J. Appl. Res. Ind. Eng. Vol. 5, No. 4 (2018) 346–355

Journal of Applied Research on Industrial Engineering
www.journal-aprie.com

# Toward the Optimization Resource Discovery Service in Grid Systems: A Survey

***Javad Pourqasem***[*]

*Department of Computer Engineering, Guilan University, Rasht, Iran.*

**PAPER INFO**

**Chronicle:**
Received: 04 July 2018
Revised: 07 September 2018
Accepted: 27 November 2018

**Keywords**:
Grid System.
Peer-to-Peer.
Discovery Service.
Unstructured.
Super-Peer.

**ABSTRACT**

As Grid systems distribute geographically, they become more heterogeneous and dynamic in terms of resources. Discovery service is an important issue in grid systems. Grid efficiently is as regards resource discovery in scalable and dynamic environments. In this paper, we review the peer-to-peer (P2P) based decentralized discovery methods: Unstructured and Super-Peer. We describe major development of these approaches and provide discussion concerning efficiently and scalability features.

## 1. Introduction

The term “Grid” was invented in the mid-1990s offering access to a vast collection of heterogeneous resources as a single, unified resource to solve large scale computing and data intensive problems in advanced science and engineering areas [1-6]. There are many types of resources in grid such as computers, cluster of computers, online tools, storage space, data, and applications. These resources are widely distributed, heterogeneous, and dynamic in comparison with traditional and cluster systems [7]. The objective is to virtualize the geographically distributed resources and enable users and applications to access resources in a transparent manner. A large-scale and complex grid applications include hundreds or thousands of geographically distributed resources, which influences the gird performance, fault tolerance, and scalability criteria [8].

Resource discovery is a basic service in the grid systems which finds the resources matching the user's application requirements [9, 10]. Discovery is one of the fundamental challenges and a complex process in grid. It discovers appropriate resources based on the submitted queries in dynamic and scalable environment [11]. Discovery service involves the enormous number of resources, geographically distributed, heterogeneous and dynamicity of resources, resource breakdown, and reliability of discovery service [12].

---

[*] Corresponding author
E-mail address: jpourmail@gmail.com
DOI: 10.22105/jarie.2018.88379

The traditional approaches of discovery include a centralized or hierarchically structure [13, 14] to index resource information in grid. However, efficiency of the traditional approaches is limited: the root node of the hierarchical method suffer from the inherent drawback of a single point of failure; the bottleneck of system in highly dynamic environment; there is not possibility to scale well. To cope with the mentioned problems, grid integrate with Peer-to-Peer (P2P) architecture [15, 16]. P2P systems have inherit features of autonomy, robustness, scalability, and availability of distributed resources. P2P grid optimize the discovery service in dynamic, heterogeneous, and distributed environment [17-19].

In this research, we describe the decentralized discovery methods in grid system: Unstructured and Super-peer architecture [20]. The unstructured approaches are decentralized, autonomous, and need no precise control over network topology, consequently some methods propagate the query messages to all nodes by flooding the query in grids [21-25]. The super-peer networks provide a balance between the inherent efficiency of centralized approaches, and the autonomy, load balancing, and fault tolerant features offered by the decentralized methods [26-28]. The super-peer method implements the efficient discovery service for large-scale grids [29].

The rest of this paper is organized as fellows: Section 2 present the implemented unstructured discovery service usable for grid environment. Discovery services based on super-peer is discussed in Section 3. Finally, Section 4 concludes this paper.

## 2. Unstructured Discovery Services

In unstructured peer-to-peer systems there are no specific constraints for placement of resources and nodes. The resources are randomly spread across peers in the network. According to random placement of resources in unstructured architecture, such systems usually use flooding or random walk approach to forward a query among a large number of peers [30]. In this environment each peer is randomly connected to a fixed number of peers and there is no information about the location of resources [31].

### *2.1 Napster*

Historically, Napster was the first P2P system which was introduced in 1999. It contained a central server which stored the index of all files shared by the peers in the network. For locating process, a user sends a query to the central server using the name of the file and receives the IP address of a peer containing the file. In file downloading faze, the communications between the requesting peer and the peer containing the file is direct. However, the central index server used in Napster is not easy to scale and is a single point of failure. Although Napster is generally considered as the first unstructured P2P system, the existence of a central index distinguishes it considerably from today’s unstructured P2P systems [31].

