

Resource Discovery Approaches for Grid Environments

Mahdi Mollamotalebi^{1,*}, Raheleh Maghami¹, Abdul Samad Ismail²

¹Department of Computer Engineering, Buinzahra Branch, Islamic Azad University, Buinzahra, Iran

²Faculty of Computing, Universiti Teknologi Malaysia, 81310, Skudai, Malaysia

Abstract Resource discovery approach is one of the most important services affecting grid performance. There are many aspects of characteristics of global grids such as dynamicity, scalability, and heterogeneity. The resource discovery approaches face with many challenges such as latency, quality of results, efficiency, reliability, maintenance cost and so on. This paper focuses on introduces the resource discovery approaches used in grid computing environments along with their behaviors in different conditions.

Keywords Grid Computing, Resource Discovery, Centralized, Peer-to-Peer, Agent

1. Introduction

The grid computing handles complex applications in a distributed management manner. The grid applications send their resource requirements for execution to the grid resource management system and wait to take the resource references. The efficiency of resource discovery is critical for grid applications since lack or latency of response fails the grid applications execution.

In recent years different approaches are provided for resource discovery in grid environments. They can be classified to centralized, Peer-to-Peer-based (P2P-based), and agent-based approaches. The centralized approach is based on gathering all resources information and keeping them in one or a limited number of central index servers. Primitive resource discovery techniques were based on central databases following the client/server structure of computer networks[1]. Each grid node shares some resources in the grid by publishing its resources information to central server(s). Moreover resource changes caused by resource allocations are sent to the central servers either immediately or periodically. Periodic updates are used to decrease the message overloads however it can make the indexing servers information unreliable in very dynamic conditions. Central servers analyze received request and find the required resources in their own database, thus the requested resources can be located quickly and accurately [2].

Most of centralized methods exploit existing information directory protocols such as LDAP (Lightweight Directory Access Protocol) and UDDI (Universal Description Discovery and Integration) to handle their resource

information index[3].

If central servers leave the network, a lot of critical information of whole grid (e.g. resource information) will be lost. Moreover because all messages of gathering, updating and querying the resource information are transferred through the central servers, they suffer from burdensome message and processing loads. These problems raise more in large-scale environments[1, 4, 5]. Hierarchical administration of centralized servers declines the overloads by distributing the loads on different servers hierarchically but it cannot solve the problem completely. In this manner, index servers at each level of the hierarchy gather the resource information from their descendants and forward them to their parent index server. Whatever a node is located in a higher levels it is busier and contains the information about more resources in the network. Therefore it has a more critical role in the resource discovery system. Each node in the hierarchy is responsible to answer the queries received from its descendants. This structure is prevalent in many of current resource management systems such as GT MDS, gLite, etc.[6, 7].

P2P-based approach dictates a fully-distributed and cooperative network design without any supervision. Resources information are not indexed on central server(s), rather each peer has the role of both client and server and acts as autonomous and self-organized. A resource requester sends the request message to its neighbors and the neighbors subsequently forward the request to each other until reaching the required resource. P2Ps can be categorized as unstructured and structured. In unstructured case, peers are connected to each other without topology or indexing constraints. In this case, the resource discovery is usually handled by flooding mechanisms[8, 9].

On the other hand, structured P2P-based systems often use distributed hash tables (DHT) to index resource information[10, 11]. Peers record some information about their neighbors and previous successful search results to be

* Corresponding author:

motalebi@gmail.com (Mahdi Molla Motalebi)

Published online at <http://journal.sapub.org/ijnnc>

Copyright © 2013 Scientific & Academic Publishing. All Rights Reserved

exploited subsequently[12]. Furthermore some P2P-based techniques act as hybrid. They use both centralized and P2P structure to benefit their advantages concurrently. In this manner, there are some nodes acting as super-peers which manage a set of regular peers. But the super-peer nodes are related to each other as P2P.

In the agent-based approach, agents are used to disseminate and search the resources. The agent is a piece of software with the particular characteristics such as autonomy, ongoing execution, and mobility. It can be exploited in grid environments to increase the flexibility of resource discovery[13]. Each agent can be mapped into one grid node and it maintains grid node's resource information, or it navigates the network for requested resources. The autonomy of agents implies that they can exhibit intelligent behavior, and mobility ensures that they can operate in dynamic and heterogeneous environments. Furthermore, they can be used in a gradual discovery process and satisfy different requirements. On the other hand, the indexed resources information in this approach is modified periodically once the deployed agents return back to the indexing nodes. They use flooding to disseminate the resource information and resource requests among the network.

