

Research Article

Optimizing Resource Discovery Technique in the P2P Grid Systems

Wang Tun,¹ J. Pourqasem,² and S. A. Edalatpanah ²

¹Department of Computer, Changzhi University, Changzhi, Shanxi 046011, China

²Department of Research Center, Ayandegan Institute of Higher Education, Tonekabon, Iran

Correspondence should be addressed to S. A. Edalatpanah; saedalatpanah@gmail.com

Received 6 April 2019; Revised 22 August 2019; Accepted 14 November 2019; Published 4 January 2020

Guest Editor: Michele Nogueira

Copyright © 2020 Wang Tun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Distributed discovery service is a main concept in the scalable and dynamic grid environments. In this paper, based on the super-peer technique, we propose a new topology for the grid discovery service. The model is designed in such a way that each super-peer within the cluster has the routing indices (RIs) based on cobweb and uses the hop-count routing index (HRI) to select the best neighbor. Besides, each super-peer includes a cache table, which stores the query and the query results. Furthermore, from the point of view of the response time and the number of submitted messages, we compare the new model with an existing method. An illustrative simulation is also presented to show the efficiency and validation of the new technique.

1. Introduction

Grid systems generally solve the science and engineering problems in the large-scale environment and integrate the high volume of computing and storage resources, data, services, and applications that are distributed geographically [1]. Discovery service in grid is directly affected by the diversity, dynamic, heterogeneity, and distribution features, which can notably increase the grid efficiency [2]. The classical approaches that are utilized for grid discovery are based on the centralized [3–7] or hierarchical [8, 9] architectures. These methods collect the grid resource information and use the central index server to sustain them. The grid environments suffer from two main drawbacks: bottleneck and single point of failure [10–12]. To avoid the mentioned problems, the P2P model has been adopted [3, 13–15]. The P2P topology in grid environments consists of the structured and unstructured models [16–23]. The super-peer mechanism is a category of P2P technology, which has been proposed in [24], for more details see [25–28].

The modified breadth-first-search (BFS) and intelligent search (IS) methods have been introduced in [29]. Breadth-first-search [30] technique is an extended protocol of

Gnutella that is implemented locally and is based on keyword searching, which selects the neighbor peers randomly. In the IS method, first for each peer, a profile is built and then this profile is used for the sending query. Ghorbani et al. [31] presented a self-adaptive resource discovery in the unstructured P2P environment based on the feedback of the network nodes. They also used the learning automata (LA) algorithm [32] to educate peers in the discovery process for finding the best neighbor peer. Löser et al. [33], using the semantic clustering method [34, 35] in super-peer network, presented the semantic overlay clustering (SOC) to link the information provider peer to each cluster super-peer semantically.

Some approaches [2, 36–39] also index and summarize the resource features to reduce the amount of information that is kept and transferred in the discovery service.

Routing indices (RIs) as a P2P technique organize the indexed and summarized resource information of nodes [40]. The benefit of this approach is that queries are disseminated and forwarded only among the locations of the network where resources existed, thus avoiding to flood query requests to the nodes which are not useful. The main drawback of this technique is that this indexing system comes from the presence of cycles in the network graph. As

an extension of the RIs, hop-count routing index (HRI) is produced that keeps the resource information in a table structure at different hops [41–43].

Marzolla et al. [44] proposed a discovery technique in which the resource information of each domain and the summarized information of the neighbor domains are maintained by the brokers. This method uses the k -bit vector to represent the indices and summarize the information of resources. Puppini et al. [41] implemented a grid information service (GIS) that utilized the HRI. This method has two significant entities: (1) the agent that is in control of a super-peer builds the node's resource information and (2) the aggregator that receives the resource information from the mentioned agent and indexes it.

Caminero et al. [42] proposed the use of HRI as a means to route jobs within a P2P system. The main drawback of this model is that it only considers numeric parameters (such as effective bandwidth, the number of total machines in the domain, and the domain workload) to perform the resource discovery. Caminero et al. [43] presented a model that uses the HRI to construct the summary information based on cobweb tree [37]. This approach calculates the *goodness function* [40] for routing and forwarding the query to the neighbor peers, which likely have more probability for that query. Furthermore, to adapt the summaries with RIs, they proposed the technique so-called n -level summaries to filter attributes that have the same values, and their probability is less than a threshold.

In this paper, we present a new discovery method based on super-peer network in which each super-peer within the cluster has the routing indices (RIs) based on cobweb and uses the hop-count routing index (HRI) to select the best neighbor. Furthermore, each super-peer utilizes a cache table to store the query and the query results.

Remaining of this paper has been structured as follows: Section 2 expresses the configuration of the grid based on the super-peer network and Section 3 introduces the proposed resource discovery method in detail. Section 4 evaluates the simulation results based on the GridSim [45] that shows the effective performance of our approach. The final section concludes the paper and presents the guideline for the later works.

