Journal of Theoretical and Applied Information Technology
31st October 2020. Vol.98. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# TOWARD ACHIEVING SELF-RESOURCE DISCOVERY IN DISTRIBUTED SYSTEMS BASED ON DISTRIBUTED QUADTREE

**[1]SALEH ALI ALOMARI, [2]MUHYEEDDIN ALQARALEH, [3]EMRAN ALJARRAH, [4]MOWAFAQ SALEM ALZBOON**

*[1,2]Department of Science and Information Technology , Jadara University, Irbid, 21110 Code, Jordan.*

E-mail: [1]omari08@jadara.edu.jo ,[2] m.qaralleh@jadara.edu.jo , [3]emran@jadara.edu.jo and [4]malzboon@jadara.edu.jo

**ABSTRACT**

Shared computing infrastructures such as Peer-to-Peer (P2P) networks and grid technologies allow sharing resources between a large amount of geographically and dynamically distributed resources to achieve an efficient degree of the performance of supercomputing. The main issue of this performance refers to the resource discovery as it is considered an essential and significant part in the infrastructure of shared computing. Several approaches pertaining to the resource discovery are suggested to perform resource discovery objective. These approaches cause an extensive response time, higher query cost and exceptional network traffic where the reasons behind these causes refer back to the dynamic construction and the distribution call of these approaches. Additionally, very rare approaches could tackle the problems of query hitting rate, memory utilisation and locality. These issues are critical to performance of a system. Therefore, an in-network querying organisation method, namely, the Distributed Quadtree (DQT) is proposed in this paper. This method produces efficient in-network resources of values nod's, including the distance conscious querying. The answering price of a query that is related to the value of definite resources at their maximum value is a static factor of the distance "d" to the close resources within the network. The construction of the DQT is local and does not include any existing announcements. Further, according to its stateless setting and orderly organisation, the proposed method is considered flexible within a failure intelligence.

**Keywords:** *Shared Computing, Resource Discovery, Quad-Tree.*

## 1. INTRODUCTION

Internet World Wide Web (WWW) was the applicant organizing improvement of the "Defense Advanced Research Project Agency (DARPA)" [1]. At first, it was utilized just by a few military areas in the United States to trade data [2]. In a brief timeframe, the undertaking turned out to be so well known and was used by numerous to frame a system of PCs. This development of "WWW" was accomplished through the instructive and research foundations which further extended to different spaces [3], [4]. Next, the Internet has ended up being a fundamental piece of the human race where a larger part of people can't envision their existence without the Internet.

In the present decade, Internet assumes a fundamental part as a noteworthy data thruway which can be utilized as a part of perusing and trading data, distributing web-based business and different sorts of mixed media, for example, "video conferencing", sound communication, and so on [5]. The Internet has turned out to be one of the individuals from the ware list in the figuring scene [6]. By and by, there are a billion PC gadgets that effectively associated with the Internet. The greater part of these PCs is not being utilized by any program or doing any calculation except if perhaps it is running a "screensaver." In any case, a few analysts [7] assert that the usage proportion for PC gadgets isn't surpassing 40% amid a run of the mill work day top paying little heed to occasions and the ends of the week. In this situation, we are expending assets consistently, with respect to "central processing unit (CPU) cycles," "hard disk (HD) space," "Random Access Memory (RAM)," and "network bandwidth (NBW)." Therefore, new research has all around pulled in the consideration of individuals for the use of sit still PCs that are associated with the Internet[8], [9]. This region of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

research is called "grid systems". "Grid systems" can be grouped into three noteworthy parts: "computational grid or Grid Computing" (GC), "service grid and data grid." Likewise, the GC can be parceled into three principle sorts: "distributed super-computing," "high throughput computing," and "Peer to Peer (P2P) computing" which are also referred to it as a "Shared Computing Infrastructures (SCI)"[10]–[15].

The "Chord protocol" proposes a method that is based on a resource discovery for O (log N) messages that are required to perform in O (log2 N) messages and in an inquiry handing out in order to update the routing information subsequently for a node that is being connected to a network. A number of messages for processing a query in the proposed method of this paper depends on the number of the timeline benchmarks. Hence, in order to establish an upgrade for the worst case through Chord, less than O (log N) of the benchmark must exist. Similarly, the quantity of updated messages that is required every time when a parent is provided to the timeline is based on the number of parents that is available within the benchmarks. The reason behind this is that updates should propagate to the newest benchmark. In order to represent the improvement of Chord, a number of parents among the benchmarks must not be greater than the messages, which are provided by an update in Chords [14], [16]–[20]. Fuzzy Classification Based Chord method implements a fuzzy classification by categorising different elements through a fuzzy set. Additionally, its functional membership is identified by a true value that relates to a fuzzy propositional function.

The Chord is mainly constructed within 2m nodes. The ring is distributed into three rings based on an indispensable Fuzzy-rule. Nodes that include a greater than 66% of resources are all gathered in a "HOTTEST RING". Nodes that also include resources ranging between 34% to 65% are all gathered in a "HOTTER RING". On the other hand, nodes that include less than 34% of resources are all gathered in a "HOT RING". This method aims at attempting to create a model that includes the whole needed conditions in order to differentiate between each ring efficiently. Once any new node arrives, the algorithm predicts the corresponding ring in an effective manner and the node joins the targeted place where the idea of the Divide-and-Conquer algorithm is implemented in order to disseminate the nodes through separated subsets. Hence, this process is recursively conducted for a subset of available nodes [21], [22]. Fuzzy numbers represent fuzzy subsets, which are based on the real line. They numbers contain a peak or plateau of a membership with Grade 1 in which the entire universe members are involved in the set. The membership function is rapidly increasing towards the peak and is being reduce apart from it [23].