The rest of the paper is organized as follows. Section 2 to Section 4 introduce grid resource discovery techniques based on centralized, P2P and agent approaches respectively. Section 5 presents a comparison of different approaches in terms of the most important factors of efficiency, and Section 6 concludes the paper.

2. Grid Resource Discovery Techniques Based on Centralized Approach

2.1. Yu et al. (2003)

The centralized grid resource management system developed by Yu and his colleagues[14] uses an XML formatted message to provide the service. Users publish their resource information in the central repository. Also resource information updates are sent to the central repository by the resource owner. It also includes a query processing web service to discover requested resources.

The resource repository index and query processing module are centralized, so the system suffers from bottleneck and single point of failure especially in large scales. Moreover, it is not adapted to a high dynamic environment and also it cannot support dynamic-attribute queries. But it supports multi-attribute and ranged-attribute queries.

2.2. Elmroth and Tordsson (2005)

This resource discovery system[15] benefits from Grid Laboratory Uniform Environment (GLUE) project[16] which is a model to describe Grid resources information in terms of computing and storage elements. A module named

Information-Finder is responsible to discover the required resources and retrieve details of resources information such as their usage policies. Once receiving a query, the Information-Finder performs the resource finding by querying index servers. After finding the resources matched to the query, it gets detailed up-to-date resource information, so all static and dynamic information of the resources can be retrieved.

The indexing servers' addresses are also kept and searched in Information-Finder as hierarchical. The parallel mechanisms of resource discovery and information retrieval are controlled by a thread pool to avoid the overloads. Moreover, a time limited cache is used to keep the information of frequently requested resources in order to decrease the resource discovery overheads.

Using the hierarchical structure can reduce the bottleneck probability and subsequently increase the scalability. But the system still suffers from the single point of failure since leaving a hierarchical server results a potentially large set of the resources to be eliminated from the search domain. On the other hand, it supports all types of queries because it does not use hash functions and the latest resource information is verified by direct inquiry.

2.3. Ramos et al. (2006)

They proposed a hierarchical technique[17] which the grid environment is divided into different virtual organizations. Inside virtual organizations, there exist some hierarchical related master nodes each including a number of slave nodes. The master nodes update the resource database and the slave nodes gather the resource information from grid nodes. The slave nodes can be looked up by checking a description file of the slave nodes. The description file is an XML document kept in the master node and includes IP addresses and names of all the slave nodes related to the master node.

The resource discovery is verified by a configuration file containing the requested resource information. This file is disseminated between all slave nodes and a new service containing the search is issued. Once a slave node receives such a query, it checks requested resources satisfaction and returns the result to its master node. Now the search process in master node processes the received files and returns the answers to the requester.

Because the resource discovery is directed by master nodes, occurring failure in master nodes potentially makes some slave nodes became invisible to the system. The system also supports all types of queries.

2.4. Kaur and Sengupta (2007)

They provided a centralized grid resource discovery technique[18] consisting a query model which uses UDDI (Universal Description Discovery Integration) standard[19] to handle resource information as key-value pairs. It also utilizes GWSDL (Grid Web Services Description Language) to describe grid resources and SOAP (Simple Object Access

Protocol) to accomplish communications between web services in the grid. Furthermore it benefits from HTTP (Hyper Text Transfer Protocol) to provide the interface for issuing the resource requests and responses.

Considering the use of central databases to handle the web services, the system suffers from bottleneck and single point of failure. Moreover, the dynamic-attribute queries cannot be supported because UDDI cannot handle frequently changing information such as CPU load. But it supports ranged-attribute and multi-attribute queries.

2.5. Molt et al. (2008)

Molt and his colleagues[20] proposed a service oriented architecture developed on top of Globus Toolkit[21, 22]. It can fill the gap between a simple user interface and complex Globus resource management primitives. It uses standard protocols such as SOAP and XML (eXtensible Markup Language) for data exchanges. The system accesses the resources information through an abstraction layer. It gathers the resource information and save it in GIIS (Grid Index Information Service) or BDII (Berkeley Database Information Index) repositories. The static information is also kept in caches to be used in subsequent requests.

All the processes are handled by a central server, so the central server is a potentially bottleneck and single point of failure in high dynamic environments especially in large scales. Multi-attribute and ranged-attribute queries are supported but since dynamic attributes are updated in the central database in discrete intervals, so the dynamic-attribute queries cannot be supported efficiently.

