ISSN: 0974-5823 Vol. 5 No. 2 December, 2020

International Journal of Mechanical Engineering

Resource Discovery Attempt for Manufacturing Grid Environment

Avijit Bhowmic[1],
[1]SRMS College of Engineering and Technology, Uttar Pradesh, India

Arup Kumar Nandi[2],
[2]Principal Scientist, CSIR-CMERI, West Bengal, India

Goutam Sutradhar[3],
[3]Director, National Institute of Technology, Manipur, India.

Abstract
One of the significant issues of in distributed systems like Manufacturing Grid is how to effectively search among so many different resources to get the one that effectively matches the requirements. The primary issue presented by a lot of distributed systems is the removing the possibility that there may be a communication bottleneck or a single failure point. When faced with such circumstances, standard methods may not be instantly appropriate because of the restricted inquiries and varied properties of the resources. a completely decentralized resource discovery service constructed on top of an undefined layer and that only utilizes data on resource attributes and characteristics is an alternate strategy. One of the most difficult aspects of providing such a service is locating necessary and acceptable resources without having global knowledge of scattered sharing resources. This is one of the service's greatest problems. As a direct consequence of this, the amount of network overhead rises in direct with proportion to the quantity of nodes that are used by the resource discovery service. Inside the scope of this research, we proposed a decentralized resource discovery service that utilizes a direction-aware method. If implemented, this approach has the potential to lower network traffic within unstructured information systems. According to the results of the experiments, the method that was proposed results in a higher rate of success at a cheaper cost and with more scalability.

Keywords: Grid, manufacturing Grid, Resource, Distributed systems.

Introduction

Manufacturing Grid is a kind of distributed systems where resources are geographically scattered collaborating together for a common objective. By encapsulating and integrating all of the manufacturing resources deployed across heterogeneous systems and geographies into multiple matching resource services, MGrid enables users to access a wide range of manufacturing services in a transparent manner. As a result, users may utilize all MGrid resources as easily as they would local ones. In fact, MGrid represents the realization of resource sharing. Manufacturing resources are quite different from those computing resources or data resources, including software like Computer Aided Design (CAD), Computer Aided Process Planning (CAPP), and Computer Aided Manufacturing (CAM), as well as different kinds of machine tools like Computerized Numerical Control (CNC), Rapid Prototype Manufacturing (RPM), etc.

Now, in this work we consider the nodes are CAD computing nodes which are connected together and form a set of networks. So, one of the most basic challenges presented by such distributed systems is the search for and integration of the necessary resources to carry out the allotted tasks and services. For instance, in order to provide a scalable runtime environment, grid or cloud service providers task schedulers to locate low-cost resources that can be used for the provisioning of cloud services. This is done so that they may accept batched executions and other similar tasks [1]. The information system is crucial for aggregating huge-scale, heterogeneous information of resources in grids and clouds [2,3]. The main objective of the information sites is to index and distribute resource information among various administrative nodes in order to provide a directory service. [4,5]. A centralized or hierarchical structure is often used by conventional information systems in order to facilitate the tracking and searching of directories of allocated resources [6,7]. The management of information pertaining to global resources is made more simpler by these solutions. As there are more resources available, the efficiency of the system will decline as a connection bottleneck or one point of failure is more likely to arise. This is due to the fact that there will be one less resource that can handle the increased demand [8,9]. The underlying or physical network design is concealed by the P2P layer, an abstract network. This network has peers as well as logical links. In this sort of P2P layer, as network nodes, peers serve and communicate by one another via connections that are established between any associated nodes. As a consequence of this, in a decentralized system, any peer may use a routing mechanism in order to broadcast or search information about the sharing of material or resources. In respect of scalability, the system's performance and durability may be improved by the use of P2P techniques for discovering resource in grids and clouds. [10-13]. P2P techniques are traditional excellent in finding exact queries when they make use of identity information. It is more difficult to conduct searches in the context of grids since the many resource features that are supplied for resource identification include a wide array of characteristics [14,15]. These qualities include things like the machine speed and memory capacity. In addition, the qualities of a resource may shift in their status over the course of time, and the optimal value of a desirable resource feature may lie anywhere within a fuzzy range. As a result, the provision of an efficient method for resource identification that can enable difficult enquiries in the large-scale environment of distributed systems is of the utmost importance. Many cutting-edge solutions for addressing distributed systems queries with several attributes settings have been presented as a result of the research that has been conducted on this subject. Unfortunately, the connectivity of network design and the indexing information of resource locations are subject to stringent limits due to the presence of these structured systems. As a consequence of this, the performance of the system may be negatively impacted when there is a high rate of churn and a broad variety of requests are being requested for information [16,17].

Previous research used a formless P2P technique in distributed environment like Grid to find the dedicated several resources, taking into consideration the importance of both simplicity and high levels of resilience. They also presented some concepts that may be used to the development of a general resource finding solution. In a system like this, the information on the resources is not broadcast to a structured system; rather, it is saved locally. As a result, the system does not ask for any extra effort to be done in order to either maintain a closely connected topology or dynamically republish the resource information. If you choose a system that is not organized, on the other hand, there is a possibility that the problem of the inquiry will be mindlessly transmitted to successive neighbors. In addition, in contrast to data files, the computational resources that are part of distributed computing systems cannot be copied at will [18,19]. For the purpose of providing a completely decentralized resource discovery service, this research makes use of a cutting-edge approach to answer resource queries over an unstructured overlay. Finding a single sought resource without any knowledge of global resources provides the greatest obstacle; as a consequence, the research process for desired resources strongly depends on the resources found in neighboring areas. Our purpose is to respond to these inquiries in a way that is more effective and entails less administrative work within the framework of widespread resource sharing. The technique may give direction awareness depending on the features of the resources to make it easier to find the required resources. Experiments are utilized to evaluate how successful the proposed strategy is by using a number of different factors and an uneven distribution of information about the resources. The findings indicate that recommended tactics exceed beyond the traditional approaches by more than 30 percentage points, and in some instances, by as much as 40 percentage points [20].

There are three primary contributions to this discussion. To begin, a one-of-a-kind approach for distributed resource information along with discovery system utilizes simply one unstructured basis layer in conjunction with a great number of attribute overlays [21 23]. In contrast, the conventional strategy makes use of a number of organized base layers. This study also provides a strategy for direction-aware resource searching that takes into consideration the properties of the resources being searched for in order to cut down on unnecessary network overhead. The solution that was provided may eventually answer the challenging issue with a greater success rate at a lower cost than an unstructured network. This is accomplished without the use of any worldwide information [24-26].

we have shown the decentralized resource information and discovery system as well as the processes that are related with it and we address important works and connected research.

2. Related Work

Manufacturing Grid is completely heterogeneous distributed systems. So, this approach would be ideal for MGrid as well. For an information system to be effective in mitigating the negative impacts of hierarchical or centralized organizational structures, it must take a decentralized approach to problem solving. Therefore, discovery of distributed resource is an essential duty for these types of systems in order for them to live up to the expectations of their users.

The confluence of P2P systems, as well as a variety of decentralized approaches to peer-to-peer resource information finding and indexing, have attracted a lot of attention lately. Grid systems and P2P have also been the object of in-depth research. Nonetheless, while looking for resources, one may often come up against three roadblocks when working within the framework of grid and cloud computing. The first difficulty is that there is a diverse selection of components, and the issues themselves are difficult to understand. In addition, there are things known as static resource characteristics in addition to things known as dynamic resource properties, both of which may have their values changed at any moment. In addition, the subject matter of a request might be an arbitrary value range for the resources that are required. This is directly leading to the development of novel approaches to enable decentralized discovery of resources for distributed computing systems.

This study also takes use of the fact that the structured overlay exists. For instance, MAAN, Mercury, SWORD, and Squid are some of the systems that are discussed in this sort of research. In these types of systems, it is the job of each node, which is also known as a peer, to update and maintain a directory index that integrates part of the resource data given by other nodes in the system. These solutions allow for a high degree of control to be exercised on the network's design as well as the positioning of information pertaining to the resources that have been made available. Because of this, updating in resource status is required if there is a change in the value of dynamic attribute, and structured information service needs to be republished with the most recent version of the resource's state. It's possible that the overall cost of updating a system will go higher as a result of the large amount of information that has to be updated. As a direct result of this, the costs associated with maintaining structural processes are quite significant.