2. Configuration of Grid

Grid systems can use the super-peer topology for implementation of their infrastructure. As shown in Figure 1, the super-peers (SP) in each virtual organization (VO) or cluster operate as a server. Super peers receive the query from the client peer (CP) or the neighbor super-peers and answer them. The SPs communicate with each other and make an overlay network [46]. Generally, an SP has high capability among other CP in each VO/cluster that processes the query. When a CP connects to a VO/cluster, the local SP indexes the information of its resources. At leaving, the indexed information related to that CP is removed from the VO/cluster. In addition, if the resource information of a CP is changed, the updated information is sent to the mentioned SP. When a CP needs resources, it creates a query and

submits it to the SP. This SP seeks among indexed information to find appropriate resources. If the requested resources are found, the IP address of a CP, that is, the owner of the result, is sent to the requested peer; otherwise, the SP creates a copy of the request and forwards it to the overlay network. The SP of each cluster searches its domain to find the requested resources.

3. New Discovery Method

We present a new discovery technique based on the super-peer network. First, we create some clusters and organize the grid nodes. In each cluster, a super-peer maintains the summary of resource information that is owned by client peers. We use cobweb [37–39] to cluster and summarize the resource information. Each super-peer uses the RIs [40] to structure and keep the summary information of peers within the cluster. Table 1 is a sample of RIs based on Figure 2 that shows attributes and their probability values provided by the cobweb in each cluster. When a peer joins or leaves the cluster, the RIs entry related to it is created or deleted. Each cluster super-peer sends the maximum probability of each attribute to the neighbor super-peers. When the resource information of a peer changes, that peer sends the updates to its super-peer. Then the super-peer creates the summary again and updates the probabilities of RIs which has higher than the threshold. Next, the maximum probability of attributes is resubmitted to the neighbor super-peers. The super-peers use HRI [41, 43] to organize the summary information of neighbor super-peers in the grid environment. Table 2 shows the aggregated probabilities of the neighbor super-peers at different hops. It is noteworthy that the neighbor super-peers are selected based on the HRI information and not by random or flooding methods. This selection is based on *goodness function* [40] at the predefined hop-count. Consequently, the queries are forwarded to super-peers that are likely to match the query requirements and have nodes in their cluster that own the requested resources.

The goodness of neighbor super-peer (i) with respect to the query requirement is calculated as follows:

$$goodness(neighbor_i, Req) = \sum_{j=0 \dots h} goodness(N_i[j], Req), \quad (1)$$

where $neighbor_i$ is the i -th neighbor super-peer; h is the horizon of HRI (number of hops); $N_i[j]$ refers to the HRI entry for $neighbor_i$ at j -th hop; Req is the requested requirement; and *goodness* points to the probability of a neighbor super-peer in the (i, j) position of the HRI table with respect to Req . The cluster super-peer selects the best neighbor super-peer based on the maximum result of the goodness function and forwards the query to it.

Moreover, in our approach, each super-peer caches the submitted queries and their results in a cache table. The cache table can improve the grid performance by decreasing the delay, bandwidth usage, traffic, and the number of messages, which are sent in the discovery process. Table 3 is

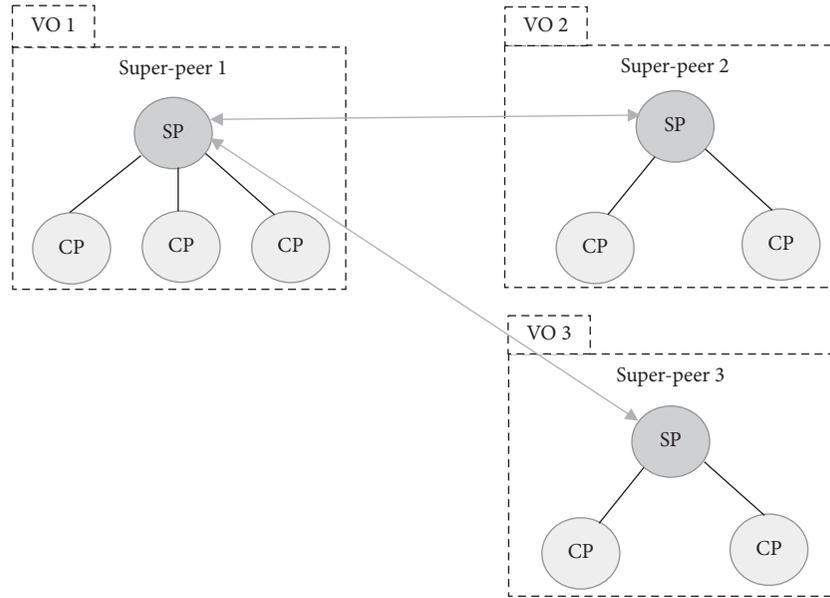


FIGURE 1: The configuration of a grid.

TABLE 1: A sample of RIs for resource information.