2.6. Yan et al. (2008)

They proposed a hierarchical resource discovery system based on the small-world network. In a small-world network, each node has different ranged (short or long) contacts with near and far nodes. The query forwards are handled by long ranged contacts through message passing between several nodes. Also small-world networks are specified by the length of path. Resource locating is performed by logarithmic number of steps. The Grid in this technique takes hierarchical tree structure which the upper node supervises its descendant nodes. The upper level nodes' bandwidth is higher than lower ones and they deal with much more processes. Different levels have their own administration node to meet the autonomous managements. Moreover the structure utilized redundancy to tackle with single point of failure and keep the data consistent. Integration of small worlds into clusters makes the messages distribution faster than typical networks. Most of the requests are satisfied by near nodes without needs to traverse the unpopular/remote nodes[23].

Two strategies, generic and senior, are used for searching the resource. In generic strategy, users are informed about clusters' resources and are able to access them directly. So it removes the investigation of clusters which are not related to the request. Senior strategy allows more attributes for

queries and uses shortcuts in small-world networks which provides higher success rates and lower query cost. Considering the lack of centralized databases in this technique, it is almost scalable. Moreover it is not dealt with bottleneck problems which increase its reliability.

3. Grid Resource Discovery Techniques Based on P2P Approaches

3.1. Andrzejak and Zhichen (2002)

They proposed a peer-to-peer grid resource discovery system based on prior CAN[24] peer-to-peer system so that the CAN is extended handling range queries. Assuming that each resource can be described using a set of attributes with globally known types, users interested in a certain resource issue a query combined of needed attribute values or ranges. To locate different attribute resources, the algorithm searches each attribute in the relative indexing infrastructure and then combines the results. Node leaving causes the owner of one of the neighboring zones take over the zone owned by the departing node. A subset of servers that participates in the grid acts as Interval keeper (IK) in a CAN-based peer-to-peer network and records the pairs (attribute-value, resource-ID). The mapping between the intervals and the zones is an important entity of range queries. Grid nodes submit their resources information to their related IK and subsequently the queries will be processed within the IKs. In the worst case, queries may be forwarded to all IKs using BFS on a hypercube[25].

This technique is scalable since it specifies a threshold value for the responsibility of IKs so that in the case of exceeding the threshold value, a new IK will be added to the same control range. The status of resources is updated in IKs as periodic. There is low probable to fall on single point of failure problems due to distribution of the resource information among the IKs and their replicas. Also, there are no false positive errors due to distribution of queries among all relevant IKs without limitations which makes the technique scalable. The algorithm supports range-attribute queries by dedicating subsets of servers for specific ranges of attributes. For a multi-attribute query, each attribute will be mapped to a different DHT and the resources are advertised by reporting to the appropriate DHT servers[13, 25].

3.2. Cai et al. (2003)

They proposed a Multi-Attribute Addressable Network (MAAN) resource discovery technique that is an extension of its prior Chord[26] peer-to-peer system so that each MAAN node is an instance of a Chord system. Chord had offered efficient and scalable lookup services, but it did not support range-attribute and multi-attribute queries which MAAN support them by extending Chord with locality preserving hashing and a recursive multidimensional query resolution mechanism. Assuming that resource information

is mapped by Chord, each node is responsible to maintain resource information of its key-space. A single attribute query is handled by the Chord search services, but a multi-attribute query is handled by dividing the complex query into sub-queries and issuing them within the proper space. Another approach issues a complex query like a single query such that it performs searching of a resource that can satisfy all the attributes solitarily[27].

Due to using of Chord algorithm as its underlying peer-to-peer system, the time and message loads are of logarithmic orders. Moreover the load of nodes is balanced so that it is not dealt with the bottleneck problem. Because its search complexity is lower in order than the number of nodes, so it is scalable. There is no false-positive error due to query distribution using a predefined schema. MAAN handles a multi-attribute query by building multiple DHTs for each attribute. The weakness of MAAN is that since the queries are disseminated to one neighbor at a time, leaving of a node can cause query loss. Also dynamic-attribute query is not supported because the DHT is produced once and is updated at discrete intervals[13].

3.3. Iamnitchi and Foster (2004)

In their proposed technique, each peer publishes its resource information to some local servers. Request messages that are sent to the network contain a Time-To-Live parameter to terminate the message forwards after a limited number of steps. If receiving peers do not contain the desired resource, they forward the message to one of their neighbors and the process continues until finding desired resources or termination by Time-To-Live. Four routing strategies for queries are proposed by this method: random walk, learning based, best neighbor, and learning-best neighbor.

In the random walk, the peer to which a query is forwarded is chosen randomly and it is not needed to store extra data on peers. But in learning-based, peers learn from prior searches and record the satisfied queries in their index. Subsequent queries are forwarded to the peers that satisfied similar prior queries. If no similar queries exist, the received query will be forwarded to a random set of peers. In best-neighbor, the number answers received from each peer is recorded and subsequent queries are forwarded to the peers who answered the largest number of queries. The learning-best neighbor is similar to learning-based strategy except that if no similar queries found in the index, the query is forwarded to the neighbor with the most number of previous answers[28].