The distributed quadtree overlay DQTO preserves a minimalist structure and makes this structure become stateless . The reason behind that refers to the use of an indexing technique, which is based on plotting a Quadtree into an in-operation field. When using the coordinates of the location information (X, Y) at a given node, the indexed maps distribute the computing environment node along to the equivalent level of a single rectanglur index. Level one rectangle is considered a minimum division reign within the DQTO [14], [24]–[27]. The index of the neighbouring parents and parent through each stage of a particular node is mathematically and efficiently calculated according to the index of the given node.

The basic idea is that creating the DQTO is performed within its location where it does not need any communication to be involved with it. When using the position of the information (X, Y) coordinates, this position is laid apart from a costly bottom-up building. When including the DQTO, parents are selected in every level in order to be the closest to the edge (borders) of the hierarchy instead of having these parents the closest to the core of the rectangle pertaining to that position. When selecting parents according to this technique, it is ensured that there are no backward links operations are advertised and discovered. Further, it is ensured that the network failure is avoided. A quad-tree mechanism is covered by the DQTO so that computing environments could be shared and suits a range of sensitive in-network that is revealed over an efficient storage of information. Various types of queries are supported by the DQTO, such as exact match query, partial match query, Boolean queries and range queries.

The remainder of the paper is organised as follows. Section 2 introduces the DQTO building algorithm for resource discovery in distributing computing systems. The DQTO routing algorithm is discussed in Section 3. Indexing the value of a resource is highlighted in Section 4. The validations of the results are analysed in Section 5. The conclusions are drawn in Section 6.

## 2. THE DQTO CONSTRUCTION ALGORITHM

Journal of Theoretical and Applied Information Technology
31st October 2020. Vol.98. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

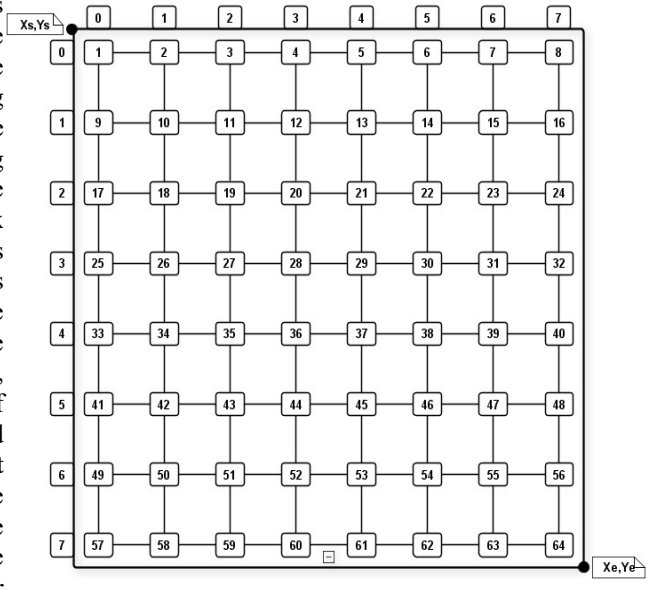
An easy The idea of utilising the Quadtree for in-network discovering when distributing computing environments is closer to the idea of the Quadtree through different centralised algorithmic domains [14], [24]–[27], particularly, through the geometric computation part. For example, the Quadtree aims at searching for one pixel within a two-dimensional picture. The picture that is hooked on recursive four quadrants is divided by the Quadtree such that each node consists of four children. The picture is maintained in various layers with a more refined declaration at the bottom layers due the hierarchical building of the picture. Covering a quad-tree in a shared manner through different grid computing environments is important for the DQTO building. Based on the Nearest Neighbour Discovery (NND) of the resource's values, the DQTO building is expanded to random and complex queries replacing the binary version, which indicates to the queries that are questioned as "is there a resource value?". By agreeing that the nodes of the grid computing environment sit on two-dimensions, their locations (X, Y) are accessible to each other.

The network is categorised into different matrix cells when a DQTO is embedded through the network. The smallest cell area of the DQTO building is figured by one lowest depth node (level one) of the DQTO where the entire grid nodes of the interior level of a single rectangle are ensured to have a single hop distance. In the proposed jargon, a "computer" indicates to a real distributed node of the computing environments where a "node" indicates to a virtual DQTO node (e.g. level one rectangle) [14], [15], [28]–[30].

The price of inquiring for a value of a resource is calculated based on the number of hops that are moving along from the inquiring computer to the computer, which holds an advertisement regarding that particular resource[31]. A distributed Quadtree overlay is built based on the use of a programming trick, which is used in [32]. For example, the entire level one rectangle of the building is given an index which entirely identifies one area. The index's length is equal to the number of levels. This fact is based on addressing a method that maintains the location of the node information. According to the technique of building level one rectanglular lens, there exist a free programming of the number of nodes. However, this type of programming is based on the division levels. Fig. 3 illustrates different nodes' indexes within a partitioned area of three divided levels. For the entire division level, a node is considered a parent node within the area. The parent and its own child are put together in their lower levels at all times. A node in the DQTO could fit into several levels through the hierarchy according to its site. If a node at level H is considered a component, it can also be a component at level one. Additionally, the neighbouring nodes are considered siblings, which indicate to 'p. sibling'.