Attribute	OS name	Available memory	Hard disk	Architecture
Value	Windows	2-3 GB	>500 GB	x64
P1	—	25%	60%	40%
P2	30%	50%	45%	74%
P3	63%	80%	—	85%
P4	52%	25%	25%	64%

the cache table of a super-peer that contains the requirements of three queries in the form of attributes and the address of nodes that are the owner of requirements (result nodes). Each entry in this table is made in the local domain or is sent by the neighbor super-peers in other clusters.

3.1. Discovery Algorithm. In Figure 2, consider the cluster 1; let P1 need some computing resources to execute its project. The cluster super-peer receives a query and searches the local domain. If the responding peer finds, the super-peer sends the address of the result node back to the requested node and inserts an entry for that query or updates its cache table. Furthermore, it sends the update to the neighbor super-peers. If no results are found, the super-peer looks up the cache table before forwarding the query. If there is a related entry for that request, the super-peer answers the query locally and returns the query result to the requesting node. If not, the super-peer of the cluster 1 calculates the goodness function of its neighbor super-peers (cluster 2 and cluster 3) for that query. We assume that the goodness function of cluster 2 is higher than cluster 3. Therefore, the super-peer of cluster 2 first seeks locally to find the result and then looks up the cache table. If the query result is not in the cache table of cluster 2, the first best neighbor (say cluster 4) is selected and the query is forwarded to it. If the finding process is not successful, the query is bounced back to the parent super-

peer (cluster 2). Then, cluster 2 sends the query to the second best neighbor (cluster 5). If the search process is unsuccessful and as a result of cluster 2 has no other neighbor domain except cluster 1, the query is bounced back and cluster 1 forwards query to cluster 3 (the second best neighbor). When the algorithm reaches a response, the cache table is updated and forwarded to the neighbor clusters. Algorithm 1 shows our discovery approach.

3.2. Stability of New Approach. In this section, we compare our model with the existing model in [43]. To this end, consider Figure 3. This is a sample of P2P topology based on the model in [43], in which each peer is responsible for the discovery process. We can see in the mentioned model in [43] that there is a tree structure with 3-hop-count deep and 15 connected peers. For using our model, first, we eliminate all connections and reconfigure this topology based on super-peer as shown in Figure 4. This figure shows that there are three clusters in topology with 1-hop-count deep. The new configuration increases the scalability of discovery as shown in Figure 3.

Consider Figure 3; let a user in the domain of P0 send a query, and the node P0 searches locally to response it. If there is no required resource, P0 must send the query to the best neighbor peer. We consider two scenarios (optimistic and pessimistic statuses).