The information distribution in this technique removes bottleneck and increases the scalability. It does not deal with single point of failure and it supports all main types of queries. However the usage of TTL can eventuate to false-positive errors means that it may return unsuccessful answers even if the resource exists in the network[13].

3.4. Mastroianni et al. (2005)

In this technique some nodes act as super-peers. Resource owner nodes are connected to super-peers. They publish their resource information and also forward their resource queries to their related super-peer. Mastroianni and his colleagues supposed that each physical organization has only one super-peer that accomplishes two main tasks: it enables communication with other physical organizations, and it maintains local physical organization's metadata. Resource queries are sent to the local super-peer and it checks the local index for the required resource existence in its local nodes. If the required resource is found, it sends a query-hit message along with the resource owner address to the requesting node. The forwarding messages between super-peer nodes are limited by Time-To-Live to keep the time and message loads low[29].

The structure of this method is balanced and there is no bottleneck in the resource indexing, but the dynamic nature of nodes may affect badly on the quality of results. If super-peers leave the grid, some indexed information of resources can be inaccessible whereas the resources exist in the grid. Also the search process deals with the false-positive error due to using TTL for query forwards. It can support all types of resource queries[2, 13].

3.5. Marzolla et al. (2005)

They provided a super-peer based technique using the concept of Workload Management Systems (WMS) definition. It acts as an indexing service for a subset of virtual organizations. Resource attributes may be constant or varying over times. For locating the resources, it uses bitmap indexes as a simplified version of histograms which partitions relation attributes to buckets and stores them. Compressing the information of buckets enables fast answering by query evaluation. Each WMS keeps the information about its neighboring resources by checking the bitmap indexes of its neighbors. Users submit their queries to a WMS and it forwards the request to other nodes and collects the results. Queries are always forwarded to a subset of neighbors excluding the sender using breadth first search approach. Each node W forwards the query Q only to neighbors which their bitmap indexes satisfy the bitmap representation of Q . The satisfaction is checked by applying the logical AND on the bitmap representations of the query and the neighbor resource information. So the queries are not routed WMSs whose resources surely does not match them[30].

If the value of a resource changes, its owner WMS modifies the bitmap index related to resource. After modification, if the bitmap is same to the old one, it will not be published otherwise it is published to all the neighbors.

The worst-case message complexity in this algorithm is $O(S)$ where S is the number of super-peers. The super-peers are related in a tree schema and they have nearly same loads with low probability of the bottleneck. Nevertheless, if one super-peer left the grid, it can cause the tree connectivity failure, and subsequently prevent a large portion of the grid

being investigated even if resources are existed. Therefore, this technique has the weakness of potential single point of failure. Range-attribute and multi-attribute queries are supported, but dynamic-attribute queries are not supported due to periodic advertising of resources information[13].

3.6. Talia et al. (2007)

This technique proposes a distributed hash table (DHT) based resource discovery system suitable for large-scale grids. It supports requests for multiple resources by utilizing multiple DHTs. There is also a general purpose DHT to handle dynamic attribute queries. In this DHT, there are not maps between peers and resources because of frequently changes in resource values. So the requests related to this DHT floods to all of its peers.

Each node has the same probability to receive queries, so there is no bottleneck in the system. The order of time and message complexity is also less than or equal to the number of nodes which causes the system to be scalable. Each node is responsible by itself, so single point of failure does not exist in terms of the query result, but during query relays each node can be a single point of failure. Moreover, the queries are routed using DHT, so a false positive error cannot occur. Using DHT broadcast for a dynamic attribute query can ensure that the result is always up-to-date and thus it can support dynamic attribute queries. The system also can support multi-attribute and range-attribute queries due to use of multiple DHTs[13, 31].

3.7. Filali et al. (2008)

Filali and his colleagues provided an unstructured P2P-based mechanism for grid resource discovery without central administration on resource indexing and discovering. The resources information are also cached temporarily to be used in search processes. The nodes broadcast their resource information periodically to their neighbors. The neighbors record the resource information in their cache to be used by subsequent queries. If the information of cache is not useful, the requesting peer sends the query to the network as flooding.

The main idea is looking for resources by a push-pull model which resources are advertised and information is cached. If a peer's cache includes useful information, the request will be forwarded to the specific locations which satisfy the requests. Upon reception, the cache of the incoming link is updated. Requester nodes associate reservations for the found resources. So the reserved resource will be unavailable for other requesters for a specific period of times. Because there is not any central coordinating among the nodes, deadlock is probable. It introduces three states including free, reserved and allocated for resources to avoid the deadlock. The resources are free by default. A reserved resource has satisfied a query but it is not doing any task yet. It becomes allocated once a task starts. But, if no task is started in a specified time, the resource switches again to state free[32].

