

A Survey of Resource Discovery Approaches in Distributed Computing Environment

B.Senthil Murugan

Assistant Professor

School of Information

Technology and Engineering

VIT University, Vellore

TamilNadu, INDIA

Daphne Lopez

Associate Professor

School of Information

Technology and Engineering

VIT University, Vellore

TamilNadu, INDIA

ABSTRACT

With the increasing need for networked applications and distributed resource sharing, there is a strong incentive for an open large-scale service infrastructure operating over multi-domain and multi-technology networks. Distributed computing such as peer-to-peer networks, promise a distributed computing infrastructure that can provide globally available network resources. The network size and complexity continue to increase dynamically. Huge number and heterogeneous nature of computing resources make the resource management a significantly challenging task. Discovering resources in a Grid environment is complex due to the heterogeneous nature and dynamic availability of resources and are owned by different individuals and organizations, each having their own resource management policies and different access and cost models. The motivation behind this paper is to explore the various resource discovery mechanisms that have been designed and implemented with a variety of algorithms and approaches.

Keywords

Resource Management, Resource Discovery, Gnutella, Agents.

1. INTRODUCTION

Resource management in the distributed computing environment is the process of managing available resources and system workloads accordingly. It includes resource discovery, resource monitoring, resource inventories, resource provisioning, fault isolation, autonomic capabilities and service level management activities. Out of these resource discovery is the first and the foremost important process. In peer-to-peer network, a resource discovery system to locate specific resources is required. Resources are bound to specific hosts, representing hardware devices, logical system objects, files or software entities that are managed by an application. The resource discovery is a basic functionality that enhances the accessibility of resources in the distributed context, i.e., given a user request, a resource discovery mechanism should locate and return a set of peers' addresses that match the description of requested resources. Resource Management System performs resource discovery to obtain information about the available resources for the particular job.

Resource Discovery process is critical for efficient resource allocation and management. For making the Resource Discovery more efficient and reliable large numbers of Approaches are there. This paper analyses some of existing Approaches for Resource Discovery, which can search for the preferred resources quickly and efficiently (return the correct results quickly and reduce network complexity). There have been many

projects that have designed and implemented the resource discovery with a variety of algorithms and Approaches. Though this paper does not introduce any new discovery approach, this could be useful for analyzing and using the existing approaches in an efficient manner.

2. SEVERAL APPROACHES

Resource Discovery is a systematic process of determining which grid resource is the best candidate to complete a job in shortest amount of time, with most efficient use of resources at minimum cost.

2.1 Group Manager Approach

The Group Manager component is proposed [1] to support the discovery of nodes with resources that are currently reachable. It supports two kinds of groups: managed group created and owned by a particular node and other nodes may join the group, which allows the creator node to access the group nodes, and peer groups, that do not have an owner or manager and there is no mechanism to join a peer group. Nodes that create groups with the same name will implicitly be members of the same peer group. Nodes attach and dynamically update the application specific data and also propagate their availability to the groups they create. The search algorithm implemented by the Group Manager is a peer-to-peer search wherein the search request is qualified by a group name, which limits the search to be processed only by other nodes that are also part of the specified group. The search term is forwarded by the Group Manager to neighboring Group Managers, which pass the term onto the applications on those nodes. If a peer application wishes to respond to a search request, the application provides a response term to its local Group Manager and is transmitted back to the original node that initiated the search. The hop count and flood probability are the parameters that are used to constrain the search operation. The information propagated and searched are opaque to the Group Manager.

2.2 Distributed Indexing

Distributed index component called DIndex is proposed in the paper [2] that can be used in peer-to-peer networks for resource discovery. An Indexer is a node in the peer-to-peer network which administrates a small portion of a search space managed by a distributed index which is referred to as a Sub-Index. All Sub-Indexes form together a Distributed Index. It uses a splitting algorithm to partition and equally distribute the registered resources among the existing and the newly created Indexer. When an ordinary peer searches for a certain resource, it sends a lookup message to its connected super-peer. Every super-peer is connected to one or more Indexers and manages a maximum

number of connected ordinary-peers. The super-peer selects one of the Indexers it knows that is the closest to the target Sub-Index that contains the desired resource. The super-peer has to know the Indexer-ID of the corresponding Indexer, and must be able to connect the Indexer-ID carrying the attribute in the lookup-request. This procedure offers the advantage that the lookup-request-message is forwarded towards the direction of the target Sub-Index. On one hand DIndex offers the lookup functionality alike a centralized database but with decentralized technology. On the other hand a peer-to-peer system using DIndex becomes more complex, and the implementation of the communication is more difficult.