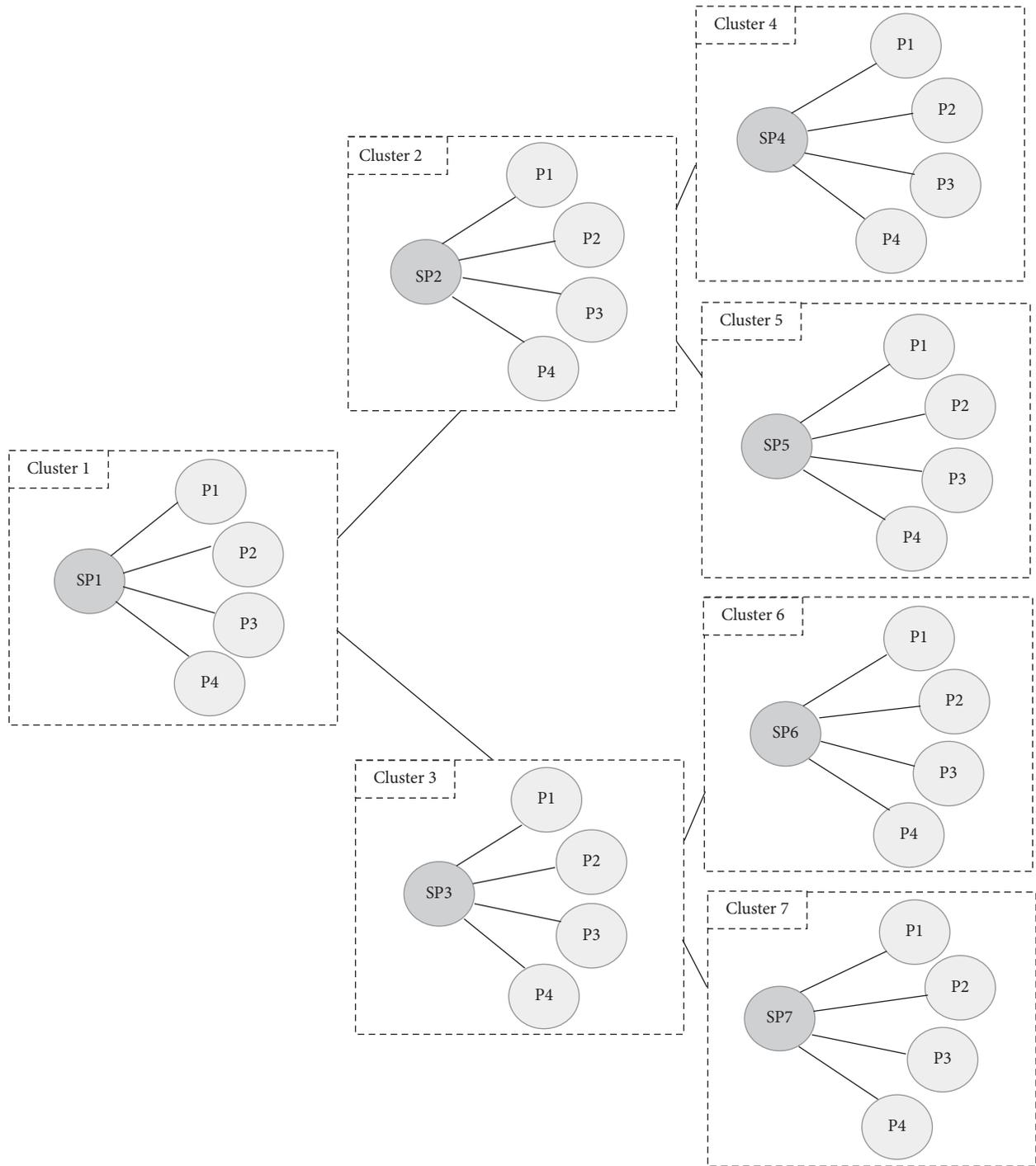


FIGURE 2: The P2P overlay connection between clusters.

TABLE 2: HRI for cluster 1 (SP1).

Attribute	1hop				2hop			
	OS name	Available memory	Hard disk	Architecture	OS name	Available memory	Hard disk	Architecture
Value	Windows	2-3 GB	>500 GB	x64	Windows	2-3 GB	>500 GB	x64
SP2	—	25%	45%	52%	39%	85%	50%	—
SP3	65%	26%	28%	32%	36%	54%	39%	47%

TABLE 3: A sample of the cache table.

Attribute	OS name	Available memory	Hard disk	Architecture	Query result
Query1	Windows	2-3 GB	—	—	URL
Query2	—	4-6 GB	>500 GB	X64	URL
Query3	Windows	—	—	X86	URL

```

(1)  $q$ : new incoming query
(2) ClusterResource: a resource in cluster
(3) Cache: a query result in cache table
(4) BestNeighbor: a neighbor super-peer selected by goodness function
(5) Neighbor: next neighbor super-peer
(6) for incoming  $q$  do
(7) ClusterResource = MatchQueryClusterResource ( $q$ )
(8) if (ClusterResource == null) then
(9)   Cache = MatchQueryCache ( $q$ )
(10)  if (Cache == null) then
(11)    BestNeighbor = HRI ( $q$ , Neighbor)
(12)    if (BestNeighbor == null) then
(13)      Receiver = Sender ( $q$ )
(14)    else
(15)      Receiver = BestNeighbor
(16)    end if
(17)    ForwardQueryToReceiver ( $q$ , Receiver)
(18)  else
(19)    SendResponseToRequester ( $q$ )
(20)  end if
(21) else
(22) SendResponseToRequester ( $q$ )
(23) Store/UpdateResultToCache ( $q$ )
(24) Send update to Neighbor
(25) end if
(26) end for

```

ALGORITHM 1: Pseudocode of the new discovery algorithm.

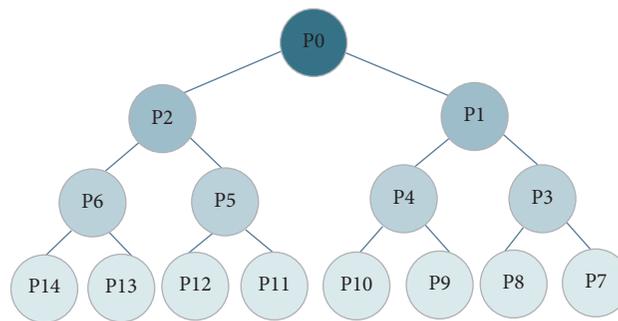


FIGURE 3: Tree topology of P2P network.

In the optimistic status, the response node is at 1-hop-count level (P1 or P2). We assume that P1 is the best neighbor for P0, then P0 sends the query to P1 and the discovery process successfully finishes. In the pessimistic status, P1 receives the query from P0 and searches locally. If P1 cannot find the requested resources, the query is sent to the best neighbor (say P3). We assume that node P3 is not the owner of the needed resources, and then it sends the query to the best neighbor peer (say P7). If P7 cannot

respond to the request, the query is bounced back. Then P3 sends the query to the second best neighbor (P8). We assume that P8 is not the owner of the needed resource, and then the query is bounced back to P1. P1 sends the query to the second best neighbor (P4), and P4 sends it first to P9, and if there is no response, then it submits it to the P10. Let the requested resources are not found in P10, so the query is bounced back to P0. Similarly, for the left side of P0, in the pessimistic status, we through the following routes

