P2P systems will open up to more sophisticated applications and they will have to support more complex queries and different Qos levels. An era of group based resource-aware communications is opened by P2P systems for many new applications and services. New P2P service scenario can be envisioned by adding together location information, presence information in the end-to-end communication path and local resources. So the market is driving P2P services based on IP into wireless mobile devices. VoIP, Messaging, file sharing, navigation systems and multi-party video conferencing are a few examples of such services. Associated P2P and Grid applications such as mobile Grid for access to Grid resources or distribute/store shared information across Virtual Organizations (VOs), such as web, digital cameras, own digital community-TV broadcast, music, etc., will also be possible in future. Nowadays, the research and development community agrees that the adoption of the Peer-to-Peer paradigm could favour Grid scalability [25]. Grids and P2P systems share several features and can profitably be integrated, bringing benefits to both the fields resulting in future convergence. The introduction of the web services resource framework is particularly useful for this convergence trend, since WSRC Grid services can support P2P interactions between hosts belonging to different domains.

4. CHORD

Chord, CAN, Tapstry and Pastry are the original distributed hash table protocols. Chord protocol is proposed by Ion Stocia, Robert Morris, David Karger, Frans Kaashoek and Hari Balakrishnan, and was developed at MIT. 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 will store the values for all the keys which are responsible for keys

and nodes on behalf of the resource information are mapped to a virtual space. 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 everytime. Each node maintains a routing table which contains address of m successor nodes. When a node joins and leaves the system, maximum number of messages needed are $O(\log^2 N)$ in most cases. Fingers are arranged in such a way that the distance to 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 2^i . i th finger = successor ($node\ ID + 2^i$) Lookups are routed clockwise through the network using finger table. The lookup starts with the closest finger and process is continued till it gets the result.

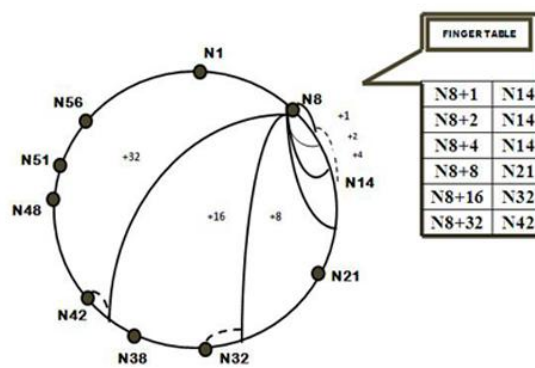


Fig 1: Chord Ring with 127 Nodes

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 their neighbor list. Similarly node joining also informs the corresponding neighbors and stabilization is processed.

Fig. 1 shows an identifier circle with $m=6$. The circle has seventeen active nodes and the identifier ranges from 0 to

127. In this example, the successor of identifier 1 is node N1, so the key 1 would be located at node N1. In the same way, key 20 is located at node N21, and key 41 at node N42. Consistent hashing is designed to let nodes enter and leave the network with minimal disruption. To maintain the consistent hashing mapping, when a node n joins the network, certain keys previously assigned to n 's successor now are assigned to n . When node n leaves the network, all of its assigned keys are reassigned to n 's successor. As Chord uses hashing function, it is well suited for single keyword search. In this paper, physical location of nodes in P2P networks is considered for resource discovery.