This structure is simple where it adjusts to multi-dimensional computer resources, such as the RAM size, CPU speed, Bandwidth speed and Hard Disk space. Nonetheless, building the DQTO does not depend on the value of the resource node. Another difference of the centralised quad-tree and the DQTO is that there is no need for the DQTO to have the tree root. The four nodes exist on the top level function forming the root.

The Fig. 1 shows a two-dimensional array as a matrix, which contains 64 nodes where a product of 23 X 23 with integer ( X, Y ) coordinates is of $0 \leq X \leq 7$ and $0 \leq Y \leq 7$. A matrix can reflect a grid of numbers that are aligned in rows and columns by using this data structure type for encoding the information of any interesting things. In order to allocate each element, two nested loops should be used for a two-dimensional array. This provides a counter variable for every column and row within the indicated matrix. Additionally, a two-dimensional array is used for storing many objects. This array is appropriate for programming sketches, which involves a few types of "grid" or "board". Figure 1 demonstrates a grid of cell objects that is stored into a two-dimensional array. Every cell forms a rectangular shape, which contains a number of nodes ranging from 1 – 64 nodes [26].



ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Figure 1: The two-dimensional array forming a matrix of 64 nodes*

The DQTO is implemented in the place building as a replacement for the bottom-up building in order to minimise messages that are being transferred through to the main building. In place and the indexing of the rectangle is implemented by a static mechansim such that this type of indexing is considered appropriate for calculating each neighbouring level and its parents. Every node can have neighbours at different positions in the West (W), South (S), East (E) and North (N). The following equations can search for the parents and neighbours at level x [32]. The Z order or Morton order is considered to be a space-filling curve in which a multidimensional information is mapped in a single dimension while retaining the information points' locality. In a point of multi-dimensions, the Z value is designed by incorporating a binary demonstration for the values of its (X, Y) coordinates. When the data is aligned based on this order, any one-dimensional data structure could be used, such as the binary search trees [32]. The Z ordering is being used to effectively build the quad-trees and to involve structures of higher dimensional data. In the DQTO, the DQTO address of level one partition by every node where this address exists in the node's (X, Y) coordinates. Assume that (Xs, Ys) (where s denotes the start) at NW and (Xe, Ye) (where e denotes the end) at SE form two endpoints of the territory such that the DQTO should be overlaid. Assume also that x levels are included in the DQTO. The part of each level of a single rectangle of a partition comprises:

$$Width \quad (W) * Length \quad (L) \qquad (1)$$

Where width,

$$W \quad = (Y_e - Y_s) \div 2^x \qquad (2)$$

And length,

$$L \quad = (X_e - X_s) \div 2^x \qquad (3)$$

Then, the DQTO index that belongs to node (X, Y) is calculated as follows:

$$DQT \quad Index = \left[ \left| \frac{x - x_S}{w} \right| (Mod\ 2) \right] + \left[ \left| \frac{Y - Y_S}{L} \right| (Mod\ 2) \right] * 2 \qquad (4)$$

For example, in order to index the node number 37 according to Fig. 1 above based on the steps of the indexing algorithm, the output will be 211. If the same indexing algorithm for every node is implemented, the output is given, as shown in Figure 2.

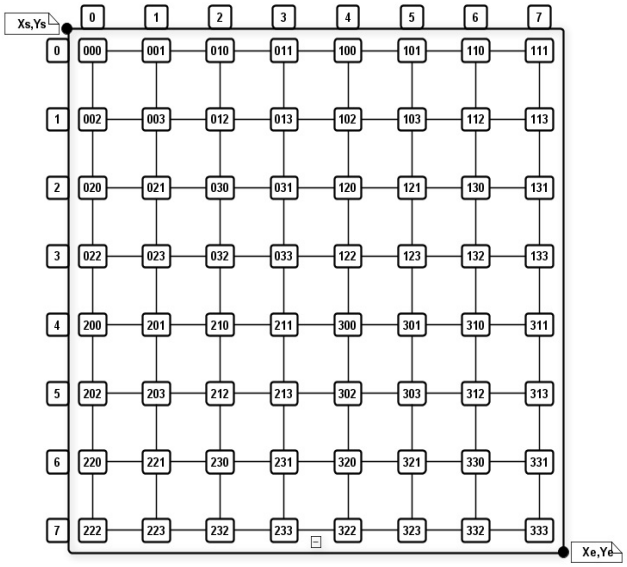
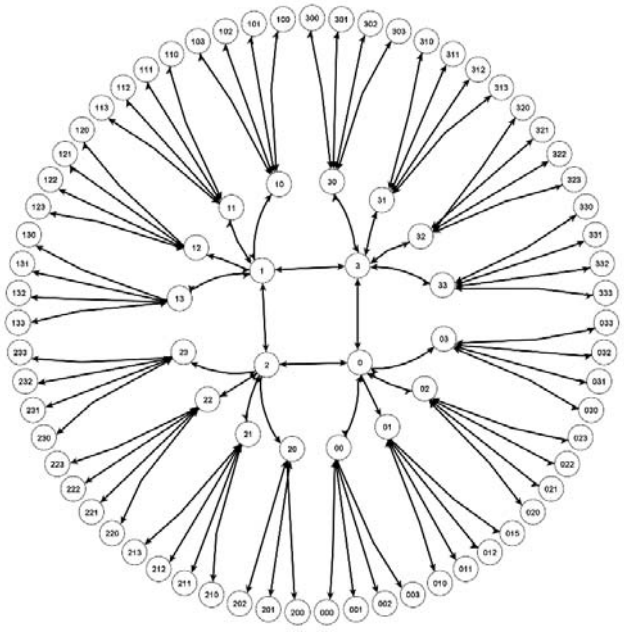