This technique is completely distributed and there is not any bottleneck. Existence of cache makes the search process faster, but in large-scale environments the scalability can be limited. It acts as suitable in dynamic networks because node's leaves can not interrupt sending the flooded queries. The false-positive error can be occurred due to using TTL parameter to limit the domain of message flooding.

3.8. Li and Qi (2009)

The resource discovery techniques which are based on DHTs are just able to handle exact matches for resource names through single hash keys. Though this is suitable to P2P systems but in grid environments it is not realistic to use a single hash key for different attributes of resources with ranged values. This technique makes a shared tuple space on the structured P2P aiming to reach higher scalability and support the ranged queries. The use of tuples allows handling the range queries. Each node shares some resources. The nodes accesses to local tuple space interfaces to get the resources information from global tuple space. The tuple space passes the calls of underlying DHTs[33].

Though the logarithmic rate of this technique's message load causes it to be scalable but it still only supports direct match queries. Thus the grid applications must know the exact resources to be located which limits its usability.

3.9. Mokadem et al. (2010)

They proposed a super-peer based resource discovery system using hierarchical distributed hash table (DHT) to reduce both lookup and maintenance costs especially during peers join and leave. They focused on search the metadata describing the data sources in a database context.

Their proposed system has a tree structure and only super-peers are connected to the top level of the hierarchy. Each leaf peer sends the list of its neighbors periodically to only one second level peer, and other second level peers update their super-peer neighbors' directory during the resource discovery. For each virtual organization, there is one DHT. Also super-peers act as the proxy for simple peers and establish a structured DHT. Each super-peer forms an entry point for its relative virtual organization. The resource queries can be intra-VO and inter-VO. The intra-VO uses normal DHT search and it is not used for super peers. The requests are first routed to local super-peer responsible of current VO by intra-VO mechanism. the inter-VO lookup is used to lookup other super-peers' DHTs if it is necessary[34].

This algorithm decreases maintenance overheads in DHTs by reducing lookup and maintenance costs. It also acts efficient when the nodes join and/or leave the grid frequently.

3.10. Palmieri (2010)

It is a scalable discovery framework to locate and retrieve shared replica information in a random unstructured P2P

grid. It is based on local caching. It uses bond percolation which is a transition model adapted to random walk locating. It protects the low connected nodes against high traffic problems. It reduced the total traffic scaling sub-linearly with grid size. Using the percolation search allows to distinguish several types of peers acting at various scales. Some nodes are connected to a few number of neighbors and forward low number of requests but the nodes with high number of neighbors deal with a lot of message forwards.

A mapping is constructed such that an open grid edge of the percolation model corresponds to existence of the active peer connections between grid nodes. Also, an open percolate path represents the search tree on the P2P grid infrastructure and the random-walk search method is adopted in this technique. In this method, after issuing a resource request, a walker is deployed from the initial node and traverses the network until reaching a matched resource. In the search process, replicated data are kept in the cache of high degree peers and percolation is able to locate the resources with high probability[35].

4. Grid Resource Discovery Techniques Using Agent-based Approach

4.1. Ding et al. (2005)

They proposed an agent-based resource discovery system so that an agent is mapped into a node and cooperates to locate required resources. Agents maintain AIT (Agent Information Table) to record their neighbor resource information. When new resources are submitted, their agent disseminates the resource information to neighboring agents. If a request is received by an agent, first it checks the local AIT for the required resource. If it is not found, the agent investigates the AITs of the neighboring nodes. If the resource exists in one of them, the query will be forwarded to that node; otherwise the query is forwarded to all neighbors using flooding approach[36].

This agent system can bridge the gaps between users and resources and makes the system scalable since dissemination and discovery are carried out step by step among neighboring agents. It is also adaptive to high dynamic grid environments because agents are able to adjust their dissemination and discovery behavior. It does not deal with the bottleneck and single point of failure, but flooding can limit the scalability of the system. Lacking of limiting factors for the flooding may cause the system taking long times to converge in some cases. Due to flooding the queries as parallel, dynamic nature of nodes cannot negatively affect the searching process. Also because it does not use TTL for flooding, false-positive errors are not foreseeable[13, 36].