5. PHYSICAL LOCATION BASED RESOURCE DISCOVERY METHODS

On the basis of in-depth analysis of the mapping storage methods of the Identifier-Locator separation network, Jianwei Zhang et al., [26] proposed a physical topology aware Chord model (Ant-Chord) which is based on ant colony algorithm. The ideas of Ant-Chord is to regard the storage nodes in the whole Chord as a TSP problem and solve the TSP problem quickly by using the ant colony algorithm, then to build the Chord with the obtained TSP solution, and "Luoyang shovel" method is applied to optimize the Ant-Chord's routing hops. Adding proximity routing table and dividing proximity physical Region are the two main ways to achieve physical topology match. Adding proximity routing table has the minimum changes with the Chord structure, but increases the routing table overhead. Dividing proximity physical Region on the one hand, needs to solve the balances between the number of Regions and the number of nodes within a Region, on the other hand it is opaque between each Regions, so the efficient of inter-domain communication is not ideal. As a widespread concerned academic issue, TSP problem has been researched for a long time and is also a standard test platform for a number of new algorithms. Metaheuristic algorithm is very successful in solving TSP problem and ant colony optimization (ACO) is one of the classic meta-heuristic algorithms. Simply matching the physical topology has little effect to the logical routing hops of Chord, "Luoyang Shovel Method (LSM)" optimization method is designed which records the storage space information of successor nodes when generating the routing table of Chord nodes to increase the Chord routing flexibility and achieve the optimal routing hops.

In CLT Chord [15], the cache location tables are maintained to store the details of nodes which are at its close geographical range. These details are collected while the node is joining to the network. For this reason, geographical range of new node is attached to the join request and it is sent to the server. After receiving the request, the server send a response message to the new node in which IP address of its successor node and one of the nodes in the same geographical range are contained. Then the new node joins the network and communicates with the node that is in its geographical range to put entries in the cache location table. In this work, cache location table is stored in cache memory. Due to the continuous change of network, the nodes in geographical

range must communicate with each other in specified intervals, ie the nodes ping each other. Since node joins and leaves the network frequently, cache location tables should be updated to route the requests properly, cache location table arrange the nodes based on goodness like lifetime of a node, desired number of response sent to other node, desired number of response sent to other nodes, and time delay between the nodes. The node begins lookup requests from the first better node and if it doesn't receive its desired response, forwards the request to the second better node and so on. The algorithm continues to work like the basic Chord protocol.

Quasi-Chord [16] is in the light of the ideology of Chord. This model utilizes global network (GNP) system to coordinate the host on the physical layer and uses the Cantor space filling Curve to map the 2-dimensional geometrical space into 1-dimensional. Then Quasi-Chord model is built according to the Cantor Value. Quasi-Chord constructs the logic topology on the basis of the underlying topology information. So the nodes close in the underlying layer will be also close in the overlay network, which can greatly reduce the network traffic. To build up Quasi-Chord model, three steps are focused. In the first step, GNP [27] is used to get each host's coordinates in P2P geometric space; In the second step, Cantor space filling curve [28] is used to convert the 2-dimensional space into 1-dimensional; Quasi-Chord circle is constructed according to the host's Cantor value. The two main advantages of the method are the correspondence relationship of physical layer and logic layer which more are precise and combine network positioning and space filling curve to construct the P2P network, which can greatly reduce the network traffic brought by the TTL.

In [29], F.Dabek et al., proposed Chord PNS algorithm in which each routing table maintaining a set of candidate nodes and selects the proximal node from the candidate nodes to route the query request. The algorithm reduces the query delay, but within the process of proximal node discovery and routing table maintenance, ChordPNS brings heavy network traffic overhead.

Javad Taheri et al., introduced the TAC [30], a novel topology-aware protocol which is based on Chord. TAC introduces the local ring concept by dividing the geographical space into smaller areas. Through binding each new node to a proper local ring concerning its physical location, TAC considers the physical network topology of the overlay network to demonstrate more efficient key lookup.

6. ANT COLONY OPTIMIZATION

ACO is a class of algorithms, which is inspired by the behavior of real ants. Ants keep on depositing a chemical known as pheromone while moving from one place to other in search of food. An important and interesting behavior of ant colonies is their foraging behavior, and, in particular, how ants can find the shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called pheromone, forming in this way a pheromone trail. When more paths are available from the nest to a food source, a

colony of ants may be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back. This pheromone trail directs the search of future foods. Simulation of ant's behavior in computer system is a kind of parallel search over several constructive computational threads. Concurrent and asynchronous set of computational agents, known as colony of ants, move through the states of the problem corresponding to its partial solution. The decision to move to next state is influenced by pheromone trail on the route and heuristic information. Each ant constructs the solution during its tour. Pheromone trails are updated at the completion of tour, which will direct the search of future ants.