*Figure 2: The two-dimensional array forming a matrix of 64 nodes after applying indexing algorithm*

Figure 3 shows the hierarchy of distributed quad-tree containing 64 nodes when every node is already indexed and given a unique ID number. The single top root is removed since a single failure point is symbolised.



Journal of Theoretical and Applied Information Technology
31st October 2020. Vol.98. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Figure 3: A distributed quadtree hierarchy with a radial view*

| | |
|---|---|
| **Neighbors** | Y= 2, by adding and subtracting 1 from each coordinate so 012, 021, 032, 031 are my direct neighbors. |

## 3. THE DQTO ROUTING ALGORITHM

All The routing algorithm is considered important as it indicates to how nodes converse among each other when information is broadcasted to facilitate selecting routes between any two nodes where the most effective choice of the route is searched for. The paper not only explains the selection of the most effective paths of a network, but also explains the way of how messages are being forwarded as broadcast, multicast, geo cast and unicast. As each node has an earlier knowledge only of nodes those are immediately connected to it over the networks. This information is disseminated by a routing algorithm among direct neighbours, and after that, over the whole network. The following sub-objectives are introduced when the routing algorithm is applied. The first sub-objective states that the nodes obtain knowledge regarding the network topology [26]. The second states that according to the updated messages within the networks, an ultimate network traffic does not exceed Log (N), as shown in Table 1.

*Table 1: Summary of Detection Algorithm with Example*

| Algorithm | Function |
|---|---|
| **Total Number of Node** | Since it is index consist of 3 digits that mean 23 x 23 = 64 and this is the whole numbers of nodes within the network. |
| **Number of levels** | Since it is index consist of 3 digits that means the tree consist of 3 levels. |
| **My Children** | Since my index equal 030, I will add a number from 0-3 to the end of my index to know my children's: 0300, 0301, 0302, 0303 |
| **My Root** | My index equal 030 it starts at 0, and this represents my root. |
| **My Parent in each level** | My index equal 030 my parent is 03 at level 2 |
| **My Brothers** | My index equal 030 I will replace the last digit in the index with a number from 0-3. (e.g., my index is 030 so 031, 032, 033 are my brothers or siblings) |
| **Number of Roots** | Always is fixed to 4 roots. |
| **Brother Root** | My index equal 030 it starts at 0, and this represents my root, so we replace the first digit with 1,2,3, and these are the roots brother to my root. |
| **Exact location** | Since using the position encoding for X, Y coordinates. |
| **Direct** | My index is 030 and my location is X= 2, |

The connectivity protocol contains two operations, which puts an impact on the network structure when new requests for connecting the network are made by a particular node where joining, in general, is considered easy. This protocol only calculates its (X, Y) coordinates. The required message is routed over the hierarchy in order to search for its parent node whose index approximately contains the same index pertaining to the new node. After that, this information is sent by a message to two hops in order to guarantee the its parent's arrival to the network. Finally, the parent checks out the index as to whether it is available or not based on the use of the IP address. If that index is previously available, the parent will repeat with another indexed message for including the node by inserting the latest digit to the final index. Meantime, the nodes of the upper-level parent is updated by the parent in the hierarchy. For example, assume that a new node needs to connect through to the network in an area of X=3 and Y=2, as shown in Figure 4. The new node will immediately calculate its index 031 and transfers its information by three hops starting from its location and till the arrival of its parents. The parent checks the existing index and produces a replaced indexing message in order to involve it to its lowest level within the hierarchy, as shown in Figure 5.

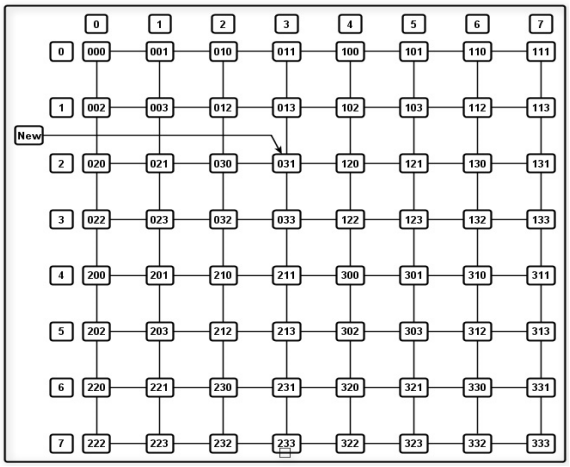



*Figure 4: New nodes connecting the DQTO*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