The system which is unstructured lack a thorough grasp of the network's architecture or the locations of its resources, they are unable to make efficient use of either of these aspects of the network. Because of this, each node either randomly or naively bombards its neighbors with requests, which results in a significant amount of unnecessary overhead for the network. This makes it simpler to search unstructured P2P networks for the data that is being sought, as shown by the results of a number of different pieces of study. Traditional P2P techniques, on the other hand, do not take into account the peculiarities of the resources; as a result, it is impossible to directly use these strategies within the framework of an information system that makes use of computational Grid.

In their first study, A. Iamnitchi and I. Foster concentrate on request processing as well as different request propagation mechanisms to identify the needed resources while designing a completely decentralized strategy to address the issue of resource locating in Grid systems. At the same time, a strategy to solve the issue of resource finding in Grid systems is being designed, and this is being done as part of that process. During the process of formulating a solution to the challenge of identifying resources in Grid systems, this step is taken. On a resource finding solution that is more general in nature, four architectural axes have been taken into consideration. The four axes that comprise this system are the pre-processing, request processing, the design of the overlay, and the membership protocol. An additional piece of study has revealed that a genuinely decentralized resource discovery service might be able to shorten the time it takes to find new resources as well as the total number of resources used. The concept of transmitting an inquiry, whether or not the additional information that was lying on nodes was the primary focus of the majority of the experiments that were conducted in the past. When it is possible to make use of the additional information, a request is propagated using the histogram data. If it is not viable to make use of the additional information, the request is given to an ignorant node that has been arbitrarily selected from among its neighbors. However, neither the attribute type nor value of the resource characteristics are taken into consideration in any manner by these forwarding approaches.

Depending on the required resource attributes and the values associated with them, each query in a Grid system may be handled differently. This enables each node in the system to pass the request to a neighbor that provides higher advantages. The objective of this study is to examine a completely decentralized and flat resource locating service that is built on an unstructured overlay. Each node will then just keep records of its own resources after this is complete, rather than broadcasting this information to other nodes or the network that is rigorously regulated. As a consequence of this, there are no additional costs connected with the process of either preserving the painstakingly organized topology or republishing the data to the appropriate nodes. Both of these processes are performed flawlessly.

However, if we utilize pure flooding in an unstructured network, it is probable that we will add more nodes that are not necessary for detection. This is because unstructured networks have more nodes than structured networks do. Because of this, a sizeable portion of the routing messages are made useless, which, in turn, causes the system's overall performance to decline as the network's nodes increase in number. An approach known as a random walk search is one of the options that is being examined for use in the effort to cut down on the total amount of messages and provide some relief from the problem of flooding. In this particular search, each node will only transmit the request to one of its neighbors at each successive phase. As a direct result of this, the request that is being sent across the overlay network will be arbitrarily sent from one node to the next as it moves along. The visitation coverage range may need to be lowered in order to decrease the overall volume of messages that must be sent. On the other hand, the system that utilizes a selection method that is entirely based on chance wanders about aimlessly and in no particular direction while going to each of the nodes.

It is possible to cut down on the overall cost of messages in distributed systems, which in turn brings the administrative burden of the network down. As a direct result of this fact, the primary focus of this body of work is on making a proposal for a resource finding approach that takes into account the features of the resource. The intention behind this study is to improve the performance of conventional techniques.

3. System for resource finding and information

Dispersed distributed systems like MGrids have an increasing number of distributed resources. Scalable and reliable architectural designs are used for the purpose of resource management. In order to avoid the problems that are associated with hierarchical or centralized methods, decentralized strategies should be used in information systems. Distributed resource discovery is thus an essential feature for such systems, since it is necessary to satisfy the expectations of users. In recent years, there has been a great deal of attention. Several decentralized methods for peer-to-peer indexing and the discovery of resource information exist at the junction of Grid and P2P technologies. However, when it comes to the resources of Grid, resource discovery often runs into two issues. First, there is a large amount of variety in the resources, and searches are complicated. Second, the values of the qualities of resources may remain the same throughout time or they can be dynamic, meaning they can change at any moment. In addition to that, a request might specify an arbitrary value range for the resources that are being requested. Innovative approaches for the generation of decentralized resource discovery in distributed computing systems are being developed as a direct result of this.

In these types of systems, it is the responsibility of each node, also known as a peer, to keep a directory index that includes a portion of the data from the resources held by other nodes. These approaches exert a stringent level of control on the topology of the network as well as information on the locations of the published resources over. As a result of this, the status of the resource has to be changed If a dynamic attribute's value changes and the structured information service receives an update needs to be re-published with the resource's most current state. The amount of information that has to be updated will, in turn, drive up the total cost of a system. As a direct consequence of this, the upkeep of the structure technique is somewhat pricey.

To link the peers who are taking part in the system, some systems with laxer controls produce an unstructured network, while other systems, in contrast to the system with stricter controls, produce the structured network. The topology of the entire network is not known to these unstructured systems. As a consequence of this, each node either floods or distributes requests to its neighbors in an arbitrary manner, which results in substantial overhead for the network. Numerous research has proven that carrying out actions such as these makes it easier to search unstructured P2P networks for the data that is required. Conventional peer-to-peer (P2P) techniques, on the other hand, do not take into consideration the features of the computing resources; as a result, these approaches are inapplicable to a grid-based information system's configuration. In light of this, this article presents a novel method for discovering the perfect matched resource in the environment of distributed systems.

A. Iamnitchi et.al have started looking at the design of a technique that is entirely decentralized as a first step toward finding a solution to the challenge of determining where resources should be put in Grid systems. In order to determine the resources that are required, they concentrate on the processing of requests and the many alternative tactics for the propagation of requests. Four architectural axes have been taken into consideration for a general solution for discovering the resource: the membership protocol, layer design, pre-processing, and processing concerned request. A follow-up investigation showed that the effectiveness of a completely decentralized resource discovery service significantly increased in terms of the quantity of resources it consumed and the time it required to complete its responsibilities. The most significant thing that can be learned from previous research is that a request may be sent with or without the supplementary information that is kept on nodes. When the extra information is being utilized, the request is dispersed using the histogram data; otherwise, it is delivered to an ignorant node that has been randomly picked among its neighbors. On the other hand, these forwarding approaches do not take into account either the resource characteristic's value or attribute type.

Each query in a system that makes use of a Grid has the possibility of receiving a response in accordance with the prerequisites of resource characteristics and the values that are connected to them. Because of this, each node is able to forward the request on to a neighbor that has a better chance of fulfilling it. An unstructured overlay-based flat resource finding service is the subject of investigation in this body of work. The provision of the service is wholly decentralized. Because of this, the only thing that each node is able to do is record information about its own resources, and it is prohibited from spreading of information to either the strongly regulated system or to different nodes. Because of this, the expenses associated with the maintenance of the meticulously organized topology and the republishing of the data to the appropriate nodes are not increased in any way.

However, if we employ flooding in an unstructured network, it is probable that we will include additional nodes that are not necessary for the discovery process. This may happen if you utilize pure flooding. Because of this, a significant portion of the routing messages are deemed meaningless, and the general performance of the system will suffer as the network's nodes increase in number. The use of a random stroll search as a way to reduce the total size of messages is something that is currently being considered as a solution to the flooding issue. At this stage of the process, each node will only send the request to one of its neighbors, as opposed to sending it to all of them. Because of this, the request will be distributed around the overlay network in a manner that is completely random, moving from node to node. It's likely that decreasing the average age of visitors might assist reduce the quantity of messages that need to be routed, so keep that in mind. On the other hand, the approach that chooses nodes completely at random does so without following any particular guidelines or having any specific goals in mind.

The conception of a resource discovery method that considers resource features is the main topic of this work. The ultimate goal of this research is to enhance the effectiveness of conventional techniques. This is attainable as a result of the fact that the total message cost in distributed computing systems is able to be minimized while concurrently decreasing the network overhead.