Basic ACO Algorithm:

1. Randomly place ants at different points in the search space.
2. For each ant k (currently in state s) Repeat steps 2.a, 2.b till ant k has completed its tour.
 - a. Choose the probability to move to next state.
 - b. Add the chosen move to ant's set.
3. Update the pheromone trails.
4. If terminating condition met Stop.
5. Else go to step 2.

At each iteration of the algorithm, each ant moves from a state to state, corresponding to a more complete intermediate solution. Thus, each ant k computes a set $A_k(x)$ of feasible expansions to its current state in each iteration, and moves to one of these in probability. When all the ants have completed a solution, the trails are updated. Based on the pheromone value, ants will select the shortest path to search the food.

7. PROPOSED METHOD

Scalability, load balancing, good performance, simplicity are the main features of the Chord P2P protocol. Despite these

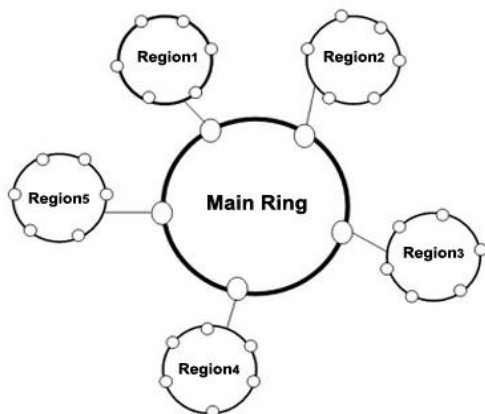


Fig 2: Chord with Multiple Region Ring

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. These drawbacks are focused in this work 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.

In our proposed work, while initializing random grid node placements, 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. 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.

Region-length finger table is used in our 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. In 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.

7.1 Geo-Chord Algorithm

The following are the steps followed in our proposed method.

1. Randomly generates the required node positions of all the nodes in the network. Node position includes the data about longitude and latitude.
2. For each node $u \in V(G)$, identify the neighbors of u (i.e., set of nodes which are within the Ring of node u).
3. Compute the Euclidean distance matrix and Hop distance matrix to construct Regions.
4. Region heads form the Main Chord and have links with other nodes through heads.

Extract a node, say x , from N such that $SS(x)$ is maximum. Any tie is broken by choosing a node having higher weight. For any node u in the network, the resource value denoted by $SS(u)$ is defined to be $SS(u) = (m_1 + m_2/2 + m_3/4)K$, where K is any constant (a fixed threshold value) and m_1, m_2, m_3 denote respectively the number of strong, medium and weak neighbors of u . Based on this value, Region-head is selected.

5. Each Region is formed based on the physical location of nodes.
6. Each node maintains Region- finger table to maintain Region code and node ip address. This makes the search process very simple.
 IP address = Region code +IP address
7. To find a successor of a key, the following procedure is presented.

a. Search process by a node first checks the Region code and locates the corresponding Region Ring.

b. When the entire grid is free, the node which plays the Region-head of the Region-ring receives the query and ACO algorithm is applied to locate the query. If the key is located in the Region, the result is sent to the originated node. Otherwise go to step 7(c).

c. If the head of the specific Region ring is busy, the successor of the Region-head will receive the query and search is continued as in step 7(b). (ie), the next Region-head which is physically near that Region-head proceeds the query.

d. If the entire grid is busy, the query has to wait till any of the nodes in grid is free.