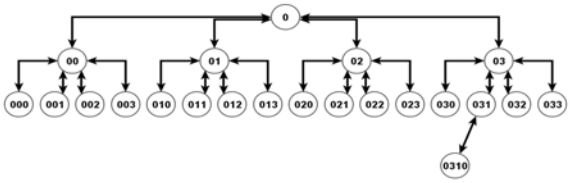



*Figure 5: The resulting hierarchies for the node that is connecting the network*

Assume that there are two types of leaving the network, which comprise the selective leave and the force leave. In the selective leave, a leave message is transferred by a leave node along to its parent. Once it arrives, parent ensures the list of its children for the least index node in order to be replaced by the leaving node. When this process is performed, an update summary message is sent by the parent to its higher parent, and meantime, a message is sent to the entire children in order to remove the leaving node from its particular list. In the force leave, leaving the parent involves similar steps to those indicated in the selective leave. Nevertheless, the difference is that the parent discovers that the child is dying when a life message is being transferred regularly. You can refer to the distributed Quadtree joining algorithm in [33] for more details.

## 4. INDEXING OF RESOURCE'S VALUE

The DQTO is considered a hierarchical organisation that is similar to many different hierarchical organisations. Hence, several levels of bordering nodes or many different hierarchical organisations are distant from one to another. However, they are still near from each other. This concern can start with a high latency if the inquiry traces back the hierarchy's path efficiently. The result is to employ different relations of a relative in order to close the root nodes. A relative's relation is the link between a node and its neighbours within each involved path. The relatives only provide a combination among their nodes on the same level of their organisation. A node at level x maintains the value of the resource that is related to its area and to its neighbouring nodes. Additionally, a node at level x maintains a range of a [*Min, Max*] information within its area for every concept. Higher level nodes hold broader space ranges, but a less wide information that is related to its sub-tree. A range of information is swapped by root nodes in a manner where each root identifies the range of the whole mechanism. When the value of a resource is defined at level one node p, the p connects to its direct parent node at level one. The parent node adjusts its own record pertaining to that child. Node p also connects to its relative's nodes in order to adjust their records.

The update operation is repeatedly carried up to the highest level. This is matching with the storage of the information scheme that is indicated and with the relatives' links in Stalk. The NND is defined as exploring the value of a resource that is nearer to the explored node, which is provided with a set of nodes. The classic NND precisely returns a single node as a final result. In the share of the computing environments network, an inquiry initialises at any particular location. The originator of a discovery refers to the node where the discovery is being transferred into the system. The discovery point refers to the node for which one can obtain the result of the NND. The discovery point is by default considered the same node as the originator of inquiry. Nonetheless, it can be identified at any point within the network. This algorithm evades the distribution of queering to the highest levels when it can be locally answered. Throughout charming benefit of the spatiality information, the latency and discovery efficiency are both being enhanced. The proposed discovery approach begins its discovery at the discovery point based on a local information according to the basis in which a node can refer to many different levels. Thus, multi-layer information is locally being controlled. If no any result is accomplished, the query is repeatedly broadcasted to its particular parent. When the value of a resourcing information arrives at particular levels, the discovery is terminated and is transferred back to its originator. If the inquiry point is located at a different place, this will imply that the originator of the query point and the query will belong to two different nodes. Basically, the query is transferred through to the query point from the query's originator including a Greedy Perimeter Stateless Routing (GPSR), which represents a geographical routing technique. This in fact implies that the data packages are not transferred to a certain receiver, but to certain coordinates.

The packages must be transferred to the node, which is physically nearer to their belonging coordinates. This assumes that each node should identify its own position. The algorithm terminates the propagation of finding top levels if it is locally responded. If there is no any obtained output, the inquiry is accordingly propagated to its parent frequently. When the event of information arrives at particular levels, the query is after that terminated and transferred back to its originator [31].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

## 5. METHODOLOGY AND SIMULATION SETUP

Simulation setup involves some series of phases as shown in Figure 7. The performance characteristic of SORDMs is investigated by adopting the J-Sim simulator. The main focus of the simulations includes examining the relationship between network size, the average number of hops, average number of messages, average number of communication time and the average number of memory utilization in various scenarios.