### *2.2 Gnutella*

Gnutella is one of the most popular unstructured P2P systems that emerged after Napster. Any active node in Gnutella bootstraps the network. Querying for requested files is carried out via the basic flooding. There are three messages: ping, pong, and bye. The pings messages are used to discover peers on the network, pongs messages are replies to received pings messages that contain information about the responding peer its neighboring peers. The byes messages are optional messages that notify peer of forthcoming closing connections. In query processing, Gnutella uses a simple flooding strategy, and a query is propagated to all neighbors within a certain number of hops [32]. There is no stable status because of the frequently disconnections of peers. When the network become large enough, it get saturate and often cause huge delays. To decrease the query-related traffic, Gnutella limits the maximum number of hops in discovery process or consider time to Live (TTL) duration. As a result, queries often drop and discovery process reach the unsatisfactory result as only minor portion of peers are searched [32]. The new production of the Gnutella attempted to improve the query efficiency and reduce the control traffic overhead through a two-level hierarchy that classifies peers: super-peers as the network core with high capacity and leaf-peers as clients.

### *2.3 Iamnitchi and Foster Method*

A fully decentralized discovery method in grid environments has proposed by [33]. This resource discovery solution consists of four components: membership protocol, overlay construction, preprocessing, and request processing. The membership protocol defines how the new peers join the network and how the peers learn about each other. Peers use the overlay construction function to choose the set of neighbors from the local membership list. Practically, this set may be restricted based on the available bandwidth, message-processing load, security or administrative policies, and topology specifications criteria. Preprocessing is the off-line processing used that enhances the search performance before executing the query requests. For example, a preprocessing technique spreads the resource descriptions all over the peers, which is descriptions advertisement of the local resources. The request processing task is done locally and remotely. A peers searches a request in own information and has ability to forward the request. In this method, every participant in a Virtual Organization (VO) that publishes information in one or more local servers called node or peer. A node may publish information about one or multiple resources. Users generally send their requests to a local node, that node responds with a matching resource description, else it forwards the requests to another nodes. Intermediary nodes forward the request until its TTL expires or the matching resources is found [33]. This mechanism supports attribute-based search and there is not a central control. However, it does not scale well because of the large volume of query messages produced by flooding. The search results are not deterministic and this approach cannot warranty to find the desired resource even if it exists. Furthermore, in order to avoid flooding of entire the network, the number of hops on the forwarding path are bounded by the TTL.

### *2.4 Learning Automata-Based Resource Discovery*

Akbari Torkestani proposed a decentralized learning automata-based resource discovery algorithm for large-scale unstructured P2PGrids [34]. This algorithm decreases the negative impacts of the flooding problem on the network performance and supports the multi-attribute range queries. In this mechanism, each peer uses a learning automaton. The resource queries are forwarded through the shortest path (the path with the minimum hop count) toward the peers more likely having the requested resources. Each peer selects a communication link to route the resource owner, so that this link is randomly selected by the automaton. If the selected route at each step is shorter than the average length of the routes selected so far, algorithm prizes the selected route, else it is penalized. Therefore, as the algorithm proceeds, it converges to the route with the minimum length. This algorithm provides scalability and dynamicity for P2PGrids systems.

### *2.5 Ant Colony Optimization (ACO)*

Ant Colony Optimization (ACO) [8] is a resource discovery mechanism suitable for large-scale P2PGrid systems. In this approach, when a user queries the nearest Meta-Computing Directory Service

(MDS), the MDS which is called nest, will give out one ant or several ants to find the resources in regard to the user's requirement. Firstly, the ants walk randomly from nest to nest, if the ants find the requested resources, return the path to their initialized nest, the routing information are. The other ants which are searching for the same resources, travel the network based on routing information, consequently most of the ants with most probability select the shortest path. This approach avoids a large-scale flat flooding of the unstructured method and sends the packets along the routes that are frequently traveled. Furthermore, to improve the efficiency, this method utilizes multiple ants in parallel. The ants in ACO approach can carry a large amount of information in their memory. This method provides multi-attribute range query and multiple user's requirements are maintained in ant memory.