8. SIMULATION ANALYSIS

NSC_SE (Network Simulation Capture Special Edition) simulator is used to evaluate the performance of the proposed GEO-Chord. Motes, Androids, T-Engines, Nokia epocs, Hackman tool, UH_ABC (Universal Hack for All Block Ciphers), Linphone, Netsnooper, Wireshark and many more

applications/libraries can be connected with NSC_SE. NSC_SE provides all information including connection establishing time, jitter, IP delay, Latency, Error rate, Pocket loss, Power Consumption and so on. Along with the Simulator, Wireshark, the free and open-source packet analyzer is also used for network communication and to maintain all the required basic information of Chord protocol.

Chord network is considered before and after improvement with $N=256, 512, 1024, 2048, 4096, 8192, 16834, 32768$. The comparisons among base Chord, ACO-Chord and Geo-Chord are made in terms of average number of hops, messages, and average communication time. As shown in table 1, the average number of messages, hops and communication time taken by Geo-Chord is better than ACO-Chord and base Chord.

9. PERFORMANCE EVALUATION

Figure 3, 4 and 5 respectively shows the average number of hops, the average communication time and the average number of messages for Chord, ACO-Chord and Geo-Chord with changes of total number of nodes in P2P systems. Obviously, the average number of hops, communication time and messages are reduced in Geo-Chord. The average number of hops taken by the Geo-Chord is the same as the ACO-Chord but Geo-Chord reduces the communication time dramatically than Chord and ACO-Chord. The Region finger table is periodically updated to maintain the exact number of neighbouring nodes. From the simulation result, it is concluded that when compared to the Chord algorithm, Geo-Chord performs well in terms of communication time.

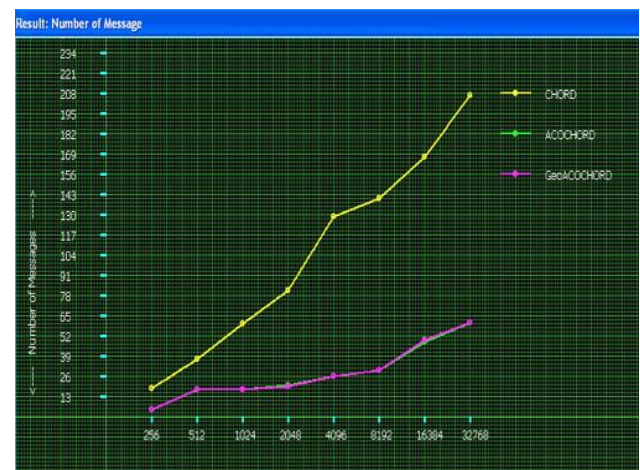


Fig.3. Number of Messages per peer

Table 1: Comparison Table

	Nodes							
	256	512	1024	2048	4096	8192	16384	32768
Messages :								
Chord	20.4	41.3	64.8	87.2	136.4	149.1	179.4	219.5
ACO Chord	5	19.4	19.2	21.6	28	31.1	52.8	68.4
Geo Chord	5	19.3	19.4	21.9	28	31.4	53.2	66.2
Hops:								
Chord	4.5	6.6	7	6.7	8.2	8.3	9.3	9.4
ACO Chord	3	4	4.4	4	4	5	6	7
Geo Chord	3	4	4.3	4	4	5	6	7
Communication Time: (ms)								
Chord	1029	1121.9	1215.6	1264.1	1417.3	1532.0	2037.1	2846.6
ACO Chord	411.2	526.1	642.0	754.2	863.8	979.9	1141.1	1263.5
Geo Chord	246.5	315.3	379.9	448.6	517.5	584.2	679.1	765.4