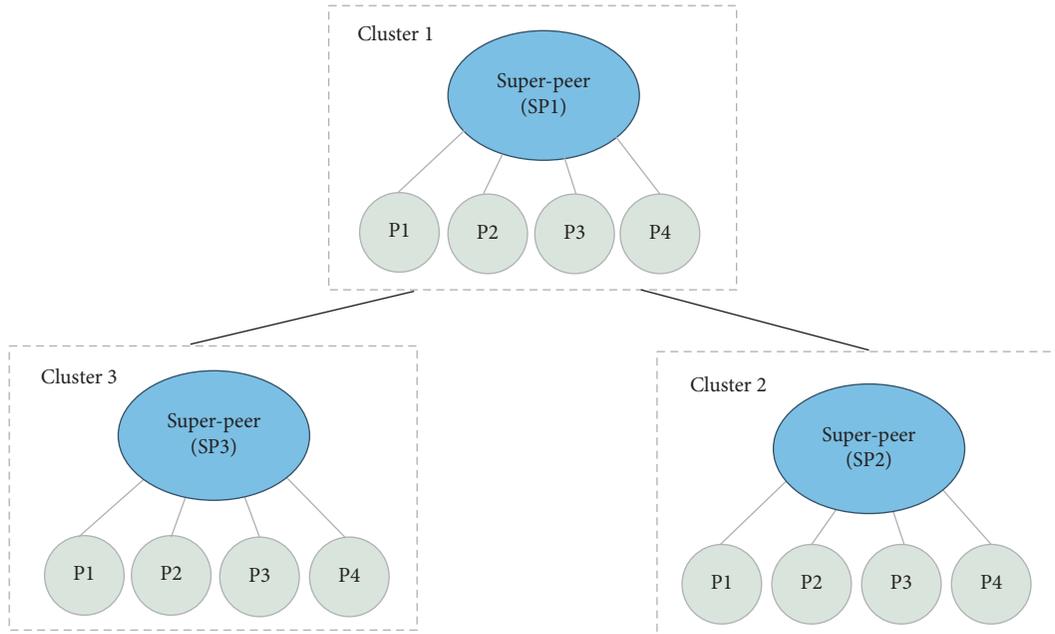


FIGURE 4: A sample of the clustered diagram of Figure 3.

$P2 \rightarrow P5 \rightarrow P11 \rightarrow P12 \rightarrow P6 \rightarrow P13 \rightarrow P14$. Therefore, considering P14 has the requested resources, the discovery process of the method [43] can be finished.

Next, consider Figure 4; let us assume that a user in cluster 1 sends a query to the cluster super-peer (SP1). The query process is begun, and the SP1 searches its domain locally. If it finds the requested resource, obviously the process is finished. Except for this situation, we consider two scenarios (optimistic and pessimistic statuses). Let SP1 forwards the query to the best neighbor (say cluster 2); for the optimistic status, the discovery process in cluster 2 is successful and the response node is found. In the pessimistic status, the search in cluster 2 is not successful, so the query is bounced back and forwarded to cluster 3. If this cluster finds the requested resource, the discovery process is finished successfully. It is well-known that the resource discovery scalability is related to the cost of the discovery process in terms of time and the query messages that are sent to reach the response. Therefore, based on the above demonstration, it is easy to see that the new approach is superior to the existing model in [43] from the scalability point of view. In the next section, we will simulate and compare the above two topologies (Figures 3 and 4) based on the scalability feature.

4. Simulation Results

In this section, we simulate the discovery process in the grid system. First, we simulate the discovery process based on the new method (Algorithm 1) and evaluate the query response time and the submitted messages in comparison with the discovery approach presented by Caminero et al. [43]. Second, we compare the scalability of discovery related to the two topologies presented in Section 3.2. Furthermore, the grid environment has been simulated by using the GridSim toolkit [45].

Suppose that 100 users cooperate in our simulation environment; the connection bandwidth is 100 Mbps; the propagation delay is 10 second, and the packet are transmitted by 1500 packet/sec maximally. The resources contain one machine that each of them has four processing elements and “Intel” architecture, and their operating systems are “Linux.” The network routers use the RIP protocol, and their scheduling method is FIFO. The first part of our simulation consists of three scenarios that the responding cluster selected randomly: (1) the result node is in the neighbor cluster at 1-hop count, (2) the result node is in the neighbor cluster at 2-hop count, and (3) the result node is in the neighbor cluster at 3-hop count. The second part of the simulation consists of two scenarios: (1) the result node is found in the optimistic status and (2) the result node is found in the pessimistic status.

Figures 5 and 6 compare the query response time and the number of forwarded discover messages in two grid environments: (1) the grid environment simulated based on the new model in which all super-peers use the cache table to response the queries and (2) the grid environment simulated with the model in [43].

The simulation results show that the using of the cache table improves the query response time and decreases the discovery delay. Furthermore, the results illustrate that the number of submitted query messages also decreases. We can see that when the number of hop count increases, the result of the new model is much better than the existing model in [43].