4.2. Jian et al. (2006)

This technique is based on mobile agents and an information server creates mobile agents according to the

requested resource characteristics. It also defines resource bindings and route rules. The resource information management is handled by the collaboration of mobile agent and LDAP protocol to get local and global resources information. The shortest distance routing and transferring (SDRT) system is used in mobile agents due to its direct effects on the system performance. An information server builds related mobile agent using resource features, defines binding conditions, sends mobile agents, and analyses data brought back by mobile agents. Mobile agents transfer according to routes between the nodes and alternate with the system agent in each node[37].

The scalability of this technique is limited due to using centralized information server. In the view of relaying nodes, there is no bottleneck due to hold the query migration by resource nodes. Mobile agents migrate according to a routing rule and on a specified path, so the dynamicity of the nodes perturbs query migrations.

4.3. Kang and Sim (2012)

Kang and Sim provided a protocol using multiple agents for brokering to handle the very high amounts of resources information and improve the scalability. This protocol utilizes three kinds of agents including user-agent, provider-agent and broker-agent. The user-agents act on behalf of users. They provide an interface enabling users to locate proper Grids providing desired resources. The provider-agents act similarly as user-agents but they act on behalf of providers, and broker-agents connect user-agents and provider-agents. They match the users' requests and providers' resources.

The protocol also uses agents to handle overload of requests submitted to broker-agents and failing of provider-agents. In the case of request overloads, a broker-agent can redirect the requests to other broker-agents. It also recognizes the provider-agents failure and removes them from the selection list. Using multiple broker-agents keeps the protocol scalable since the broker-agents are able to exchange recommendation messages even with broken connection between user-agents and provider-agents in one broker-agent[38].

5. Comparisons

The behavior of different approaches introduces in above sections are compared in terms of the most important factors, i.e. scalability, dynamicity, reliability, quality of results, and maintenance costs in the Table 1. The quality of results is related to the accuracy and number of found results. Also, the maintenance costs are mainly related to the operations that are necessary to be performed in order to keep and update the resource information in the indexing nodes of the resource discovery system. The reliability is evaluated based on the possibility of occurring single point of failure, bottleneck, and false positive errors. Moreover, the scalability refers to ability of a resource discovery

system to work as efficient in large-scale (e.g. more than 10000 nodes) Grid environments. Furthermore, dynamicity is evaluated by the adaptability of a resource discovery system to dynamic conditions of the Grid environments such as more than 20% join/leave requests for nodes/resources among all the issued requests.

Centralized/hierarchical techniques are able to find the resources quickly and accurately in small scale environments because they keep and search the resource information in their integrated centralized databases. But taking into account the high message and processing loads on the limited central indexing servers, they cannot act efficiently in large-scale environments. Hierarchical deployment increases the scalability of centralized approach because it uses multiple servers as hierarchical and distributes the loads more than centralized approach.

They also deal with single point of failure problem in the central servers which make them unreliable. Also dynamic attributes of resources cannot be updated rapidly especially in hierarchical techniques due to latency of applying the resource updates to higher level indexing nodes of the hierarchy.

The P2P-based techniques are more reliable and scalable than centralized ones because of resource information distribution. Structured P2P-based solutions are scalable

because they use efficient distributed hash tables but with pay high costs of maintenance. Unstructured P2P-based solutions are not scalable because of using flooding mechanisms which dramatically increases the number of messages and latency. Also the hybrid deployment of P2P is more scalable relative to unstructured P2Ps but it still suffers from potential bottleneck in the point of super-peer nodes.

Unstructured P2P-based and agent-based solutions are adapted to dynamic environments because they send the queries to the target resource owners to be analyzed, so the last up-to-date resource status would be investigated. But super-peer-based and structured P2P-based solutions rely on their resource index which is updated periodically.

Agent-based techniques are not suitable for large scales due to using flooding mechanisms which increases the message loads dramatically. Also, using the limiting factors such as TTL causes the agent-based techniques to potential false positive errors.

Almost none of the approaches can be completely reliable because the approaches using flooding mechanisms potentially fall into bottleneck and false positive errors, and approaches following the informed searches are prone to caused by the single point of failure problems.