Fig.4. Number of hops per peer

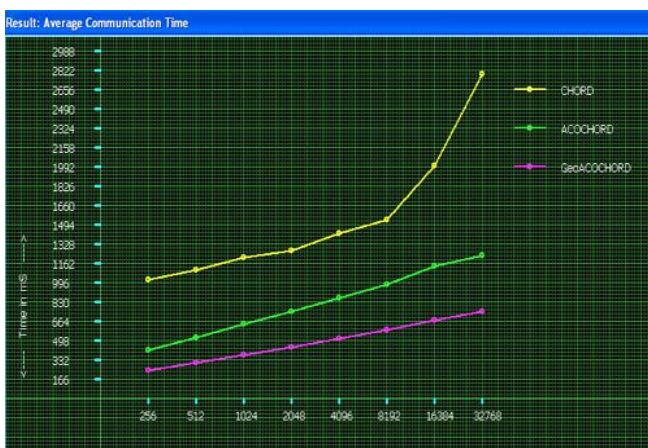


Fig.4. Average Communication time

10. CONCLUSION

As more and more resources appear in grids, there is an increasing need to discover these resources effectively and efficiently. The aim of this work is to efficiently locate the resources in Grid environment by using P2P techniques. Chord P2P protocol plays a vital role in Grid resource discovery. Due to its lack of physical network topology information, Chord suffers from low routing efficiency, and ineffective use of bandwidth in data lookup. To alleviate this problem, Geo-Chord, physical topology aware Peer-to-Peer system is proposed which is based on Chord. Since Chord is the appropriate choice for reliable search, it is experimented based on the geographical location of the nodes in the network. From the Simulation results, it is concluded that the Geo-chord method effectively reduces the required number of hops, messages and communication time.

REFERENCES

- [1] I. Foster, C. Kesselman, J. Nick, S. Tuecke: The physiology of the Grid: an open Grid services architecture for distributed systems integration: in Open Grid Service Infrastructure WG, Global Grid Forum, 2002.
- [2] W. Xing, O. Corchob, C. Goble, M. Dikaiakos: An ActOn-based semantic information service for Grids: Future Generation Computer Systems 26 (2010) 324–336.
- [3] I.Foster: Globus toolkit version 4: Software for service oriented systems: Journal of Computer Science and Technology, vol. 21,pp.513-20, July 2006.
- [4] I. Foster, C. Kesselman, J. Nick, S. Tuecke: The physiology of the Grid: an open Grid services