In the second part, the discovery processes are simulated in the optimistic status and the pessimistic status with two mentioned topologies. The simulated results show that using of new topology compared with the existing topology, in both statuses (optimistic status and pessimistic status), decreases the discovery time and improves the scalability of grid discovery (Figure 7).

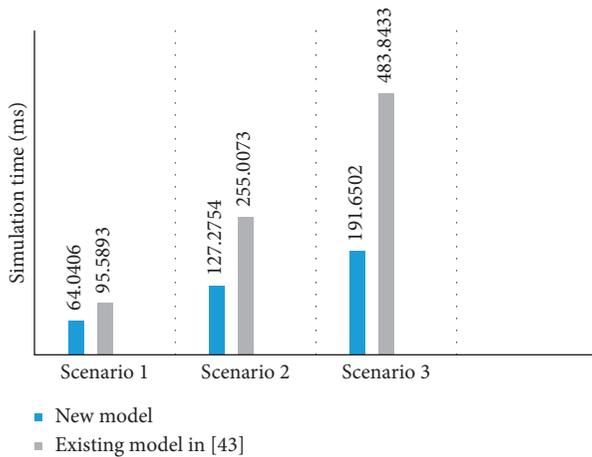


FIGURE 5: The comparison of the response time for the discovery process.

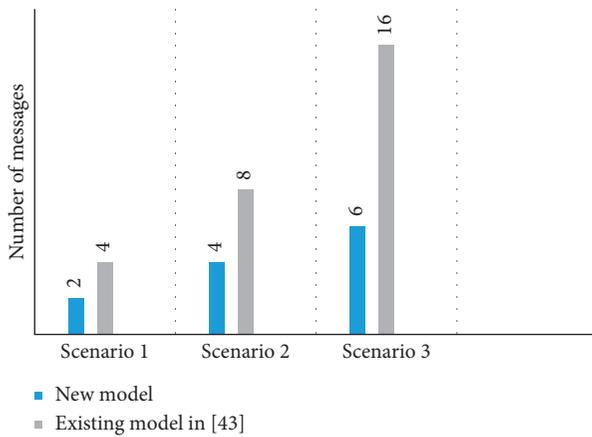


FIGURE 6: The comparison of the submitted messages in the discovery process.

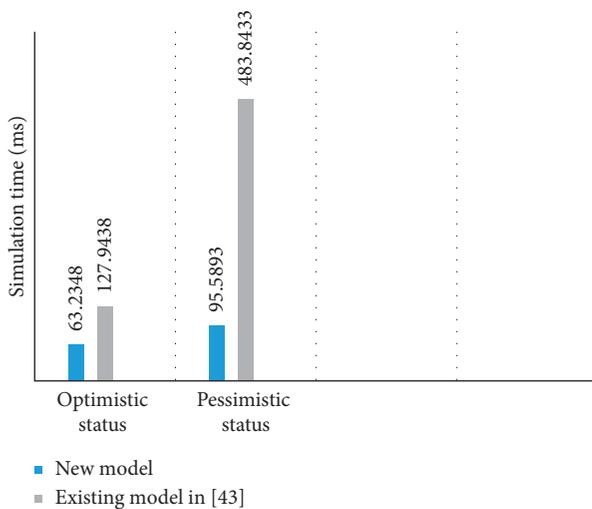


FIGURE 7: The comparison of scalability of the discovery process under optimistic status and the pessimistic status.

5. Conclusions and Future Works

This paper introduced a new model of grid discovery using P2P technology. For improvement of the scalability of the discovery process, the new model was configured based on the super-peer approach. Furthermore, to reduce the discovery delay, a cache table has been used in the model. The simulated results show that the new model compared with the existing method optimized the query response time and decreased the submitted query messages. For the future, we intend to use some intelligent clustering approaches in this topology and choose the super-peers based on the network attributes such as traffic and bandwidth of the network.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] V. Muppavarapu and S. M. Chung, "Semantic-based access control for grid data resources in open grid services architecture-data access and integration (OGSA-DAI)," in *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence*, pp. 315–322, IEEE, Dayton, OH, USA, November 2008.
- [2] L. Mohamad Khanli and S. Kargar, "FRDT: footprint resource discovery tree for grids," *Future Generation Computer Systems*, vol. 27, no. 1, pp. 148–156, 2011.
- [3] V. Nagarajan and M. Mohamed, "A decentralized two phase resource discovery model for peer-to-peer grid environments," *International Journal of Advanced Engineering Technology*, vol. 1092, p. 1095, 2016.
- [4] J. Zarrin, R. L. Aguiar, and J. P. Barraca, "Resource discovery for distributed computing systems: a comprehensive survey," *Journal of Parallel and Distributed Computing*, vol. 113, pp. 127–166, 2018.
- [5] P. Trunfio, D. Talia, H. Papadakis et al., "Peer-to-peer resource discovery in grids: models and systems," *Future Generation Computer Systems*, vol. 23, no. 7, pp. 864–878, 2007.
- [6] R. Guharoy, S. Sur, S. Rakshit et al., "A theoretical and detail approach on grid computing a review on grid computing applications," in *Proceedings of the 8th Annual Conference Industrial Automation and Electromechanical Engineering (IEMECON)*, pp. 142–146, Bangkok, Thailand, August 2017.
- [7] P. Bellavista, M. Cinque, A. Corradi, L. Foschini, F. Frattini, and J. Povedano-Molina, "GAMESH: a grid architecture for scalable monitoring and enhanced dependable job scheduling," *Future Generation Computer Systems*, vol. 71, pp. 192–201, 2017.
- [8] M. Mollamotalebi, G. Sadeghi, and M. Mehdi, "Weighted-HR: an improved hierarchical grid resource discovery," *Journal of Computer & Robotics*, vol. 11, no. 2, pp. 7–16, 2018.
- [9] E. Huedo, R. S. Montero, and I. M. Llorente, "A recursive architecture for hierarchical grid resource management,"