3.1 System Overview

This study takes into consideration a distributed system, which refers to a network configuration consisting of several information systems, such as a grid or the cloud. Each information system inside such a system is responsible for organizing its own resources into a certain region and keeping a directory of resources that are positioned in close proximity to one another. The environment for the distributed sharing of resources is produced by a node in a loosely connected network acting as an information system. This means that the information system is responsible for establishing this environment. Every node in a network that has been thus fully decentralized is in the dark regarding the resources that are pooled and dispersed among the network's other nodes on a global scale. This is because every node in such a network is completely independent from the others. In the preceding example, for instance, every node in a distributed system is only aware of the neighbors to whom it is directly related. This is because every neighbor has a direct connection to every other neighbor. When it is necessary, a node will make contact with other nodes in order to fulfil the objective of flat and decentralized resource discovery. This will allow the aim to be realized. The node finds the necessary resources that meet the parameters' requirements more easily as a result.

Figure 1 shows a graphical representation of a logical system architectural model for a network of nodes connected with matching neighboring nodes which gives this portrayal. The structure consists of a single basic overlay in addition to a number of attribute overlays. An overlay which those functions as the support system for the system forms the basis of the overlay base. This overlay also serves as the foundation of the base overlay. The basic overlay is layered with many additional attribute overlays, each of which represents a different form of resource data. Every node contributes to the creation of an attribute overlay by doing so, taking into consideration the surrounding area of the unstructured overlay and the kind of resource information it has. To be more exact, each attribute overlay is composed of the same nodes that are a node's neighbours; however, various attribute overlays could have different request forwarding paths.

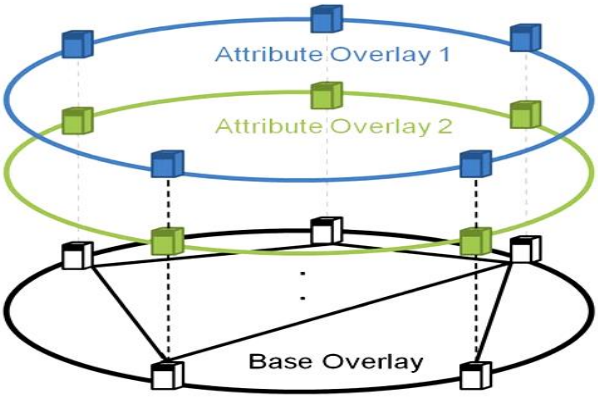


Fig. 1: System design seen from a logical perspective

3.1.1 Membership

In order for the process of starting up to commence, it is necessary for each node to first establish its membership in the fundamental overlay. An additional node is first added to the unstructured overlay in order to facilitate the establishment of connections to other nodes that are already in place. A node that is a joining node will connect directly to a group of K other nodes that are already established. These nodes will be chosen by another node that will be referred to as a bootstrapping node. This will allow the joining node to get its initial data. A new node has the chance to join the system for resource sharing and successfully take part in the unstructured overlay if it first establishes connections to the already-existing nodes and then participates in the unstructured overlay. After that, the new node will begin the process of establishing an adjacent group of neighbors and will start communicating with them about the state of resources that are nearby.

Even though each attribute status may have a distinct attribute value, all of the nodes in this system have the same category of resource attributes. This is true despite the fact that the attribute statuses may vary. To put it another way, many providers of resources may be willing to selflessly share their devoted resources with other people. Despite this, the skills and capacities of the various sharing resources may vary from one another. Sharing information with neighbors is a procedure that does not result in any extra overhead being added to the traffic on the network. This is because the process of updating information may be "piggybacked" on the operation of keeping the overlay maintained. In addition to this, the process of exchanging data with nearby neighbors does not call for any more bandwidth to be made available. As a consequence of this, the node that matches the query does not have to republish the information to the carefully restricted indexing service; rather, all it has to do is update the status of the information on the shared local resources that it has with its neighbors.

3.1.2 Neighbor Set

Using the neighbors' knowledge concerning shared resources, each node executes an approximate calculation of the ranges for each and every attribute between its local resources and those of its neighbors related to them. We are capable of transforming any resource's attribute value into a related sequence. Considering the comparative' outcomes every node distinguishes its neighbors into two different types of sets denoted by (L) and denoted by (S).

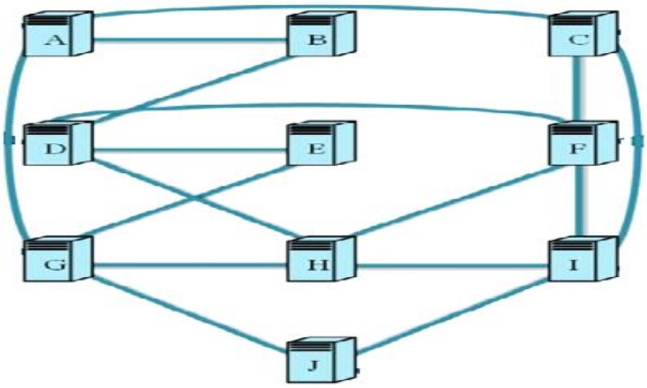


Fig. 2: Using an unstructured overlie network as an illustration of a decentralized information system

Let fa be a function which fetches the number of a certain feature from the resource characteristics of a node or from query criteria. The kind of collection of neighbors of node ni is named NEi. As of the result, ni can divide the NEi into La and Sa for each attribute a, such that

$$La = \{nl \mid nl \in NEi, fa(nl) \geq fa(ni)\}$$
$$Sa = \{ns \mid ns \in NEi, fa(ns) < fa(ni)\}. \quad \text{.....................................} (1)$$

If the obtained parameter is greater or equal to the local data, the neighbor is then designated as a large-neighbor; else small-neighbor set. Significantly, for the large-neighbor set, if ny is a neighbor of nx in Sa and ny is a neighbor of nz in NEy, Consequently, fa(nz) must be less than fa (nx), and vice versa. This is as a result of the contributing nodes in our system being configured using an individualized network architecture rather than a structured layer.

Figure 2 is a representation of an unstructured information system, which will serve as the basis for our demonstration of how our method operates later on. The quantity of work that is being done by the computing node and the amount of memory that is available are two examples of features that may be found in resources. Assume that a value for the system load attribute is assigned to each node in the range nA through nJ, and that this value changes between 10 and 100 percent. This value is used to describe the amount of work being done by the system. Memory capacity attribute values may vary anywhere from 1 GB all the way up to 10 gigabytes and are given to every node in the tree, starting with A and going all the way through J. Each node communicates resource information with its neighbors, classifying those neighbors according to the membership management and splitting them into the categories of big and little neighbors' groups respectively. As an example, the quality of the resource known as memory capacity as an illustration. The nF node's L memory is where nC and nD are kept, whereas its memory is where nH and nI are kept.

3.2 Discovery in Resource through Direction –aware Strategy

An investigation into the information system which is unstructured is being passed out in order to put a discovery which is reliable resource service into operation, which will allow for the transmission of network enquiries. In an information system like this one, it is standard procedure for each node to first search through the local data in order to discover the necessary resource. This is done in order to get the procedure forward more quickly. If there isn't a resource that is nearby that meets the requirements, the node is supposed to send concerned request to different other nodes of the environment that are nearby and ask them to search through their local indexing data for the resource that is needed. If there isn't a resource that is nearby that meets the requirements, the node will transmit the request. When a request of this sort is received by a neighbor, that neighbor will verify the availability of its local resource, and if the resource in question is accessible, it will send a response to the node that requested the resource. If, on the other hand, the resource in issue is not accessible, the request will be sent to further neighbors, and it will continue to be passed around until either the essential supply is located or a stop situation is met.

In this regard, one of the most essential issues is finding out how to identify productive nodes so that the search may go further. When it comes to unstructured P2P networks, two tried-and-true routing algorithms are often utilized to identify in the network the necessary resources. The deluge is one kind of scenario, while the random stroll is another. In order to acquire the required resources, the flooding-based strategy makes contact with the neighbors of each node. The random-walk-based system is less effective than the flooding-based approach. Because of this, there is a probability that a bigger quantity of network overhead will be experienced the extra inappropriate nodes are included in routing system. This is because of the increased likelihood that the network will need to do more processing.

3.2.1 Explanation of the Mechanism

The resources in the system being looked into differ from one another in a variety of ways. These attributes comprise, but are not limited to, the capacity that is available and the level of system load. In situations like these, including the sharing of resources, resource discovery has to respond to a set of queries necessary pairings of trait with significance combinations. In order to offer a response to the sophisticated query and identify the appropriate resource that fulfils the requirements, the attribute-based routing strategy has been chosen as the way to utilize. This decision was made after careful consideration.