### *2.6 Brunner et al. Approach*

A study [35] analyses read-dominated data [36], and concentrates on reducing the lookup costs associated with the discovery of a suitable resource to execute a batch of jobs. This technique uses the data summarization based on cobweb clustering [37] to generate summaries of resource properties. This method supports scalability of P2P-based content networks, because this way reduce the amount of data that transfer between domain's brokers to discover the requested resource. Summaries are generated in three different steps, namely pruning, leafing, and filtering. To reduce the latency in network, brokers filter the results from the summarization technique and order and select the search results based on a Round Trip Time (RTT) below a threshold. Brokers in each domain act as peers and propagate the summaries of the local resources to other domains. When a user wants to execute a job, he contacts his local broker. If the requested resource don't find in local domain, the broker searches among the summaries that other domains forward to it. The broker finds the domain that more likely has the resources matching the query and yield the lowest RTT. Finally, the query is send to the chosen domain to answer it.

### *2.7 Caminero et al. Method*

Approach [38] tackled this limitation by using the Routing Indices (RIs) [39] and a new version of it, called Hop-Count Routing Indices (HRI) to construct summaries and route queries to neighbor peers. HRIs consider the number of hops that is need to reach a datum. To adapt summaries to be used within RIs, n-level summaries is presented. In n-level summaries, the precision of the j-level summary of a domain become higher than that of i-level summary if $i < j$. This method filters the attribute–value pairs whose probability is below a threshold. This approach use goodness function [39] based on HRI table to decide which of the neighbors is more likely to have resources matching some set of requirements. This function is computed for each neighbor domain using the summaries stored in the local HRI, and represents the probability of each neighbor domain of having a resource with the needed requirements.

### *2.8 Marzolla et al. System*

Marzolla et al. [40] were proposed another approach for grid resources discovery based on Routing Indices (RIs). In this mechanism, Resource Brokers (BR) are organized in a tree-structured overlay network (Fig. 1). Each BR maintains information about the set of resources that it manages directly and a condensed description of the resources are presented in the sub-trees of its neighbor BRs. This condensed description consists of a bitmap indexes of the value of resources attributes. This method uses the k-bit vector to represent the indexes and exploits the bitmap indexes to route queries towards areas where matches can be found. For query processing, each user submits request to one of the BRs, which routes the query and returns response from other BRs in the network. Each BR which receives a query from another BRs, first maps it into a binary vector and then examines whether has some resources matching the request; if so, a query hit returned. Query is only forwarded to each BRs whose indexes satisfy query based on a Directed Breadth First (BFS) Search algorithm. In order to prevent flooding the network, query are forwarded only to a subset of neighbor BRs, excluding those paths which certainly will not reach any useful resource. If the value of resource attribute changes, update messages are transmitted to neighbors only if the new bitmap representation of the resources differs from the old one. In this approach update messages involve a constant number of peers, regardless of the network size.

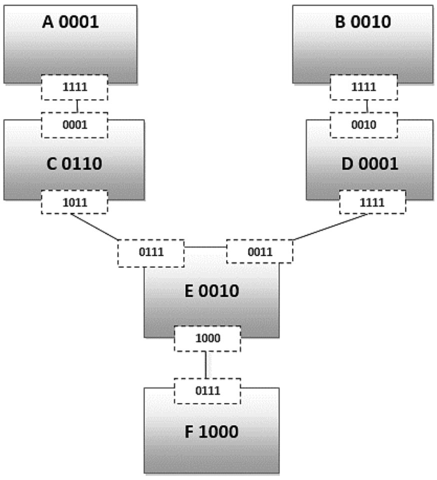

***Fig. 1.*** *The P2P network with bit vector indices.*

## 3. Super-Peer Mechanisms

The super –peer mechanism is an approach that facilitates the merging of P2P models and Grid environments. The super-peer networks have been proposed to realize a balance between the inherent efficiency of centralized search, and the autonomy, load balancing and fault-tolerant properties offered by distributed search [27]. In a super-peer network, all peers can be organized as peers and super peers. A super-peer node acts as a central server for a number of regular peers and connects to other super-peers to form an overlay network [41]. This model is used to implement very efficient implementation of the information service and it is naturally appropriate for large-scale Grids. A Grid can be considered as a network composed of small-scale, proprietary Grids, called Virtual Organizations (VOs). Inside each VO, one or more nodes, e.g. those that have the greatest capabilities, can act as super-peers, while other nodes can use super-peers to access the Grid. A super peer works as a server to connect many client peers, and is responsible for tasks such as discovery and routing [41].