- Future Generation Computer Systems*, vol. 25, no. 4, pp. 401–405, 2009.
- [10] Y. Deng, F. Wang, and A. Ciura, “Ant colony optimization inspired resource discovery in P2P grid systems,” *The Journal of Supercomputing*, vol. 49, no. 1, pp. 4–21, 2009.
 - [11] S. Asghari and N. J. Navimipour, “Resource discovery in the peer to peer networks using an inverted ant colony optimization algorithm,” *Peer-to-Peer Networking and Applications*, vol. 12, no. 1, pp. 129–142, 2019.
 - [12] M. Hasanzadeh and M. R. Meybodi, “Grid resource discovery based on distributed learning automata,” *Computing*, vol. 96, no. 9, pp. 909–922, 2014.
 - [13] L. Wang, W. Jie, and J. Chen, *Grid Computing: Infrastructure, Service, and Applications*, CRC Press, Boca Raton, FL, USA, 2018.
 - [14] B. Gueye, O. Flauzac, C. Rabat, and I. Niang, “A self-adaptive structuring for large-scale P2P Grid environment: design and simulation analysis,” *International Journal of Grid and Utility Computing*, vol. 8, no. 3, pp. 254–267, 2017.
 - [15] C. Mastroianni, D. Talia, and O. Verta, “Designing an information system for grids: comparing hierarchical, decentralized P2P and super-peer models,” *Parallel Computing*, vol. 34, no. 10, pp. 593–611, 2018.
 - [16] N. J. Navimipour and F. S. Milani, “A comprehensive study of the resource discovery techniques in peer-to-peer networks,” *Peer-to-Peer Networking and Applications*, vol. 8, no. 3, pp. 474–492, 2015.
 - [17] A. Ahmadian, M. Zavvar, A. Saeedi, and F. Ramezani, “Resource discovery in non-structured peer to peer grid systems using the shuffled frog leaping algorithm,” *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 4, pp. 9–14, 2018.
 - [18] N. J. Navimipour, A. M. Rahmani, A. H. Navin, and M. Hosseinzadeh, “Resource discovery mechanisms in grid systems: a survey,” *Journal of Network and Computer Applications*, vol. 41, pp. 389–410, 2014.
 - [19] P. Felber, P. Kropf, E. Schiller, and S. Serbu, “Survey on load balancing in peer-to-peer distributed hash tables,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 473–492, 2014.
 - [20] Y.-H. Chen, E. J.-L. Lu, Y.-T. Chang, and S.-Y. Huang, “RDF-chord: a hybrid PDMS for P2P systems,” *Computer Standards & Interfaces*, vol. 43, pp. 53–67, 2016.
 - [21] J. Furness, M. Kolberg, and M. Fayed, “An evaluation of Chord and Pastry models in OverSim,” in *Proceedings of the Modelling Symposium (EMS)*, pp. 509–513, Manchester, UK, November 2013.
 - [22] B. Yin, X. Lu, J. Huang, and Y. Kang, “Analysis of topology dynamics for unstructured P2P networks,” *Computer Communications*, vol. 80, pp. 72–81, 2016.
 - [23] A. Hameurlain, D. Cokuslu, and K. Erciyas, “Resource discovery in grid systems: a survey,” *International Journal of Metadata, Semantics and Ontologies*, vol. 5, no. 3, pp. 251–263, 2010.
 - [24] B. B. Yang and H. Garcia-Molina, “Designing a super-peer network,” in *Proceedings of the 19th International Conference on Data Engineering (Cat. No. 03CH37405)*, pp. 49–60, IEEE, Bangalore, India, March 2003.
 - [25] A. M. Saghiri and M. R. Meybodi, “An adaptive super-peer selection algorithm considering peers capacity utilizing asynchronous dynamic cellular learning automata,” *Applied Intelligence*, vol. 48, no. 2, pp. 271–299, 2018.
 - [26] S. Hashemi Najafi, J. Pourqasem, and S. A. Edalatpanah, “The use of super node to process query in peer-to-peer networks,” *Science and Education*, vol. 1, no. 1, pp. 28–32, 2015.
 - [27] J. Pourqasem and S. A. Edalatpanah, “Verification of super-peer model for query processing in peer-to-peer networks,” in *Innovative Research and Applications in Next-Generation High Performance Computing*, pp. 306–332, IGI Global, PL, USA, 2016.
 - [28] Y.-H. Tan, k. Lü, and Y.-P. Lin, “Organisation and management of shared documents in super-peer networks based semantic hierarchical cluster trees,” *Peer-to-peer Networking and Applications*, vol. 5, no. 3, pp. 292–308, 2012.
 - [29] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, “A local search mechanism for peer-to-peer networks,” in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pp. 300–307, ACM, McLean, VI, USA, November 2002.
 - [30] A. A. Jamal and W. J. Teahan, “Alpha multipliers breadth-first search technique for resource discovery in unstructured peer-to-peer networks,” *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 4, 2017.
 - [31] M. Ghorbani, M. R. Meybodi, and A. M. Saghiri, “A novel self-adaptive search algorithm for unstructured peer-to-peer networks utilizing learning automata,” in *Proceedings of the 3rd Joint Conference of AI & Robotics and 5th RoboCup Iran Open International Symposium*, pp. 1–6, IEEE, Tehran, Iran, April 2013.
 - [32] M. Ghorbani, M. R. Meybodi, and A. M. Saghiri, “An adaptive topology management algorithm in P2P networks based on learning automata,” in *Proceedings of the 7th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pp. 1–4, IEEE, Tehran, Iran, January 2019.
 - [33] A. Löser, F. Naumann, W. Siberski et al., “Semantic overlay clusters within super-peer networks,” in *Proceedings of the International Workshop on Databases, Information Systems, and Peer-to-Peer Computing*, pp. 33–47, Springer, Berlin, Heidelberg, September 2003.
 - [34] A. Kutuzov and E. Kuzmenko, “Neural embedding language models in semantic clustering of web search results,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 3044–3048, Portorož, Slovenia, May 2016.
 - [35] Y. He, C. S. Yang, L. C. Yu et al., “Sentiment classification of short texts based on semantic clustering,” in *Proceedings of the 2015 International Conference on Orange Technologies (ICOT)*, pp. 54–57, IEEE, Hong Kong, China, December 2015.
 - [36] R.-S. Chang and M.-S. Hu, “A resource discovery tree using bitmap for grids,” *Future Generation Computer Systems*, vol. 26, no. 1, pp. 29–37, 2010.
 - [37] D. H. Fisher, “Knowledge acquisition via incremental conceptual clustering,” *Machine Learning*, vol. 2, no. 2, pp. 139–172, 1987.
 - [38] A. C. Caminero, E. Huedo, O. Rana et al., “Summary creation for information discovery in distributed systems,” in *Proceedings of the 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pp. 167–171, Ayia Napa, Cyprus, February 2011.
 - [39] R. Brunner, A. C. Caminero, O. F. Rana, F. Freitag, and L. Navarro, “Network-aware summarisation for resource discovery in P2P-content networks,” *Future Generation Computer Systems*, vol. 28, no. 3, pp. 563–572, 2012.
 - [40] A. Crespo and H. Garcia-Molina, “Routing indices for peer-to-peer systems,” in *Proceedings of the 22nd International*