Throughout the rest of this article, we will refer to the collection of attributes provided in the demand of a query as AQ. The letters Q and AI stand for a collection of attributes that are included in the local resource information I; the letters AOa stand for an attribute overlay that is based on an unstructured network. Local resource information I is represented by the letter I. In order to find the resource that is appropriate, a multi-attribute query is made to one AOi. The major attribute that users have specified may be the attribute denoted by the letter "I," or it could be an attribute that was picked at random from the set of available attributes. Because each node has access to all of the information that pertains to its neighboring resources, the enquiry may be resolute by any joined feature over lie. This is because of the fact that each node has access to all of the information.

In order to cut down on costs associated with maintaining a network, the number of nodes that need to be connected should be reduced. Because every node in our system keeps a record of the geographical resource information that is relevant to it, it is feasible for a node in our system to make effective use of the request and resource information. It is hypothesized that unstructured information systems direction-aware resource discovery might be carried out when using this kind of knowledge about resources.
This comparison causes the two values to be compared. The approach known as direction-aware routing (DAR) is predicated on the overall premise that may be summarised as follows:

$$\begin{cases} La, \text{ if } fa(AQ) \geq fa(AI) \\ Sa, \text{ if } fa(AQ) < fa(AI) \end{cases} \qquad \text{...(2)}$$

The proposed DAR approach might be capable of responding to each's range inquiry resource property if the value is provided as an abstract range. This is due to the fact that abstract ranges are far more challenging to query. As illustrated in Equation, the mechanism that is responsible for resource allocation has the ability to select whether or not to issue a plain range query that employs just one kind of relational operator. If it does so, the query will be as follows: (2). On the other hand, the conceptual series may be outfitted using both a logical operator and a relational operator, which would make it feasible to execute complicated range searches. This would be conceivable if the abstract range was supplied with both of these operators. One example of this would be a need for the amount of memory capacity to be between 4 and 8 gigabytes (GB).

Simply dividing the complex inquiry into its component parts will make it much easier to obtain an answer to the difficult issue posed regarding the range. There are two sub-queries: the first one is to find the resource with memory (AI) that is more than 4 gigabytes, and the second one is to find the resource with memory that is less than 8 gigabytes. These two subsidiary queries will now be incorporated into a single primary query. After obtaining each set of sub-results queries, the outcomes are combined, and intersection is used to the combined set-in order to acquire the entire list of resources which matches sub-query. Despite this, it is abundantly evident that there has been a rise in the amount of traffic that is being carried on the network. The method that has been given provides the capability of passing all of any attribute overlay's range queries neighbors as an alternative choice. Because of this, the resource discovering procedure may decide whether to send a message or not. Because each node has the geographic required data for its own local indexing resource, this is achievable. In light of this, the resource discovery process may decide whether to send a message depending on its relevance.

$$\begin{cases} La, \text{ if } fa(\text{Min\_}AQ) \geq fa(AI) \\ Sa, \text{ if } fa(\text{Max\_}AQ) < fa(AI), \end{cases}$$

..........................(3)

where MaximumAQ and MinimumAQ denote, respectively, the upper and lower bounds of the proposed attribute values' abstract ranges. The abstract range of the intended attribute values may vary anywhere from 0 to 100.

If a local indexing resource's attribute value upper limit is lower than its important resource maximum bound, a request will be issued to a member of the set of those little neighbors. If the requested resource's upper threshold is more than the locally indexed resource's upper bound, a large-neighbor set node approves the request. Each and every query has capability to be sent to the nodes that come after it, in accordance with the overarching notion that the DAR method employs in the decision-making process. In the following sections, we will explain the process by referring to two traditional models, and then we will present some potential tactics by making use of those images.

3.2.2 Flooding Employing Direction-aware Approach

The standard form of the flooding approach involves sending a query to the nodes that are next to each system node. This is the strategy that is used in general. The need for a growth in the quantity of messages will cause an increase in the amount of network traffic, which will in turn cause an increase in the quantity of nodes which are used to construct the scheme. To mitigate the potentially detrimental effects of this limitation, the purpose of our research is to identify dedicated resources using a reduced number of meaningless nodes as communication points.

We are now putting out a novel method that is being called direction-aware routing with flooding in the hope that it would convince individuals to alter their perspectives (DAR-FLOOD).

Because each node is aware of the characteristics of the resources it controls, it is possible for a node to decide whether or not to transmit data by comparing the attribute values of itself and its neighbors. This is made possible by the fact that each node is aware of the characteristics of the resources it controls. If the value of the attribute of a desired source has a lower value, then the node will contact the v to send the request if their bandwidth is less than that of the local resource. If the value of the sought-after resource's attribute is higher, the node will get in touch with the other nodes in its large neighbor-set. It is possible for a node to lack any of the additional neighbor sets if it is subject to the exceptional case of an exception. The neighbors of a node in an unstructured network are chosen at random and are not limited in any way, therefore the value of a particular node might be the highest or the lowest of all of its neighbors. This is due to the fact that the unstructured network does not include any constraints. This is due to the fact that the network is not organized in any way. When faced with a situation like this, the node has the choice of beginning communication with the neighbor set that is hostile to it.

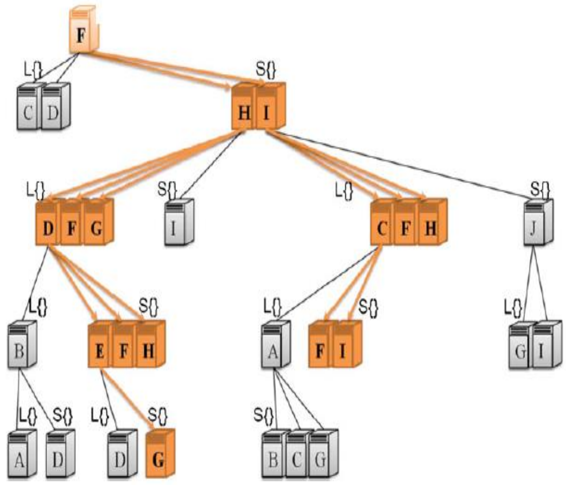


Fig.3: Visualization of resource identification using direction-aware flooding as an example

A straightforward example of the use of the resource that can be directed seeking approach in context of flooding is shown in Figure 3. We have presented the information using a tree structure so that it is easier to comprehend. The decision on whether or not to flood is made based on a comparison between the values of attribute of the nF resource's local availability and the necessary resource's capacity. This comparison is required as the source that is available in nF does not meet the requirements to fulfil such a request. In this specific situation, the attribute of the resources that is available locally is more than the value of the resource that is required. The required resource, however, has a lower value. This is because sending the request to all of its neighbors would result in an excessive amount of traffic in the network. Each node performs an evaluation of the request in accordance with the concept that was presented earlier, and it prevents the request from being sent until it has either completed the task on its own or concluded that a dedicated resource is available, as is shown here at nG. As a result of this, the resource discovery method that incorporates direction awareness is in a position to direct the request to the group of nodes that are more productive. This is in contrast to flooding the request to all nodes without discrimination, which could result in the inclusion of information that is not relevant to the search at hand.

3.2.3 Combining random walk with direction- aware strategy

In contrast to a system that just relies on flooding, a random walk system routes requests by using a walker that travels to just one randomly picked node at each step. This is in contrast to a system that solely depends on flooding. Because of this, there is a discernible drop in the total volume of traffic on the network. Nevertheless, if the walker visits a greater number of pointless nodes, hence increasing the quantity of meaningless messages, then using the random walk strategy can result in the routing route taking a longer amount of time. In order to alleviate the inefficiencies that have been produced by this circumstance, it is planned that each walker will be directed to the resources that they need. In order for us to achieve our objective, we devised a routing strategy that included both direction-awareness and random walking (DAR-RW).

Comparing the attribute value findings was the foundation for making choices about forwarding in the DAR-RW approach, just as it was in the DAR-FLOOD strategy. This specific node of the environment will be chosen from the set of small neighbors of the node that is currently being visited. In the event that this is not the case, the walker will choose at random a node from the set of big neighbors of the node that is presently being visited. Because this is a one-of-a-kind scenario, the node that is now being visited could be missing out on seeing one of the neighbor sets. As a consequence of this, the walker would have a tough time navigating the path in the desired direction in order to reach the resource that was being sought. The walker in this scenario has the option of continuing to examine other prospective neighbors by selecting a node at random If the top limit of the requested resource is greater than the upper bound of the locally indexed data, out of the set of a large-neighbor a node approves the request what would allow the walker to continue their search for further potential neighbors.