2.3 An Efficient Resource Discovery Algorithm

The algorithm proposed in this paper [3] is a hybrid between the push and pull approaches. In Push approach, resources are pushed through the network so that they reach nodes that have requested the resources. In Pull approach, a node floods the network with a resource request. The basic idea behind this algorithm is that, when a moving node comes in contact with another node, they exchange a list of queries and resources that have high priority in the network and later they update their database as soon as it receives the list from another node. The algorithm works in four distinct phases. i) Neighbor detection – the node knows its neighbors and updated information about the requirements of the neighboring nodes. ii) Dynamic ranking of the queries – the nodes has queries which are not yet serviced and they need to be spread in the network. The queries are ranked using the ranking function and the queries' signatures. iii) Dynamic ranking of the resources – resources are ranked using queries that were ranked in the previous phase. iv) Dissemination – top-ranked queries and resources are transmitted to their direct neighbors. This algorithm optimally disseminates queries and resources in the network without spreading redundant data. Important queries and resources are given more preference discarding less important ones from a node's memory.

2.4 Mobile Resource Discovery Algorithm

A resource discovery algorithm is proposed [4] for mobile phone communication based on Mobile Ad-hoc Networks. This algorithm considers each node to be equivalent and can communicate with each other by single or multi-hop. Each node plays three key roles: broadcaster, listener and the maintainer. The two main stages in the algorithm are Discovery Stage and Maintenance Stage. In discovery stage, each node broadcast their existence through self advertising mechanism and then when a node receives the packet, the listener updates the route table. In maintenance stage, the maintainer sends maintenance packet for each entry in the route table to ensure whether the table is up to date. If the node does not receive a response message in a given time interval, the path is assumed to be broken. According to the resource requests, each node indexes the route table so as to communicate with other nodes. The algorithm is deployed on Visual Studio2008 as the Windows Mobile emulator manager simulation instead of real mobile phones that make the composition of mobile Ad hoc networks. The algorithm is effective as it uses distributed hash tables for storing the route table information and there is no central server or cluster so that each node plays a similar role.

2.5 Knowledge Driven Approach

A knowledge-driven model is proposed [5] that form network ontology. The proposed system consists of two layered architecture consisting of peers and super-peers. The peers are the owners of the resources and are responsible for assigning metadata to each resource. Network's concepts and relationships among them are maintained by the super-peers. Each super-peer is responsible for one or more concepts. The peers are connected to the super-peers that support the concepts they relate to. The network peers are not directly searched to find a resource; instead super-peers forming the ontology concepts are searched. When a client sends the query, the query is first submitted to the parent super-peer and then the query expansion process takes place, wherein the keywords of the query are compared against the concepts in the ontology and the most related concepts are extracted. A heuristic technique is used to find the next best super-peer for forwarding the request. In this way, network resources are semantically categorized and the queries are routed based on their semantic evaluation. The proposed system is evaluated using a case study for a music file sharing system. Experimental results have shown that the proposed system improves the searching process with high success rate. But, this idea of network adaptivity should be implemented more in detail in a dynamic environment for testing its advantages against the current network structures.

2.6 Web Services based Grid

Resource Discovery in web-services grid can be achieved by adding UDDI (Universal Description Discovery Integration) [6], which is an open standard for publishing and discovering the software components of service-oriented architecture. It has four entities: businessEntity – has the information about the publisher of the services, businessService – has the useful descriptive details about the service, bindingTemplate – gives the link between the service description and the actual endpoints where the services can be accessed, tModel - contain lists of key-value pairs used for description and associated with multiple objects in the UDDI database. The difficulty in handling dynamic information updates are overcome by having a rich Query model. Accessing a grid service is much like accessing a page on the internet, in that it requires a URL. Simple Object Access Protocol (SOAP) enables the transmission and reception of messages for accessing distributed resources or objects. Using SOAP over HTTP is ideal for grid services. The Grid Resource Discovery service is a set of many sub-services that can be glued to the SOAP web service for larger resource discovery problem. So when the request is sent, the discovery service provides us the grid resources we require. The Web Service describes the services and then a SOAP request is sent asking for the specific resource. The Grid Service Provides replies with a SOAP response which contains desired resources or an error message if the request was incorrect. However there are some challenges to be considered for resource discovery in the grid infrastructure.

2.7 Broker based approach

The proposed system [7] uses brokers, which serve as a resource representative and selects the best resource. Resource Brokers are used to maintain a repository of resource registrations. Local Resource Manager registers with one or more Resource Brokers so that its resources are represented in the grid. A set of requests are given to the Client Agent and for each request it submits a request to its Resource Brokers for a suitable resource. The

registration content and the resource requests are compared and if the matching succeeds the client is given back with the suitable resource. If the match is unsatisfied, it is forwarded to a selected number of neighboring resource brokers and the process is continued for a predefined value. If a set of resources are received for a request the client may use resource selection algorithm approach to find the suitable one. The performance is measured against random walk algorithm and it has been proved that the response time is greatly reduced in the proposed system. The proposed system has been evaluated in a Java simulator and needs to be implemented in a real grid environment.