Table 1. Comparison of Different Resource Discovery Approaches

Efficiency Metric	Centralized Approach	P2P-based Approach	Agent-based Approach
Quality of the Results	Suitable only in small scales	Suitable for structured-P2P based	Suitable only in low dynamic conditions
Scalability	Not scalable due to high message loads on the central or upper level indexing nodes and high latency in large scales	Scalable in structured-P2Ps and super-peer, but not scalable in unstructured-P2P due to high message loads in large scales	Not scalable due to high message loads caused by the flooding agents in large scales
Dynamicity	Not supported due to potential bottleneck in the central servers or upper level indexing nodes in hierarchical structure caused by update messages referred to the central server or indexing nodes	Unstructured P2Ps are tolerant to dynamicity due to distributed load, but structured-P2Ps act poorly in dynamic conditions due to difficulties of indexing nodes' updates	Tolerant in dynamic conditions due to autonomy property of mobile agents
Reliability	Not reliable because of potential single point of failure and bottleneck in the central servers or upper level indexing nodes of hierarchical solution	Structured-P2Ps and super-peer are roughly reliable however sometimes suffer from single point of failure, unstructured-P2Ps are not reliable because of the false positive error	Not reliable because of the potential false positive error in large-scale environments
Maintenance Costs	Medium due to using limited and integrated indexing nodes. Medium for hierarchical solution	Low for unstructured-P2Ps, medium for super-peer. But high for structured-P2Ps due to keep the indexing nodes up-to-date	Medium due to autonomy of mobile agents

6. Conclusions

With regard to importance of resource availability of grid applications, the resource discovery is a critical service in such systems. But some inherent characteristics of Grid environments such as scalability, dynamicity and so on, makes the resource discovery a challenging task. Different approaches provided to index and discover the resources in grid environments are presented and compared in this paper. Taking into account the inherent tendency of grid environments to large scales and other factors discussed in the paper, the structured P2Ps and combinations of hierarchical and super-peer-based techniques can be considered as the best solutions to index and discover the resources in grid environments.

REFERENCES

- [1] A. H. D. Cokuslu, K. Erciyes, "Grid Resource Discovery Based on Centralized and Hierarchical Architectures," *International Journal for Infonomics*, vol. 3, pp. 283-292, 2010.
- [2] C. Mastroianni, et al., "Evaluating Resource Discovery Protocols for Hierarchical and Super-Peer Grid Information Systems," in *Parallel, Distributed and Network-Based Processing*, 2007. 15th EUROMICRO International Conference, pp. 147-154, 2007.
- [3] K. Vanthournout, et al., "A taxonomy for resource discovery," *Personal and Ubiquitous Computing*, vol. 9, pp. 81-89, 2005.
- [4] H. C. a. X. C. Yulan Yin, "The Grid Resource Discovery Method Based on Hierarchical Model," *Information Technology Journal*, vol. 6, pp. 1090-1094, 2007.
- [5] M. MollaMotallebi, et al., "Reliable Resource Discovery Approaches for Grid Environments," *International Journal of Computer Communications and Networks (IJCCN)*, vol. 1, pp. 42-47, 2011.
- [6] C. Mastroianni, et al., "Designing an information system for Grids: Comparing hierarchical, decentralized P2P and super-peer models," *Parallel Comput.*, vol. 34, pp. 593-611, 2008.
- [7] M. MollaMotallebi, et al., "A New Model for Resource Discovery in Grid Environment," *Informatics Engineering and Information Science*, pp. 72-81, 2011.
- [8] D. Tsoumakos and N. Roussopoulos, "Adaptive probabilistic search for peer-to-peer networks," in *Third International Conference on Peer-to-Peer Computing (P2P 2003)* Proceedings, pp. 102-109, 2003.
- [9] L. K. Daniel A. Menasc, "Probabilistic scalable P2P resource location services," *SIGMETRICS Perform. Eval. Rev.*, vol. 30, pp. 48-58, 2002.
- [10] D. T. P. Trunfio, P. Fragopoulou, C. Papadakis, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi, "Peer-to-Peer Models for Resource Discovery on Grids," *CoreGRID Technical Report TR-00282006*.
- [11] P. Trunfio, et al., "Peer-to-Peer resource discovery in Grids: Models and systems," *Future Gener. Comput. Syst.*, vol. 23, pp. 864-878, 2007.
- [12] R. Ranjan, et al., "Peer-to-peer-based resource discovery in global grids: a tutorial," *Communications Surveys & Tutorials*, IEEE, vol. 10, pp. 6-33, 2008.
- [13] A. Hameurlain, et al., "Resource discovery in grid systems: a survey," *Int. J. Metadata Semant. Ontologies*, vol. 5, pp. 251-263, 2010.
- [14] J. Yu, et al., "Grid market directory: A web services based grid service publication directory," *arXiv preprint cs/0302006*, 2003.
- [15] E. Elmroth and J. Tordsson, "An Interoperable, Standards-Based Grid Resource Broker and Job Submission Service," presented at the *Proceedings of the First International Conference on e-Science and Grid Computing*, 2005.
- [16] S. B. S. Andreozzi, L. Field, S. Fisher, B. K'onya, M. Mambelli, J. M. Schopf, M. Viljoen, and A. Wilson. (2005), *Glue schema specification version 1.2*. Available: https://www.opensciencegrid.org/twiki/bin/viewfile/Main/GIPAttributes/GLUEInfoModel_1_2_final.pdf
- [17] T. G. Ramos and A. C. M. A. d. Melo, "An Extensible Resource Discovery Mechanism for Grid Computing Environments," presented at the *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*, 2006.
- [18] D. a. J. S. Kaur, "Resource Discovery in Web-Services Based Grids," in *Proceedings of World Academy of Science, Engineering and Technology*, pp. 284-288, 2007.
- [19] E. Benson, et al., "Evaluation of UDDI as a provider of resource discovery services for OGSA-based grids," in *Parallel and Distributed Processing Symposium*, 2006. IPDPS 2006. 20th International, p. 9 pp, 2006.
- [20] G. Molt, et al., "A service-oriented WSRF-based architecture for metascheduling on computational Grids," *Future Gener. Comput. Syst.*, vol. 24, pp. 317-328, 2008.
- [21] I. Foster and C. Kesselman, "The Globus project: a status report," in *Heterogeneous Computing Workshop*, 1998. (HCW 98) Proceedings, pp. 4-18, 1998.
- [22] I. Foster, "Globus toolkit version 4: software for service-oriented systems," presented at the *Proceedings of the 2005 IFIP international conference on Network and Parallel Computing*, Beijing, China, 2005.
- [23] M. Yan, et al., "Resource Discovery Algorithm Based on Small-World Cluster in Hierarchical Grid Computing Environment," in *Grid and Cooperative Computing*, 2008. GCC '08. Seventh International Conference on, pp. 110-116, 2008.
- [24] S. Ratnasamy, et al., "A scalable content-addressable network," *SIGCOMM Comput. Commun. Rev.*, vol. 31, pp. 161-172, 2001.
- [25] A. Andrzejak and X. Zhichen, "Scalable, efficient range queries for grid information services," in *Peer-to-Peer Computing*, 2002. (P2P 2002). Proceedings. Second International Conference on, pp. 33-40, 2002.
- [26] I. Stoica, et al., "Chord: A scalable peer-to-peer lookup