- Conference on Distributed Computing Systems*, pp. 23–32, Vienna, Austria, July 2002.
- [41] D. Puppim, S. Moncelli, R. Baraglia et al., “A grid information service based on peer-to-peer,” in *Proceedings of the 11th International European Conference on Parallel Processing*, pp. 454–464, Springer, Lisbon, Portugal, August 2005.
- [42] A. Caminero, O. Rana, B. Caminero, and C. Carrión, “Network-aware heuristics for inter-domain meta-scheduling in grids,” *Journal of Computer and System Sciences*, vol. 77, no. 2, pp. 262–281, 2011.
- [43] A. C. Caminero, A. Robles-Gómez, S. Ros, R. Hernández, and L. Tobarra, “P2P-based resource discovery in dynamic grids allowing multi-attribute and range queries,” *Parallel Computing*, vol. 39, no. 10, pp. 615–637, 2013.
- [44] M. Marzolla, M. Mordacchini, and S. Orlando, “Resource discovery in a dynamic grid environment,” in *Proceedings of the Sixteen International Workshop on Database and Expert Systems Applications*, pp. 356–360, Copenhagen, Denmark, July 2005.
- [45] A. Sulistio, U. Cibej, S. Venugopal, B. Robic, and R. Buyya, “A toolkit for modelling and simulating data grids: an extension to GridSim,” *Concurrency and Computation: Practice and Experience*, vol. 20, no. 13, pp. 1591–1609, 2008.
- [46] C. Mastroianni, D. Talia, and O. Verta, “A super-peer model for building resource discovery services in grids: design and simulation analysis,” in *Proceedings of the European Grid Conference on Advances in Grid Computing*, pp. 132–143, Springer, Amsterdam, Netherlands, February 2005.