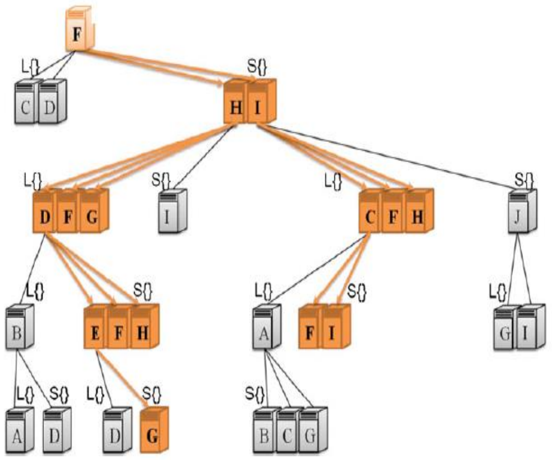


Fig.4: Resource exploration display using a random walk that takes orientation into account

The strategy that has been suggested may be seen in Figure 4, which also provides an illustration of the strategy. A node must decide whether to transmit the request based on a comparison of the request's resource characteristics with the local resource because nF's local resource and the request are inconsistent. This comparison must be done in order to satisfy the fact that the local resource and request the do not match. Here, the amount of nF resources that are being sought is more than the quantity of resources that are necessary; yet, the number of resources that are required is smaller. Only the set of small neighbors that consists of nH and nI is chosen at random when nF passes on the request to the next node. This is the only set of tiny neighbors that exists. The set of all neighbors, which consists of nC, nD, nH, and nI, is disregarded in this analysis for some reason. According to this idea, each node will continue to send requests to the next node that will benefit them until either the resource that is being sought is discovered or the stop condition is fulfilled. Request forwarding, as opposed to aimless roaming, may be directed to nodes that are more relevant based on the proposed random walk method for direction-aware resource finding. This is possible because of the random walk. This stands in stark contrast to the common practice of wandering about aimlessly.

3.3 Maintenance overhead

To be able to take part in the unstructured layer, it is necessary for each node in the system that we propose to establish communication with a specific quantity of additional nodes. Each node determines which other nodes in the network are its neighbors by checking the list, using the resource information, and comparing the attribute values. Each and every node provides a more accurate classification of its neighbors by assigning them to either the L set or the S set. Our solution makes an attempt to give a road map for re-directing the request solely based on the features of the resource that is in issue, and this is one of the ways in which it accomplishes this goal. Each and every node has the capability of searching for another node inside the network that either possesses the greatest or lowest value, depending on the sets of L and S, and vice versa. To put it another way, there is at the very least one route that can be taken from each node in the unstructured network to the node that either has the greatest or lowest value. This is true whether the network is viewed as a tree or as a graph. Inference: If there is a resource out there, it will most likely be found at some time in the future. Each node has the capacity to direct a query to the destination, which has a better chance of producing a matched query at a lower cost, as contrasted to a totally decentralized system for resource information and discovery. In order to strengthen the system, the suggested approach should also consider retaining membership and keeping data up to date. These are the two things that need to be considered. A single base overlay and a large number of attribute overlays make up the architecture of the system as a whole. As a consequence of this, the only thing that is left for us to do is to check that the essential underlying is being properly cared for. Due of the ease of use of an unstructured network and the lack of control it gives over the placement of information, it is useful to use a network which is unstructured to establish the basic overlie.

This is because network which is simple. As a direct and immediate result of this, the fundamental tasks involved in the maintenance of the overlay have a considerable influence on the system's resilience to failure.

Any new node that is part of the unstructured overlay may simply link or go away the overlie by establishing communication with bootstrapping node. Within the framework of the unstructured overlay, this is carried out. If the dynamics of the nodes follow the standard pattern, then it will be able for each node to then update its membership with its neighbors. It is possible for neighbors to be kept updated about the status of a node via the use of a keep-alive message that is frequently sent in order to detect whether or not the node has failed. As a direct result of this, as a component of the overlay's maintenance, each node may, in addition to maintaining its membership, update its neighbor set-in accordance with the piggybacked resource information from neighboring nodes. This information comes from neighboring nodes.

4. Evaluation of Performance

In this section of the piece, an analysis of the effectiveness of the various tactics that were suggested is presented. Following the presentation of the methodology for assessment that will be used as well as the major metrics that will be of importance to our study, we will go over the experimental results found from the experiments that were conducted using the suggested technique.

4.1 Evaluation Procedures

We created a simulator so that we could analyses how successful the suggested method would be within the context of a decentralized Grid. This allowed us to determine whether or not the concept was worth pursuing. This adds to the unstructured overlay. In order to take part in the overlay, this step has to be taken. In Ganglia, there is a category referred to as the resource type, and we use this category to provide the resource characteristics to each individual node. Although the values that are assigned to each resource property are different for each node, the categories that are used to classify those values do not change.

In order to identify the necessary resource, the resource discovery module is designed to implement a variety of forwarding methods. These strategies will be used to search for the resource. The most significant aspect of the demand is mirrored in the inquiry, which, at the beginning of each iteration, queries about the needed characteristics of the resource while simultaneously delivering a value picked at random for each of those aspects. This is the most essential aspect of the demand. When there have been a sufficient number of cycles, the results of the performance are averaged. In the process of carrying out experiments, a broad variety of situations are broken down and recreated for the purpose of examination. Initially, the query is transmitted to the resource discovery service, accompanied with values that are chosen at random. In the second part of this test, we investigate how effectively the query works when there is a large imbalance in the manner in which the information about the resources is disseminated. The efficiency of both uniform and non-uniform distributions must be assessed in order to determine the required resource. To ascertain which distribution strategy is more effective, an evaluation must be conducted. We also analyze how changing the time-to-live (TTL), the neighbors which affect the performance.

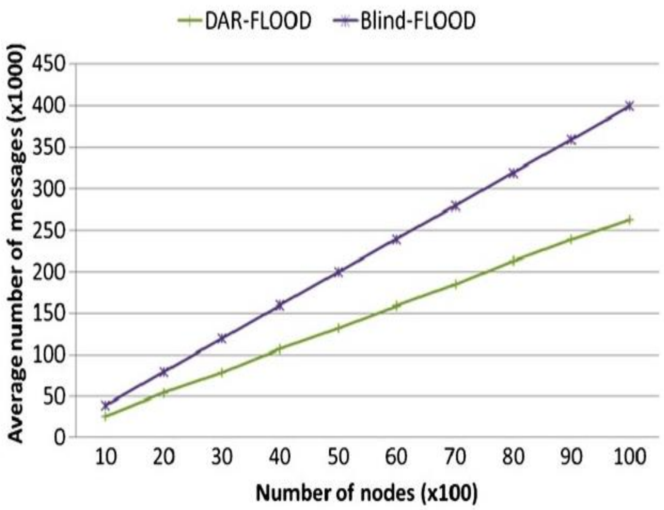
The success rate, the count of hops, and the cost are the three performance variables which is related with cost es that receive the most attention in the experiment in order to present a realistic picture of performance. As was anticipated, the resource discovery method will cover a greater number of nodes, which will, in turn, resulting in a rise in the number of messages of routing and a lengthening of the routing path. This rise in the nodes number that will be covered by resource discovery method was anticipated. The success rate, on the other hand, is the percentage of instances that the query was successful in locating at least one of the required resources. One way to interpret this is as a percentage that indicates how many times the query was successful. A greater success rate is indicative of a more efficient search technique, provided that there is no fixed condition that causes the search for resources to terminate.

4.2 Experimentation Outcomes

The outcomes of a number of tests are shown and discussed in relation to a directionally aware resource discovery service that is suggested for use by Grid information systems. This example illustrates the Blind-FLOOD approach, which includes flooding the query blindly to all of the neighbors at each step. Specifically, the example describes how to achieve this.