- service for internet applications," SIGCOMM Comput. Commun. Rev., vol. 31, pp. 149-160, 2001.
- [27] M. Cai, et al., "MAAN: a multi-attribute addressable network for grid information services," in Grid Computing, 2003. Proceedings. Fourth International Workshop on, pp. 184-191, 2003.
- [28] A. Iamnitchi and I. Foster, "A peer-to-peer approach to resource location in Grid environments," in Grid resource management, N. Jarek, et al., Eds., ed: Kluwer Academic Publishers, pp. 413-429, 2004.
- [29] C. Mastroianni, et al., "A super-peer model for resource discovery services in large-scale grids," Future Gener. Comput. Syst., vol. 21, pp. 1235-1248, 2005.
- [30] M. Marzolla, et al., "Resource Discovery in a Dynamic Grid Environment," in Database and Expert Systems Applications, 2005. Proceedings. Sixteenth International Workshop on, pp. 356-360, 2005.
- [31] D. Talia, et al., "Peer-to-peer models for resource discovery in large-scale grids: a scalable architecture," presented at the Proceedings of the 7th international conference on High performance computing for computational science, Rio de Janeiro, Brazil, 2007.
- [32] I. Filali, et al., "A Simple Cache Based Mechanism for Peer to Peer Resource Discovery in Grid Environments," in Cluster Computing and the Grid, 2008. CCGRID '08. 8th IEEE International Symposium on, pp. 602-608, 2008.
- [33] M. Li and M. Qi, "Facilitating resource discovery in grid environments with peer-to-peer structured tuple spaces," Peer-to-Peer Networking and Applications, vol. 2, pp. 283-297, 2009.
- [34] R. Mokadem, Hameurlain, A., Tjoa, A.M., "Resource discovery service while minimizing maintenance overhead in hierarchical DHT systems," in iiWAS(2010), pp. 630-638, 2010.
- [35] F. Palmieri, "Percolation-Based Replica Discovery in Peer-to-Peer Grid Infrastructures Networks for Grid Applications," vol. 25, A. Doulamis, et al., Eds., ed: Springer Berlin Heidelberg, pp. 45-56, 2010.
- [36] S. Ding, et al., "A Heuristic Algorithm for Agent-Based Grid Resource Discovery," presented at the Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service, 2005.
- [37] Y. Jian, et al., "Grid Resource Management Based on Mobile Agent," in Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on, pp. 255-255, 2006.
- [38] J. Kang and K. Sim, "A multiagent brokering protocol for supporting Grid resource discovery," Applied Intelligence, vol. 37, pp. 527-542, 2012.