### *3.1 KaZaA*

KaZaA was introduced in March 2001 and accepted the super-peer model in its design [42, 43]. Many millions of people used KaZaA program to share files without any servers. KaZaA consists of two types of nodes, super nodes and leaf nodes. An overlay network is formed among the super nodes, each of which includes a set of leaf nodes. Leaf nodes are connected to super-peers and each super-peer indexes the files that nodes own and answers the query on nodes behalf.  Therefore the search is handled among super-peers likely via flooding. Super-peers are selected dynamically based on high bandwidth, low latency, and plenty of computational and memory resources. A leaf node provides the information for a super-peer about a given file such as a single filename, the file size, the file descriptors and the content hash that is used to identify the file in an HTTP download request. When a node needs a file, it issues a request to its super node, which initiates a search process in the overlay network to locate the resource. KaZaA is more scalable than Napster and Gnutella for its hierarchical architecture. However, KaZaA cannot support complex queries since the queries are routed in spite of their content.

### *3.2 Gnutella2*

As reviewed before, Gnutella does not scale well in very large scales and suffers from drawback produced by flooding. Unscalability of the original Gnutella motivated the development of the second version of the system in 2002. The Gnutella2 consists of two types of nodes: leaves (normal peer) and hubs (cluster-heads or super-peers). Each leaf, has one or two connections to hubs. Cluster-heads index resources information of hundreds of peers by means of a Query Routing Table. During the search process, a leaf (peer) sends a request to a hub. If it answers, the peer downloads the requested file directly from the node that owns the files. If not, the request is forwarded by the current hub to another hub. The discovery process stops when either the item is found or all known hubs are searched or a predefined search limit is reached. This approach significantly decreases the traffic in the network and makes the system highly scalable compared to original Gnutella. However, as a tradeoff, the complexity of Gnutella2 is higher than Gnutella and suffers from maintenance overhead [44, 45].

### *3.3 Mastroianni et al. Method*

A study [29] implemented the super-peer model to design a P2P-based Grid information service. In this environment each super-peer exploits the centralized/hierarchical information service provided by the Grid infrastructure of the local VO. A super-peer realizes two main tasks: it is responsible for the communications with the other VOs, and it holds metadata about all the resources belonging to nodes of the local VO. The set of nodes belonging to a VO (i.e. the super-peer and the ordinary nodes) is also represented to as a cluster. This model use membership protocol for communicating that exploits the characteristics of contact nodes. A contact node is a node in Grid system that plays the role of an intermediary node. There are one or more contact nodes available in each VO. Each time a VO wants to connect to the Grid, the corresponding super-peer contacts a subset of contact nodes and registers at those. Therefore, the selected contact nodes randomly choose a number of previously registered Grid super-peers and link their addresses to the requesting super-peer: these super-peers will establish the neighbor set of the new Grid super-peers. A super-peer communicates with contact nodes either periodically or every time it detects the neighbor super-peer disconnection, in order to substitution .The resource discovery mechanism is defined as follows. A node in Grid forwards the Query messages to the local super-peer. The super-peer examines the local information service for the requested resources in some of the nodes belonging to the local VO. The super-peer returns to the requesting node a queryHit containing the ID of the node whose owns the resources. Moreover, the super-peer forwards a copy of the query to a selected number of neighbor super-peers. Every time a resource matching is found in a remote VO, a queryHit is generated and is forwarded along the same path back to the requesting node. Then a notification message is sent by the remote super-peer to the node that handles the requested resource. The set of neighbors to which a copy of query is forwarded is determined through an experiential approach. Each super-peer keeps statistics on the amount of queryHits received from all the known super-peers [29]. This approach provides autonomy, load balancing and fault-tolerance features for discovery systems. However, since a super-peer node works as a central server for a number of client peers, this mechanism suffers from single point of failures in each cluster. Furthermore, when the number of requests are increased, the super-peers may suffer from bottleneck, which may limit the scalability of this system. Lastly super-peer mechanism cannot support dynamic-attribute queries since the periodic updates of the resource information.

### *3.4 Talia and Trunfio Discovery Service*

Talia and Trunfio presented a P2P architecture for resource discovery in OGSA-compliant Grids [46]. The architecture consists of two layers (see Fig. 2): the lower layer is a hierarchy of Index Services (IS), which publish resource information owned by each VO; the upper layer is a P2P Layer, which contains two types of Grid Services: Peer Services (PS) are used to perform resource discovery, and Contact Services (CS) that allow PS to organize themselves in a P2P network. There is one PS for each VO. Each PS is connected with a set of neighbor PS and exchanges query and response messages with them in a P2P method. To process the query, a PS invokes the top level Index Service of the corresponding VO. A query response is sent back along the same path that query was forwarded. When a PS wants to join the P2P network, must know the URL of at least one PS to connect to. CSs store the URLs of known PS and a PS may communicate one or more well-known CS to obtain the URLs of registered PS [46].