4.2.1 Cost Associated with Routing

To begin, the efficiency of networks of varied sizes is evaluated by looking at the normal number of sent and received messages(N). In this experiment, each node has a fixed number of 20 participating neighbors connected to it at any one time. According to Figures 5 and 6, the routing system creates an increasing quantity of messages, and this rise is directly correlated with the system's node count. Because the approaches that are based on flooding generate a greater number of messages than the methods that are based on random walks, it is possible to employ the flooding-based strategies in order to discover the required resource in a more straightforward manner. In contrast, the traditional blind flooding approach and the blind random-walk strategy are defeated by the direction-aware routing algorithms that are advised by an average of 33.48 percent and 43.58 percent, respectively, when compared to their respective performance. This outcome is because the technique that was proposed only takes into account the nodes that have the potential to deliver the most value rewards. As a consequence of this, we got the result that we wanted. In particular, the methods that are advised bring the average cost curve to a higher level by gradually climbing it as the network's node count increases to a certain level.



.

Fig. 5: Comparison no. of messages Vs ParticipatingNodes

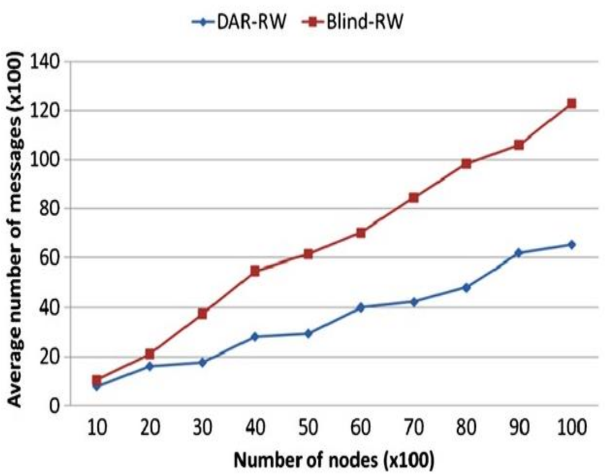


Fig. 6: Mean message count Vs node count-Randomwalk

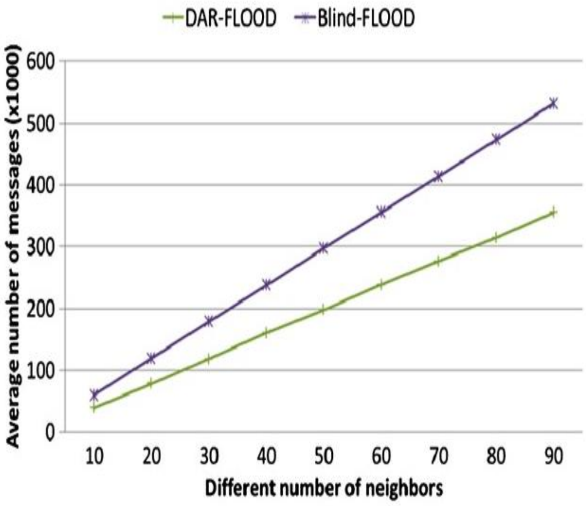


Fig. 7: Variations in the local population's impact on flooding

After that, the efficiency of the routing cost is analyzed by employing a range of population densities in the locations that are nearby. In this experiment, the size of the network is held constant at 3000 nodes, and we measure the average cost of delivering messages to a variety of distinct neighbors (K). Each of the scenarios shown in figures 7 and 8 is a potential occurrence that may have taken place. As the water draws nearer, there will be a greater need to accommodate a bigger number of neighbors, which means there will be an increase in the amount of discourse that takes place. Both of the routing algorithms produce a greater number of messages since the number of neighbors in the network rises, as seen in Figure 7. This trend is visible across both routing methods. On the other hand, the strategy that is advised works better than blind flooding by a margin of more than 30 percentage points and grows up gradually. This occurs as a result of the methodology only sending the request to a predetermined group of neighbors who are more helpful and who are aware of their surroundings.

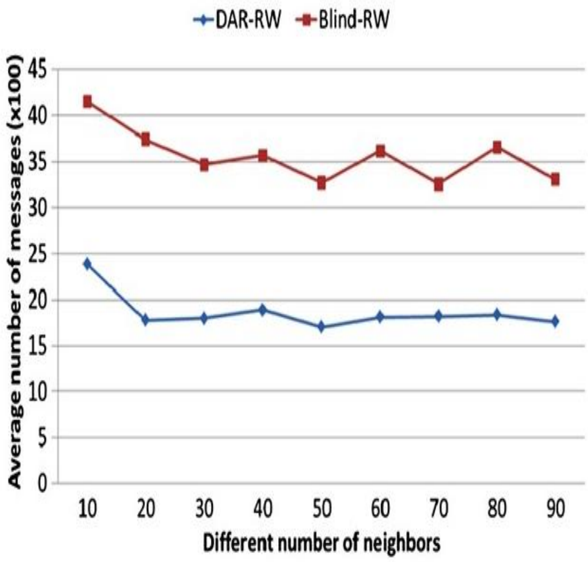


Fig. 8: Different amounts of neighbors have different effects on how easy the random walk is to do (N = 3000)

When random walk techniques are used, the system performance is not significantly impacted by the quantity of neighbors because the walker simply chooses a neighboring random node with sending the concerned request. suggests that there is no connection between the number of neighbors and the system's effectiveness. The idea is shown in Figure 8, which also shows that none of the random-walk-based strategies significantly changes the average message cost depending on the number of neighbors. This is shown to be the case for both of the approaches in the demonstration. The performance of our direction-aware routing technique is around fifty percent higher than that of the traditional routing strategy, and it is more scalable. Additionally, the performance of the conventional routing strategy is much lower.

4.2.2 Routing hops

For your convenience, the routing hops performance measurement is provided for you in Figures 9 and 10. The findings of the studies reveal that the strategy that is stands on flooding is improved than the technique based on random wandering when it comes to the discovery of new resources. The flooding strategy was used in the trials. When compared to the strategy that relies on aimless wandering, the approach to resource discovery that makes use of an awareness of where one is also results in improved performance.

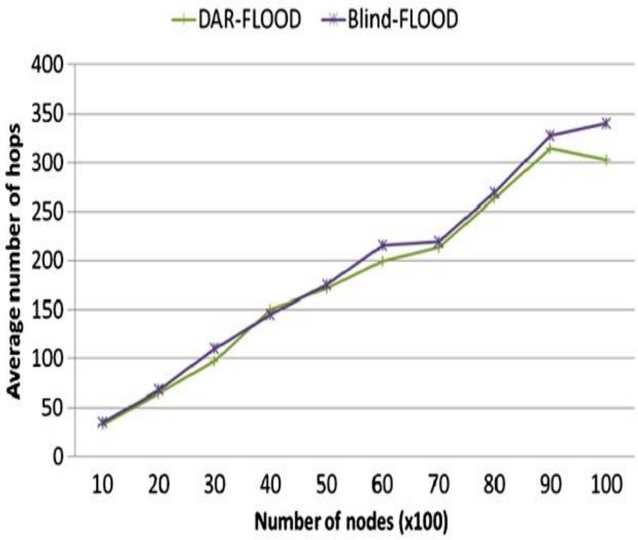


Fig. 9: Number of average hops divided by the total nodes number in a flooding graph

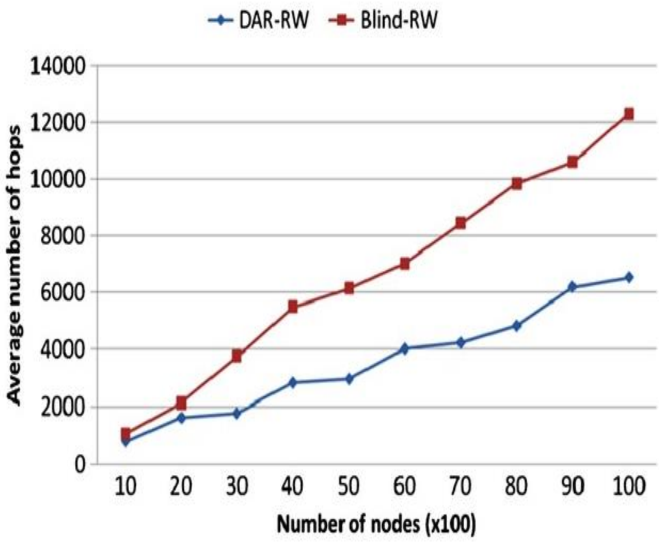


Fig. 10: Comparison of the Random Walk's Mean No. of Hops to the Total Nodes er(K = 20)