2.8 Gnutella Protocol

The Gnutella protocol [8] is an open, decentralized group membership and search protocol, mainly used for file searching and sharing. The term Gnutella represents the entire group of computers which have Gnutella speaking applications loaded in them forming a virtual network. Each node can function as both client as well as server and hence called as servent. Thus they can issue queries to other nodes as well accept and respond to queries from other nodes, after matching the queries with the contents loaded in their own hard disks. Thus queries are not sent to any central server as in any client –server network, but are handled between the nodes. When a servent joins the network, it discovers the available nodes by sending ping queries to the existing servents, which may response with pong messages. When a file is needed to be searched, a Query is sent to all known servents. If the servent finds a match, it responds with Query hit message, otherwise it forwards the query via push message to all the servents that it knows of. Once the file is found, it is downloaded outside the Gnutella network using Hyper Text Transfer Protocol. Perhaps the biggest problem with Gnutella is that each servent uses so much bandwidth for sending and receiving Ping/Pong and Query/Hit messages that are not related to it. It is also believed that most users on the Gnutella network are not making any files available for sharing, thus making the network very inefficient.

2.9 Agent based Discovery

An agent is a software entity, which can migrate between physical nodes, can work for another entity, reacts to external changes via listeners and are able to control its internal working state to achieve the final goal, which gives an agent some artificial intelligence features. Even if the host machine, which launched the agent, is eliminated from the network, the agent can still work. Thus, the mobile agents are very powerful programs, which can act even in the absence of the machine that initiated them. After completion of their assigned tasks, the mobile agents return to the host machine to report the result or simply terminate. An agent hierarchy [9] is created for addressing the scalability problem of a node in the cloud platform. Whenever a new resource is added in the cloud, a new agent is created and the agent should register with one of the existing agent and hence join the agent hierarchy. An agent in the hierarchy can communicate and learn about other low-level agents. When a resource is no longer available in the cloud, the agent should leave the hierarchy, informing its low-level agents. A Service Model is built, where the resource information are maintained. The resource performance will vary dynamically and hence the information in the service model should be updated dynamically by the agents. This service model is advertised in the agent hierarchy. So, when a request arrives, the

agents must be able to transform the information into useful performance measures and hence able to make a decision about whether the corresponding resource can meet the request requirements. Otherwise, it can contact other agents until the service is found.

3. CONCLUSION

Resource Discovery is the process of finding the satisfactory resources according to the user's request, including resource description, resource organization, resource lookup and resource selection. Supporting multiple types of resources, high performance, and massive scalability are some of the most important goals in the design of a distributed resource discovery system. In this paper, various grid resource discovery approaches have been analyzed.

4. REFERENCES

- [1] Niranjan Suri, Matteo Rebeschini, Maggie Breedy, Marco Carvalho, Marco Arguedas: "Resource and service discovery in wireless ad-hoc networks with agile computing". Published in the proceeding of the 2006 IEEE conference on Military communications.
- [2] Marco Hentschel, Maozhen Li, Mahesh Ponraj and Man Qi: "Distributed Indexing for Resource Discovery in P2P Networks". Published in 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009.
- [3] Ravi Thanawala and Jie Wu, Avinash Srinivasan: "Efficient Resource Discovery in Mobile Ad Hoc Networks". Published in IEEE ICC, 2009.
- [4] Li Liu, Yu Liu: "A Resource Discovery Algorithm Based on Mobile Ad Hoc Network and Its Implementation". Appeared in 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 2010.
- [5] Athena Eftychiou, Bogdan Vrusias: "A Knowledge Driven Architecture for Efficient Resource Discovery in P2P Networks". International Conference on Intelligent Networking and Collaborative Systems, 2010.
- [6] Damandeep Kaur, Jyotsna Sengupta: "Resource Discovery in Web-Services based Grids". World Academy of Science, Engineering and Technology 31 2007.
- [7] N.Malarvizhi, Dr.V.Rhymend Uthariaraj: "A Broker based approach to Resource Discovery and Selection in Grid Environments". International conference on Computer and Electrical Engineering, 2008.
- [8] Gayatri Tribhuvan, University of Freiburg: "A Brief Introduction and Analysis of the Gnutella Protocol".
- [9] Shunli Ding, Jingbo Yuan, Jiubin Ju, Liang Hu, Jilin University: "A Heuristic Algorithm for Agent-based Grid Resource Discovery". Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service on e-Technology, e-Commerce and e-Service.
- [10] Anju Sharma, Seema Bawa, Thapar University, Patiala : "Comparative Analysis of Resource Discovery Approaches in Grid Computing". Journal of Computers, VOL 3, NO. 5, May 2008.