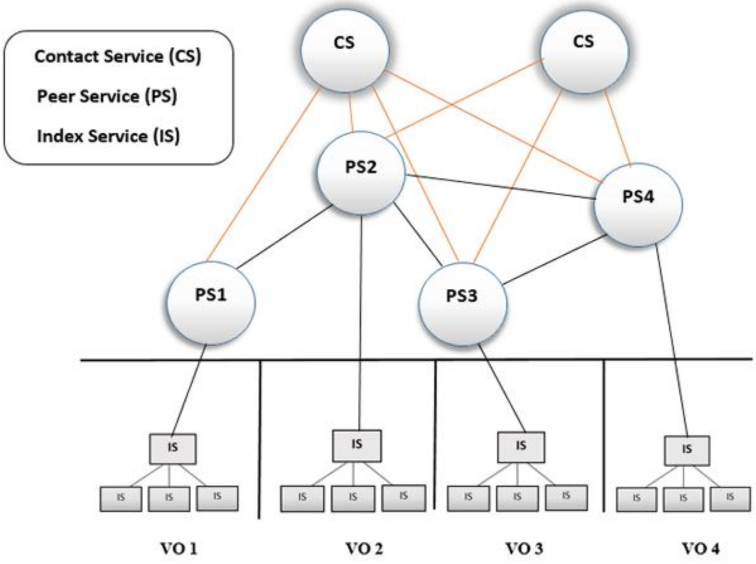

***Fig. 2.*** *The two layers discovery architecture.*

### *3.5 Puppin et al. Grid Information Service*

A Grid Information Service (GIS) based on peer to-peer (P2P) technologies and Routing Indices (RI) was presented in [47]. The system consists of two main entities: Agent and Aggregator. The agent is responsible to publish information about a node's resources belonging to the super-peer. The aggregator indexes data from agent, answers to queries and forwards them to the other super-peers in each super-peer; it also maintains an index about the information stored in neighbor super-peers. This mechanism is based on Globus Toolkit and complies with the OGSA standard. The Information System is made as a network of super-peers, which aggregate the information about resources within virtual organizations

(VOs). This method used Routing Index (RI) to improve the performance of query routing, and to avoid of flooding. The RI represents the availability of resource of a specific type in the neighbor super - peer's nodes. This method employed a version of RI called Hop-Count Routing Index (HRI) [38], which considers the number of hops needed to reach a result. Each node submits the query to its cluster's super-peer, which will pass it to other super-peers if needed. The system has problems such as false-positive errors and single point of failure on each cluster. However this approach can be built as a redundant network, where super –peers are replicated within each cluster. Furthermore, regular updates may limits the dynamic attribute queries in highly dynamic networks, however the range and multi-attribute queries are supported.

## 4. Conclusions

Resource discovery is one of the important services in grid environments that its functionality affects the performance of grid systems. We concentrated on decentralized discovery methods in a scalable P2PGrid environment. We discussed the unstructured and super-peer implemented resource discovery models in detail as regards efficiently, scalability features.

## References

[1] Jasma, B. and Nedunchezhian, R. (2012). A hybrid policy for fault tolerant load balancing in grid computing environments. Network and computer applications, 35(1), pp. 412–22.

[2] Erdil, DC. (2012). Simulating peer-to-peer cloud resource scheduling. Peer-to-Peer Networking and Applications, 5(3), pp. 219–30.

[3] Wang, L., Jie, W., & Chen, J. (2018). Grid computing: infrastructure, service, and applications. CRC Press.

[4] Younis, M. T., & Yang, S. (2018). Hybrid meta-heuristic algorithms for independent job scheduling in grid computing. Applied soft computing, 72, 498-517.

[5] Kocak, T. and Lacks, D. (2012). Design and analysis of a distributed grid resource discovery Protocol. Cluster computing, 15(1), pp. 37–52.

[6] Pourqasem, j., Karimi, s., Edalatpanah, S.A. (2014). Comparision of cloud and grid computing. American Jornal of Software Engineering, 2(1), pp. 8-12.