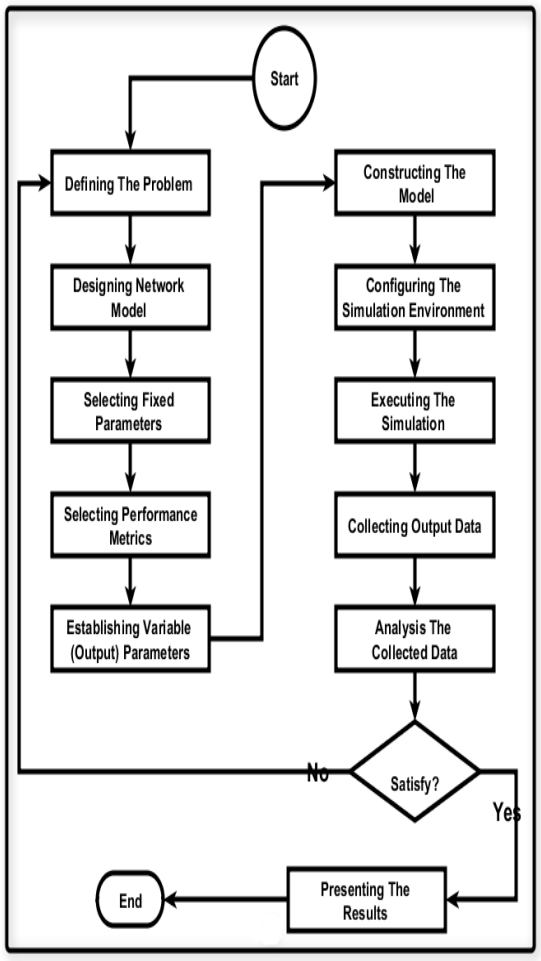



*Figure 7: Simulation Setup*

The research experiment is simulated with 4096 nodes, which has a uniform distribution in a 2-dimensional array of 64 _ 64. Node to node distance is 2m. Therefore, the average degree is about 4 in the field excluding certain border nodes in case of DQTM. This implies that not all level 1 neighbors are within reach through a single hop. There is the availability of the geographical location of every node and this is adopted in the construction of the DQTM structure in the first stage.

The DQTM height is 5, having 4 roots at the top level. The resource querying cost is estimated as the hops-number from the querying node to the node holding an advertisement for the resource. The summarizes the simulation setup shown in Table 2.

*Table 2: Simulation Setup.*

| Parameter | Value |
|---|---|
| Network size (number of nodes) N | 4 to 16384 is similar to 2^i * 2^i where i = 1-7. |
| Number of Backbones | N/2^i-1 * 2^i-1 |
| Number of Nets | N/2^i-2 * 2^i-2 |
| Number of Networks | (N/2^i-4 * 2^i-4) *2 |
| Mean Cluster Size C | 4N |
| Overall Number of Resources Offered by the Network vs. the Network Size | 4((log2N)-2)2 |
| V: maximum number of neighbors to which a parent-peer can forward a query | 1 to 4 |
| Time to live TTL | (log2N)/2 |
| Mean query generation time MQGT | 300 sec |
| Percentage of leave nodes 4 to 16384 is similar to 2^i * 2^i where i = 1-7. | from 2% to 25% is similar to N/4 |
| Percentage of joining nodes | from 2% to 50% is similar to N/2 |

The comparisons among base Chord, Recent Visited Node based Chord, Fuzzy classification based Chord and the proposed SORDM are made. Conducting a series of computational experiments showing that the SORDM achieves more scalable, stable and efficient lookup performance than other existing approaches. Experiments also show that the approach has little dependence on factors such as resource density, query type, and shared computing infrastructure size.

## 6. PERFORMANCE INDICES

The main goal of this research is to propose and develop a Self-Organizing Resource Discovery Mechanisms for Shared Computing Infrastructure (SORDM) that can be combined with the P2P approach to improving the performance of the shared computing infrastructure based on the exponential growth of the concept of distributed and decentralization. The DQTM is used for the node discovery that can locate the nodes very fast,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and support the network's high dynamicity regarding joining and leaving node as well as removing the single point of failure, so there is no bottleneck. Hence, it achieves distance-sensitive in-network querying (locality). Through the distributed hierarchy, there will be a support of efficient information storage and maintaining of a minimalist structure, which is considered as stateless. The construction of a distributed hierarchy is local and requires no communication.

The discovery mechanism proposed was initially analyzed for a shared computing infrastructure with a fixed number of nodes 4096, and a mean cluster size C within the range 1 (which is equivalent to a fully decentralized P2P network, where there is a coincidence between peers and parent-peers) to 1024 (which is equivalent to a network comprising of four clusters). Varying values of V and TTL were tested, with the aim of analyzing how to tune the parameters for enhanced performance in a case where there is knowledge of the mean cluster size. There is the high relevance to the knowledge of this information, especially when tuning the value of TTL. For instance, considering a case where the number of results is to be maximized, with C = 256, the corresponding value of TTL should be TTL >= 3, whereas if the value of C is greater than 500, it is of no use increasing the TTL value beyond 3. Hence, obtaining a very small number of results for a decentralized network, that is, having a cluster size of 1. This scenario shows the advantages attached to using the Parent-Peer model, as shown in Table 3.

*Table 3:summarizes the performance indices, definition, and its metrics*

| Performance index | Definition | Metrics |
|---|---|---|
| Average number of hops | The total number of hops that a packet passes through from its source to its destination within the network divided by the total numbers of nodes (N) in the network. | $ANH = \frac{1}{N}\sum_{n}^{N} = 1^{Kn}$ Where *Kn*, = $Ending\_Time_n - Starting\_Time_n$ |
| Average number of memory utilization | The total number of memory utilized or occupied by a job from original memory space in node "n" within the network divided by the total numbers of nodes (N) in the network. | $ANMU = \frac{1}{N}\sum_{n}^{N} = 1^{S}$ Where *Sn* , $\frac{MUn}{TMUn} * 100\%$ |
| Average number of messages | The total number of messages generated by a node "n" within the network divided by the total numbers of nodes (N) in the network. | $ANM = \frac{1}{N}\sum_{n}^{N} = 1^{Mn}$ Where *Mn* = refers to the number of messages issued from computer *n* |
| Average number of communication time | The total number of time taken by a message generated from node "n" within the network divided by the total numbers of nodes (N) in the network. | $ANCT = \frac{1}{N}\sum_{n}^{N} = 1^{Tn}$ Where Tn = $Arrival\_Time_n - Departure\_Time_n$ |