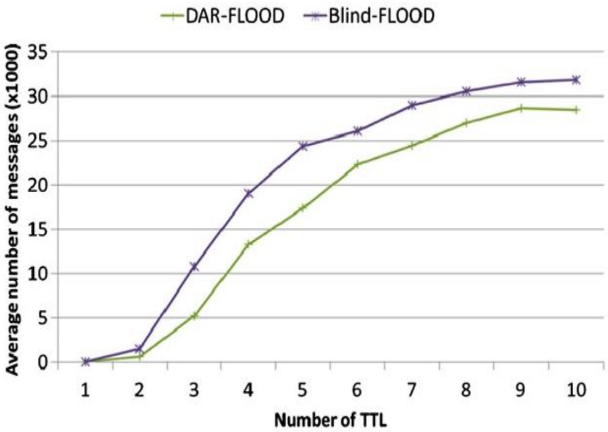


Fig. 11: The Flooding Algorithm's Typical Message Volume vs. Time-To-Live

The effectiveness of the flooding-based scheme we advocate using has an efficiency that is comparable to that of the traditional approach. Our recommended strategy for direction-aware routing uses flooding. This is done to guarantee that the flooding-based techniques can successfully find the target source as well as identify path of routing that is as small as is humanly practical. This is done in order to ensure that the flooding-based methods can successfully locate the target resource. The DAR-FLOOD approach is the only one that, according to the method that we have suggested, is capable of locating the essential resource by inspecting a portion of the neighborhood's homes. The Blind-FLOOD strategy, on the other hand, will take into consideration both large and small neighbor groups. Because of this, the DAR-FLOOD routing scheme's shortest viable routing route is also potentially discoverable using the Blind-FLOOD one. As a direct result of this, the usual distance covered by any of these two approaches in order to identify the same resource is almost same. However, in order to achieve the same degree of efficiency as the earlier experiments, the strategy that is based only on floods would incur far higher expenses. On the other side, the solution that was advised has the potential to deliver higher levels of productivity at lower levels of expense.

4.2.3 Non-uniform Distribution

To evaluate the effectiveness of the suggested technique, an extremely skewed distribution of resource information is employed in this experiment. The goal of this endeavor is to determine whether or not the strategy is effective. When it is essential to simulate a situation that has a distribution of property values that is not uniform, the Zipf (0.95) distribution is used as an approximation. The nodes number in the network is specified as 10,000, in addition to the number of nodes that are immediately next to each node is set to 20. In this experiment, a variety of different performance metrics are assessed using a range of TTL values. The results of this evaluation are presented. At the beginning of every round, the query is distributed to find the required supply by using a value that is chosen at random. During the phase known as resource discovery, the request forwarding process will be terminated if the resource that was being looked for can be found or if the TTL period expires.

Figures 11 and 12 illustrate the cost of routing, and they show that a rise in the TTL brings about a rise in the typical amount of messages. This is seen by the correlation between the two figures. The outcomes of our research show that the effectiveness of our proposed strategies is, respectively, 26.6 percent and 22.9 percent better than the blind flooding performance technique. This information was gleaned from the results of our research.

When both strategies are used with the same stop condition, The experiment's findings demonstrate that discovering the crucial resource using a flooding-based strategy generates more messages than doing so using a random-walking method. This is the case even though both strategies use the same amount of information. Even when the TTL is the same for both kinds of strategies, flooding-based strategies are capable to finish route with fewer hops than a random stroll strategy can. This is true even when flooding-based strategies are compared to random-walk strategies. The performance statistics in particular, which are given in Figure 13, demonstrate that the usefulness of the direction-aware is a suggested flooding scheme is related to effectiveness of the blind flooding technique with regard to the routing hops. This can be seen by comparing the two figures. To attain a high degree of effectiveness, the traditional flooding method incurs a bigger cost in terms of its routing. To reach the appropriate level of efficacy, this expense is required. On the other hand, the method we have suggested has the potential to be extremely effective at a cost that is acceptable.

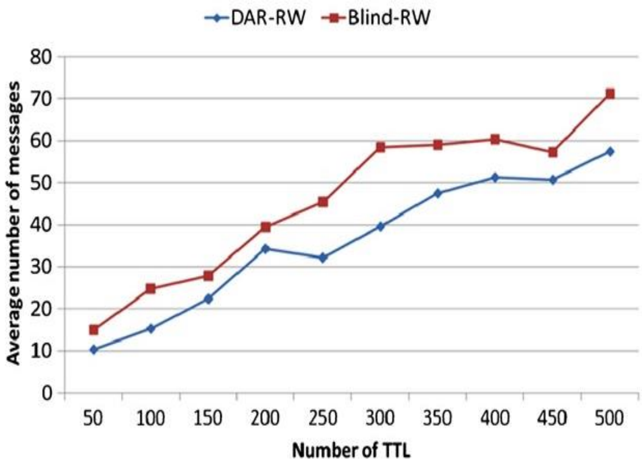


Fig. 12: The random walk method applied to the Average no of messages Vs time to live

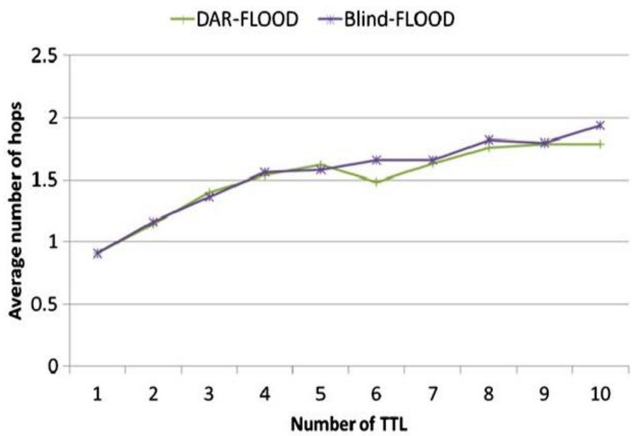


Fig. 13: Examining the flooding algorithm's average hop count against TTL

Alternatively, the experiment also evaluates presentation based on the proportion of successful efforts, as can be seen in Figures 14 and 15. According to the findings of the studies, the methodology that depends on flooding may achieve the same level of success with a lower TTL, but the method that relies on random walking requires a bigger TTL in order to achieve the same level of success. As we can see in Figure 14, there is tough possibility that the sought-after resource will be found by using one of the two strategies that are based on flooding. To put it another way, past experiments have shown that our direction-aware approach is capable of discovering the resource with a high success rate and little effort necessary on the part of the user. Figure 15 also reveals that a longer TTL is necessary for any of the random-walk-based methods in order to identify the resource of interest. This is something that can be deduced from the graph. The duration of the TTL and the success rate are intrinsically linked. In addition, in compared to the traditional random walk, the approach that we have discussed has the potential to obtain a better success rate while concurrently achieving a lower TTL. This is something that can be done simultaneously.