[7] Deng, Y., Wang, F., Helian, N., Wu, S. and Liao, C. (2008). Dynamic and scalable storage management architecture for grid oriented storage devices. Parallel computing, 34 (1), pp. 17-31.

[8] Deng, Y., Wang, F. & Ciura, A. (2009). Ant colony optimization in spired resource discovery in P2P Grid systems. The journal of super computing, 49(1), pp. 4–21.

[9] Sarhadi, A., Yousefi, A. & Broumandnia, A. (2012). New method for grid computing resource discovery with dynamic structure of peer-to-peer model based on learning automata. World Applied Sciences Journal, Volume 19(1). avalable at : http://dxdoi.org/ 10.5829/idosi.wasj.2012.19.01.1597.

[10] Hameurlain, A., Cokuslu, D. & Erciyes K. (2010). Resource discovery in grid systems: a survey. International Journal of Metadata, Semantics and Ontologies, 5(3), pp. 251–63.

[11] Zarrin, J., Aguiar, R. L., & Barraca, J. P. (2018). Resource discovery for distributed computing systems: A comprehensive survey. Journal of parallel and distributed computing, 113, 127-166.

[12] Deniz, C., Abdelkader, H. & Kayhan, E. (2010). Grid resource discovery based on centralized and hierarchical architectures. International journal for infonomics, 3(1), PP. 227–33.

[13] Cokuslu, D., Hameurlain, A., & Erciyes, K. (2010). Grid resource discovery based on centralized and hierarchical architectures. International journal for Infonomics, 3(1), 227-233.

[14] Trunfio, P., Talia, D., Papadakis, H., Fragopoulou, P., Mordacchini, M., Pennanen, M., ... & Haridi, S. (2007). Peer-to-Peer resource discovery in Grids: Models and systems. Future generation computer systems, 23(7), 864-878.

[15] Chauhan, P. (2012, March). Decentralized computation and communication intensive task scheduling algorithm for P2P grid. 2012 UKSim 14th international conference on computer modelling and simulation (pp. 516-521). IEEE.

[16] Gueye, B., Flauzac, O., Rabat, C., & Niang, I. (2017). A self-adaptive structuring for large-scale P2P Grid environment: design and simulation analysis. International journal of grid and utility computing, 8(3), 254-267.

[17] Marzolla, M., Mordacchini, M. and Orlando, S. (2007). Peer-to-peer systems for discovering resources in a dynamic grid. Parallel computing, 33(4), pp. 339–358.

[18] Ma, S., Sun, X., & Guo, Z. (2010, July). A resource discovery mechanism integrating p2p and grid. 3rd international conference on computer science and information technology (pp. 336-339). IEEE.

[19] Zarrin, J., Aguiar, R. L., & Barraca, J. P. (2018). Resource discovery for distributed computing systems: A comprehensive survey. Journal of parallel and distributed computing, 113, 127-166.

[20] Navimipour, N. J., & Milani, F. S. (2015). A comprehensive study of the resource discovery techniques in Peer-to-Peer networks. Peer-to-Peer networking and applications, 8(3), 474-492.

[21] Navimipour, N., Rahmani, A. and Habibizad Navin, A. (2014). Resource discovery mechanisms in grid systems: Asurvey. Journal of network and computer applications, 41, pp. 389–410.

[22] Navimipour, N. J., & Milani, F. S. (2015). A comprehensive study of the resource discovery techniques in Peer-to-Peer networks. Peer-to-Peer networking and applications, 8(3), 474-492.

[23] Thampi, S. M. (2010). Survey of search and replication schemes in unstructured p2p networks. Retrieved from arXiv preprint arXiv:1008.1629.

[24] Xiao, R. Y. (2008). Survey on anonymity in unstructured peer-to-peer systems. Journal of computer science and technology, 23(4), 660-671.

[25] Gaeta, R., & Sereno, M. (2011). Generalized probabilistic flooding in unstructured peer-to-peer networks. IEEE transactions on parallel and distributed systems, 22(12), 2055-2062.

[26] Yang, B. B., & Garcia-Molina, H. (2003, March). Designing a super-peer network. Proceedings 19th international conference on data engineering (Cat. No. 03CH37405) (pp. 49-60). IEEE.

[27] Tian, C. Q., Jiang, J. H., Hu, Z. G., & Li, F. (2010). A novel super-peer based trust model for peer-to-peer networks. Chinese journal of computers, 33(2), 345-355.