## 7. SIMULATION EXPERIMENTS AND ANALYSIS

The performance of the DQTO's characteristic is tested based on the use of the JSim simulator. Basically, the simulations concentrate on considering the connection among the average number of communication time, the average number of hops, the size of a network, the average number of messages and the average number of memory consumption. In many different types of scenarios, 4096 nodes are efficiently simulated in the experiment of this paper, which is disseminated through a 2-dimensional array of a 64 x 64 matrix. The calculated distance between each single node is 2 meters. Accordingly, the average degree is approximately 4 meters within the field where some border nodes are not included when the DQTO exists. Not the entire neighbours of Level 1 are reachable through one hop. The geographic location of every node is considered reachable and is used to create the structure of the DQTO through the first phase. The DQTO tree's height is 5 meters including four roots at its highest level.

The calculation of the price for querying a resource depends on the number of hops starting from the querying node and ending with the node that holds an advertisement regarding the resource. In this paper, the experiments focus on the leaf-level and the node-level behaviours. A group of computational experiments are formed in order to identify the performance of the DQTO infrastructure. The main group of these experiments is applied in order to test the impact of the system size and resource density according to the average number of communication time, the consumed memory, the average number of hops and the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

average number of messages. The DQTO is implemented into the Jsim simulator with N = 256, 516, 1024, 2048 and 4096. The differences between the DQTO and the fuzzy classification based Chord are carried out [23].

A sequence of computational experiments is conducted where it shows that the DQTO technique achieves more effective, scalable and steady lookup performance in comparison with other available approaches. Additionally, it is found to be proven from the conducted experiments that this method contains a minute dependence on factors, such as query type, grid size and resource density. Further, some experimental interpretations are validated by a theoretical investigation.

The differences between the DQTO [14], [29], [34] and the fuzzy classification are performed based on the average number of messages, communication time, memory consumed and the number of hops. Calculating the average through 10 runs of every experiment, the average number of hops and average communication time are extremely minimised in the DQTO when the average of 10 runs is calculated through each experiment. Consequently, it can be inferred from the experiments that the DQTO performs more effectively in comparison with the Fuzzy-Chord.

### 7.1 The Average Number of Messages

When applying the DQTO for resource discovery through distributing the computing infrastructure, the condition of broadcasting queries is removed and updates cross through the intended network. Comparing our DQTO with numerous other existing mechanisms by means of the average number of messages required to inquire about resources and the capability to assurance in finding corresponding resources in case they present wherever in the system. There exist many resource discovery techniques (e.g. Gnutella) that make use of different broadcasting methods for querying processes, which utilise Time-to-Live (TTL) counters in order to specify the number of formed messages. TTL counters are reduced every time once a query is being transferred to a current node. Once the TTL counter arrives at zero, the query terminates and is not progressed anymore within the network. Since queries in particular systems are not delivered to every node, resource discovery is not accordingly assured. If a resource is seen to be available, but not inside the query “horizon”, it will not be identified. One criteria pertaining to the proposed system of this paper ensures that it provides a reply to a query if it is available, without the need for a query to be delivered to every node within the network. When presenting the notion of local caches through the DQTO, the amount of messages that are required to provide a reply through to a particular query is reduced. Meantime, if a matching resource is available, the DQTO will be able to locate this resource. Figure 6 shows the results of the conducted experiment.

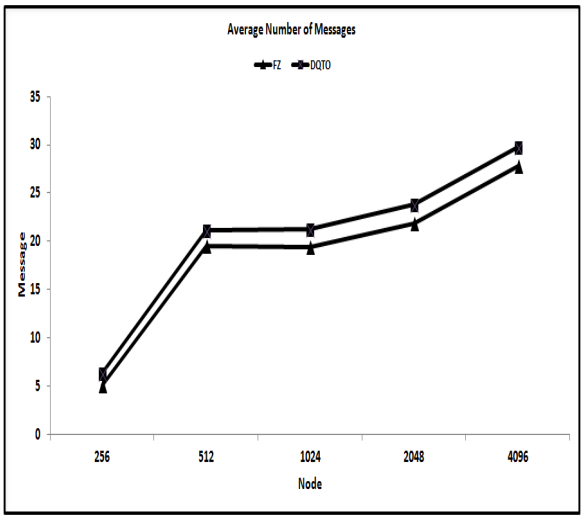



*Figure 6: The average value of messages is obtained for the FZ and DQTO techniques*

### 7.2 The Average Number of Hops

One of the most significant and successful factors, which facilitates utilising large scale resources of sharing systems is the existence of well-ordered resource discovery techniques. Hence, the major concerns of a resource sharing information system comprise the minimal network overhead and scalability. Accordingly, by proposing an answer based on DQTO and proactive information caching supported by a local structured overlay topology.

The proposed technique is characterised by its self-organized and completely disseminated overlay structure that retains a selectively based discovery algorithm and a bounded diameter overlay, which make use of local caches in order to minimise the number of visited nodes. To enhance the caching scheme when remaining a minimum use of the bandwidth, the cache contents are rarely swapped through neighbouring nodes, and hence, permitting the nodes to obtain an extra general outlook related to the network and its resources. Extensive experiment delivers a confirmation in which a normal number of hops that is required to effectively place resources is incomplete.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Moreover, the proposed technique of this paper performs efficiently according to the network head and the hitting rate. Figure 7 displays the results of the conducted experiment.