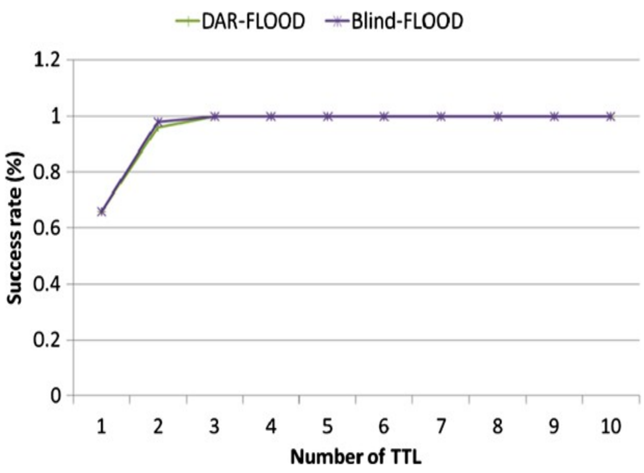


Fig. 14: Comparison of Flooding Success Rate vs. Time To Live

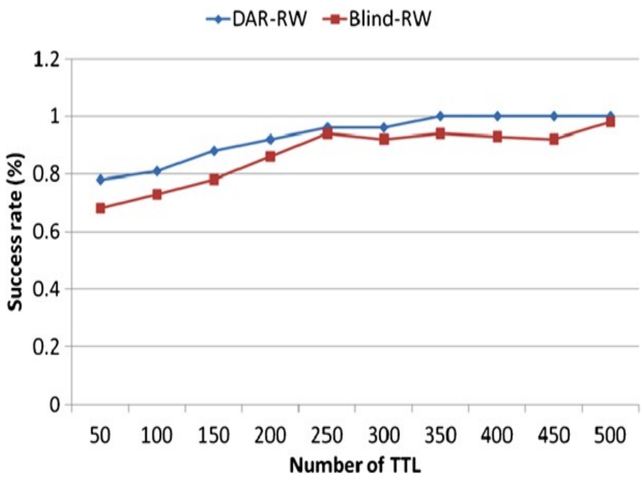


Fig. 15: successful rate vs time to failure - Random Walk

5. Conclusions and future work

One of the most important of these is the capability to index information about scattered resources and to determine which resources have been assigned. Large-scale distributed systems rely on the provision of these essential services. For the aims of this investigation, a system for complete resource information and discovering of resource that is flat and unstructured was used. Within the scope of this study, direction-aware algorithms are presented as a way of responding to resource queries made across unstructured information systems in order to give an effective resource finding service. An effective resource finding service is what this work aims to provide. The strategy that was outlined not only lessens the number of superfluous interactions, but it also accelerates the process of locating the necessary resource, does so at a low cost, and does so in a way that is highly scalable. All of these benefits come as a direct result of the strategy's implementation. The effectiveness of the system's performance may be measured in a variety of different ways, and this evaluation also takes into consideration the non-uniform distribution of information about resources. According to the findings of the studies, the method that was suggested performs much better than the approach of blind flooding as well as the strategy of random-walk by a margin of more than 30 percent and 40 percent, respectively. The development of a comprehensive system that takes into account resource information monitoring as well as service provisioning is going to need to be the primary focus of more research in the years to come. The management of the constant updating of the dynamic information presents an interesting and exciting task. It is crucial to find a balance between the expenses spent by the network in order to maintain track of the most current resource status and the accuracy of the information in order to avoid any unnecessary disruptions. High information accuracy may be attained while simultaneously minimizing the cost of message updating based on the distributed resource sharing system's retrieving technique. This is because the system has the potential to achieve both of these goals simultaneously.

Exploiting resource features to help other well-informed methods find the important resource in a distributed environment requires more considerable study. This is because the method relies on the information provided by the resource itself. This is due to the fact that the environment itself is spread out.

On the Cloud platform cloud manufacturing service that may be developed and our approach has the potential to be improved in order to deliver a resource information and discovery service that is both effective and efficient.

References

1. Chervenak A, Foster I, Kesselman C, Salisbury C, Tuecke S (2000) The data grid: towards an architecture for the distributed management and analysis of large scientific datasets. J NetwComputAppl 23(3):187–200.

2. Foster I, Kesselman C, Tuecke S (2001) The anatomy of the grid: enabling scalable virtual organizations. Int J High Perform ComputAppl15(3):200–222.

3. Marinos A, Briscoe G (2009) Community cloud computing. In: Proceedings of the 1st international conference on cloud computing, Beijing, China, pp 472–484.

4. Dikaiakos MD, Katsaros D, Mehra P, Pallis G, Vakali A (2009) Cloud computing: distributed Internet computing for IT and scientific research. IEEE Internet Comput 13(5):10–13.

5. Fitzgerald S, Foster I, Kesselman C, Laszewski G, Smith W, Tuecke S (1997) A directory service for configuring high-performance distributed computations. In: Proceedings of the 6th IEEE international symposium on high performance distributed computing, Portland, OR. IEEE Press, New York, pp 365–375.

6. Czajkowski K, Fitzgerald S, Foster I, Kesselman C (2001) Grid information services for distributed resource sharing. In: Proceedings of the 10th IEEE international symposium on high performance distributed computing, San Francisco, CA, USA. IEEE Press, New York, pp 181–194.

7. Massie ML, Chun BN, Culler DE (2004) The ganglia distributed monitoring system: design, implementation, and experience. Parallel Comput 30(7):817–840.

8. Zhang X, Freschl JL, Schopf JM (2003) A performance study of monitoring and information services for distributed systems. In: Proceedings of the 12th IEEE international symposium on high performance distributed computing, Washington, DC, USA, pp 270–281.

9. Mastroianni C, Talia D, Verta O (2008) Designing an information system for grids: comparing hierarchical, decentralized P2P and super-peer models. Parallel Comput 34(10):593–611.

10. Bharambe AR, Agrawal M, Seshan S (2004) Mercury: supporting scalable multi-attribute range queries. In: Proceedings of the 2004 conference on applications, technologies, architectures, and protocols for computer communications, Portland, Oregon, USA. ACM Press, New York, pp 353–366.

11. Cai M, Frank M, Chen J, Szekely P (2004) MAAN: a multi-attribute addressable network for grid information services. J Grid Comput 2(1):3–14.

12. Oppenheimer D, Albrecht J, Patterson D, Vahdat A (2004) Scalable wide-area resource discovery. Technical Report UCB/CSD-04-1334, EECS UoC, Berkeley.

13. Schmidt C, Parashar M (2008) Squid: enabling search in DHT-based systems. J Parallel Distrib Com- put 68(7):962–975.

14. Gummadi KP, Dunn RJ, Saroiu S, Gribble SD, Levy HM, Zahorjan J (2003) Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In: Proceedings of the 9th ACM symposium on operating systems principles, Bolton Landing, NY, USA. ACM Press, New York, pp 314–329.

15. Rhea S, Geels D, Roscoe T, Kubiatowicz J (2004) Handling churn in a DHT. In: Proceedings of the annual conference on USENIX annual technical conference, Boston, MA. USENIX Association, Berkeley, pp 127–140.

16. Iamnitchi A, Foster I (2001) On fully decentralized resource discovery in grid environments. In: Proceedings of the second international workshop on grid computing, Denver, CO, pp 51–62.
17. Iamnitchi A, Foster I (2004) A peer-to-peer approach to resource location in grid environments. In: Nabrzyski J, Schopf JM, Weglarz J (eds) Grid resource management. Kluwer Academic, Dordrecht, pp 413–429.
18. Foster I, Iamnitchi A (2003) On death, taxes, and the convergence of peer-to-peer and grid computing. In: The 2nd international workshop on peer-to-peer systems, Berkeley, CA, pp 118–128.
19. Talia D, Trunfio P (2003) Toward a synergy between P2P and grids. IEEE Internet Comput 7(4):95– 96.
20. Chung W-C, Hsu C-J, Lin Y-H, Lai K-C, Chung Y-C (2010) G2G: a meta-grid framework for the convergence of P2P and grids. Int J Grid High Perform Comput 2(3):1–16.
21. Trunfio P, Talia D, Papadakis H, Fragopoulou P, Mordacchini M, Pennanen M, Popov K, Vlassov V, Haridi S (2007) Peer-to-peer resource discovery in grids: models and systems. Future GenerComputSyst 23(7):864–878.
22. Ranjan R, Harwood A, Buyya R (2008) Peer-to-peer-based resource discovery in global grids: a tutorial. IEEE CommunSurv Tutor 10(2):6–33.
23. Kalogeraki V, Gunopulos D, Zeinalipour-Yazti D (2002) A local search mechanism for peer-to-peer networks. In: Proceedings of the 11th international conference on information and knowledge management, McLean, Virginia, USA. ACM Press, New York, pp 300–307.
24. Yang B, Garcia-Molina H (2002) Improving search in peer-to-peer networks. In: Proceedings of the 22nd international conference distributed computing systems, Vienna, Austria, pp 5–14.
25. Filali I, Huet F (2010) Dynamic TTL-based search in unstructured peer-to-peer networks. In: Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing, Melbourne, VIC, Australia, pp 438–447.
26. Tangpongprasit S, Katagiri T, Kise K, Honda H, Yuba T (2005) A time-to-live based reservation algorithm on fully decentralized resource discovery in grid computing. Parallel Comput 31(6):529– 543.
27. Lv Q, Cao P, Cohen E, Li K, Shenker S (2002) Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16th international conference on supercomputing, New York, USA. ACM Press, New York, pp 84–95.
28. Chung W-C, Chang R-S (2009) A new mechanism for resource monitoring in grid computing. Future Generation Computer System 25(1):1–7.