[28] Venkadeshan, R., & Jegatha, M. (2012). Super peer deployment in unstructured peer-to-peer networks. In Advances in Computing and Information Technology (pp. 661-669). Springer, Berlin, Heidelberg.

[29] Mastroianni, C., Talia, D., & Verta, O. (2005). A super-peer model for building resource discovery services in grids: Design and simulation analysis. European grid conference (pp. 132-143). Springer, Berlin, Heidelberg.

[30] Pan, Z., & Zhao, L. (2018, May). Incentive strategy search algorithm in unstructured P2P network. Proceedings of the international conference on data processing and applications (pp. 54-57). ACM.

[31] Trunfio, P., Talia, D., Papadakis, H., Fragopoulou, P., Mordacchini, M. and Pennanen, M. (2007). Peer-to-Peer resource discovery in Grids: Models and systems. Future generation computer systems, 23, pp. 864–878.

[32] Meshkova, E., Riihijarvi, J., Petrova, M. and Mähönen, P. (2008). A surveyon resource discovery mechanisms, peer-to-peer and service discovery framework. Computer networks, 52(11), pp. 2097–128.

[33] Iamnitchi, A., & Foster, I. (2004). A peer-to-peer approach to resource location in grid environments. In Nabrzyski, J., Schopf, J. M., & Węglarz, J (Eds.), Grid resource management, pp. 413-429. Springer.

[34] Akbari Torkestani, J. (2012). A distributed resource discovery algorithm for P2P grids. Journal of network and computer applications, 35(6), pp. 28–36.

[35] Brunner, R., Caminero, A. C., Freitag, F., & Navarro, L. (2012). Network-aware summarisation for resource discovery in P2P-content networks. Future generation computer systems, 28, pp. 563–572.

[36] Yalagandula, P., & Dahlin, M. (2004). A scalable distributed information management system. SIGCOMM—computer communication review, 34 (4), pp. 379–390.

[37] Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. Machine learning, 2(2), 139-172.

[38] Caminero, A.C., Robles-Gómez, A., Ros, S., Hernández, R., & Tobarra, L. (2013). P2P-based resource discovery in dynamic grids allowing multi-attribute and range queries. Parallel computing, 39(10), pp. 615–637.

[39] Crespo, A., & Garcia-Molina, H. (2002). Routing indices for peer-to-peer systems. Proceedings of 22nd international conference on distributed computing systems (pp. 23-32). 10.1109/ICDCS.2002.1022239

[40] Marzolla, M., Mordacchini, M., & Orlando, S. (2005). Resource discovery in a dynamic Grid environment. Sixteen international workshop on database and expert systems applications. Copenhagen, pp. 356–360. IEEE.

[41] Yi-Hong, T., Kevin, L. and Ya-Ping, L. (2012). Organisation and management of shared documents in super-peer networks baseds emantic hierarchical cluster trees. Peer-to-peer networking and applications, 5(3), pp. 292–308.

[42] Liang, J., Kumar, R., & Ross, K. W. (2005). The kazaa overlay: A measurement study. Computer networks journal (Elsevier), 49(6).

[43] Jin, X., & Chan, S. H. G. (2010). Unstructured peer-to-peer network architectures. In Handbook of peer-to-peer networking (pp. 117-142). Springer, Boston, MA.

[44] Vapa, M., Auvinen, A., Ivanchenko, Y., Kotilainen, N., & Vuori, J. (2008, March). Optimal resource discovery paths of Gnutella2. In 22nd international conference on advanced information networking and applications (aina 2008) (pp. 546-553). IEEE.

[45] Kim, H., Kim, Y., Kim, K., & Kang, S. (2008, January). Restricted path flooding scheme in distributed P2P overlay networks. In 2008 international conference on information science and security (ICISS 2008) (pp. 58-61). IEEE.

[46] Talia, D. and Trunfio, P. (2005). Peer-to-Peer protocols and Grid services for resource discovery on Grids. Advances in parallel computing, 14, pp. 83-103.

[47] Puppin, D., Moncelli, S., Baraglia, R. and Tonellotto, N. (2005). A grid information service based on peer-to-peer. In proceedings of the 11th international Euro-Par conference on parallel processing. Lisbon,Portugal: Springer- Verlag Berlin Heidelberg, pp. 454-464.