- architecture for distributed systems integration: in Open Grid Service Infrastructure WG, Global Grid Forum, 2002.
- [5] J. Nabrzyski, J. M. Schopf, J. Werglarz.: Grid Resource Management: State of the Art and Future Trends: Springer, September, 2003, 598p.
- [6] Napster web site: 2011. <http://www.napster.com>.
- [7] Gnutella Protocol Specification: 2002. http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html.
- [8] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim: A survey and comparison of peer-to-peer overlay network schemes: IEEE Communications Surveys and Tutorials 7 (2005) 72–93.
- [9] Q.H. Vu, M. Lupu, B.C. Ooi: Peer-to-Peer Computing—Principles and Applications: Springer, 2010.
- [10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications: Proc. of ACM SIGCOMM’01, San Diego, USA, August 2001.
- [11] S. Ratnasany, P. Francis, M. Handley, R. M. Karp, and S. Shenker: A Scalable Content-Addressable Network: Proc. of ACM SIGCOMM’01, San Diego, USA, August 2001.
- [12] M. Frans Kaashoek, D. R. Karger and Koorde: A Simple Degree-optimal Distributed Hash Table: Proc. Of 2nd Int. Workshop on Peer-to-peer Systems (IPTPS’03), Berkeley, USA, February 2003.
- [13] Wei Lv, Qing Liao, Jingling Zhao, Yonggang Xiao: TB_Chord: An Improved Routing Algorithm to Chord Based on Topology-aware and Bi-Dimensional Lookup Method: 978-1-4244-3693-4/09/\$25.00 ©2009 IEEE.
- [14] Eric Jui-Lin Lu, Yung-Fa Huang, Shu-Chiu Lu: ML-Chord: A multi-layered P2P resource sharing model: Journal of Network and Computer Applications 32 578–588 (2009).
- [15] Jaber Karimpour, Majid Moghaddam, Ali A. Noroozi.: CLTChord- Improving lookup at the Chord protocol using cache location table: International Journal of Soft Computing and Software Engineering (JSCSE), Vol.2,No.7, 2012.
- [16] SUN Mingsong and ZHANG Zhongqiu: Quasi-Chord: physical topology aware structured P2P network: Proceedings of the 11th Joint Conference on Information Sciences (2008).
- [17] Mohammed Bakri Bashir, Muhammad Shafie Bin Abd Latiff, Aboamama Atahar Ahmed, Yahaya Coulibaly, Abdul Hanan Abdullah and Adil Yousif, (2011): A Hybrid Resource Discovery Model For Grid Computing: International Journal of Grid Computing & Applications (IJGCA) Vol.2, No.3.
- [18] Ramya M and Nagarathna N: Survey Paper on Resource Discovery Model in Grid Computing: International Journal of Engineering Research & Technology (IJERT) ISSN 2278-0181 www.ijert.org Vol. 2 Issue 6, June – 2013.
- [19] C. Mastroianni, *et al.*, (2005): A super-peer model for resource discovery services in large-scale Grids: Future Generation Computer Systems, vol. 21, pp. 1235-1248.
- [20] Ali Yousefi and Ali Sarhadi: Proffering a New Method for Grid Computing Resource Discovery with Improved Genetic Algorithm by Means of Learning Automata Based on Economic Criteria: World Applied Sciences Journal 26 (4): 560-565, ISSN 1818-4952 © IDOSI Publications, 2013.
- [21] Saeed Javanmardi, Shahdad Shariatmadari and Mohammad Mosleh: A Novel Decentralized Fuzzy based Approach for Grid Resource Discovery: International Journal of Innovative Computing 01(1) 23-32.
- [22] Carlos Pérez-Miguel, Jose Miguel-Alonso, Alexander Mendiburu: High throughput computing over peer-to-peer networks: Future Generation Computer Systems 29 (2013) 352–360.
- [23] D. Talia, P. Trunfio: Toward a synergy between P2P and Grids: IEEE Internet Computing 7 (4) (2003).
- [24] I. Foster and A. Iamnitchi: On death, taxes, and the convergence of peer-to-peer and Grid computing: 2nd International Workshop on Peer-to- Peer Systems (IPTPS’03), Berkeley, CA, 2003.
- [25] M. Marzolla, M. Mordacchini, S. Orlando: Peer-to-peer systems for discovering resources in a dynamic Grid: Parallel Computing 33 (4–5) (2007) 339– 358.
- [26] Jianwei Zhang, Si Liu, Zhen He and Zengyu Cai: A Physical Topology-Aware Chord Model based on ACO: JOURNAL OF COMPUTERS, VOL. 6, NO. 12, DECEMBER 2011.
- [27] T. S. Eugene Ng and Hui Zhang: Towards Global Network Positioning: *Extended Abstract, ACM SIGCOMM Internet Measurement Workshop2001, San Francisco, CA, and November 2001.*
- [28] Pei Zhi Lee, Chong Wei Huang and Tian Yuan Shih: On Cantor’s Space- Filing Curve: *Journal of Taiwan Geographic Information Science* (4), pp. 13-26, 2006.
- [29] F. Dabek, J. Li, E. Sit, J. Robertson, M. Kaashoek and R.Morris: Designing a DHT for Low Latency and High Throughput [A]: In: Proc. of the 1st Symposium on Networked System Design and Implementation (NSDI ’04) [C], 2004: 85-98.
- [30] Javad Taheri, Mohammad Kazem Akbari: TAC- A Topology-Aware Chord-based Peer-to-Peer Network: Journal of Iranian Association of Electrical and Electronics Engineers: Vo1.4- No.2- Fall and Winter 2007.