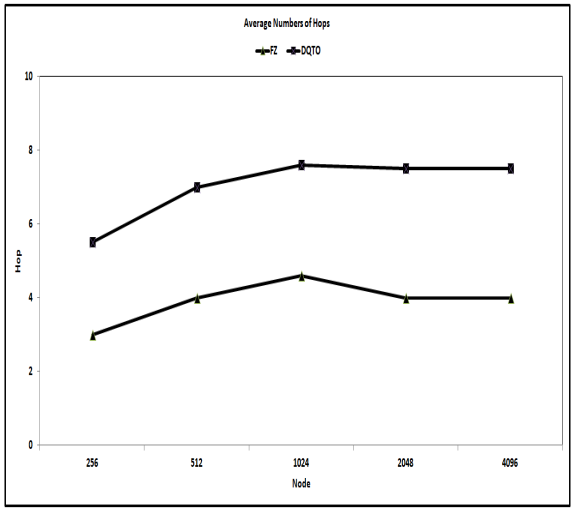



*Figure 7: The average value of hops is obtained for the FZ and DQTO techniques*

### 7.3 The Average Number of Memory Utilisation

Memory utilisation is measured as an essential metric for testing the resource discovery's performance based on the resource recovery delay. If the memory is full and the current job emerges for a particular node. However, the node does not have enough space. After that, the job is directed to the subsequent node where it spends much time on that [35][36]. The percentage of the total memory use (TMu) is expressed according to the following equation:

$$TM_u = M \div TM * 100 \quad (5)$$

Where Um denotes the used memory of the cached contents, and Tm denotes the entirely existing size of the cache. When the total reachable size of the cache is 20 GB, and the consumed memory is 5 GB, and when adding these sizes in Equation 10, we get $TM_u = 5 \div 20 * 100 = 25\%$. Here, the performance validation of the DQTO strategy is connected with the FZ. According to the obtained results from the conducted simulation, the FZ-Chord and the DQTO consume more memory when using an improved number of nodes. Nonetheless, the resource table of the DQTO is conserved by the whole nodes of the entire rings. Subsequently, the DQTO consumes more memory in comparison with the base Chord and the FZ-Chord. Since the memory is calculated based on bytes, it is not taken into account as a major issue. Figure 8 displays the results of the conducted experiment.

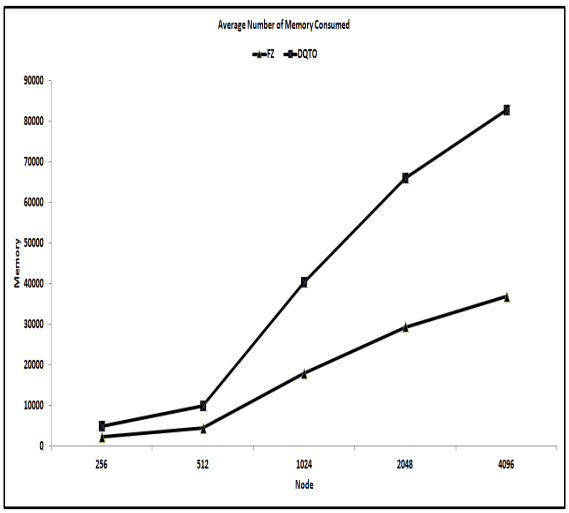



*Figure 8: The average value of the memory utilisation is obtained for the FZ and DQTO techniques*

### 7.4 The Average Number of Communication Time

The concept bandwidth refers to the propagation features of the communication systems between two nodes throughout the network. The data is calculated by the bandwidth in which a network path can move on. When the network bandwidth utilization is being reduced, the network traffic is deteriorated. In fact, the network traffic indicates to the transmission time through the network where this time refers to the period when a query is raised until the query receives a query hit. Accordingly, the bandwidth is used by the DQTO at its best. The results of the conducted experiment are illustrated in Figure 9.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

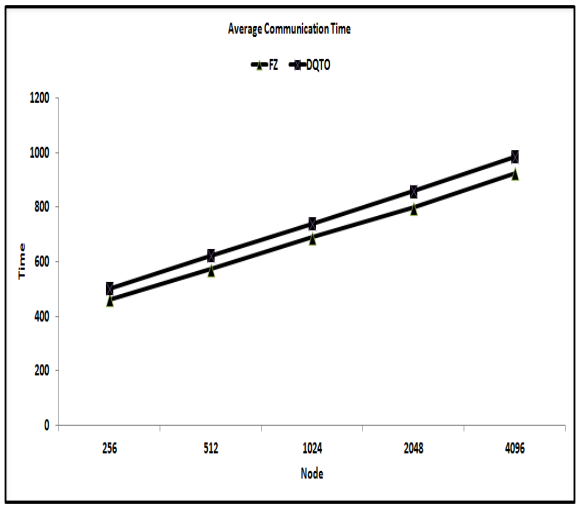



*Figure 9: The average value of the communication time is obtained for the FZ and DQTO techniques*

## 8. CONCLUSIONS

This paper proposes an in-network querying infrastructure, namely, the Distributed Quad-Tree Overlay (DQTO) structure, which is suitable for use in the real-world sharing of computing environmental deployments. The DQTO ensures distance-sensitive querying including an effective storage of the information within the network. The DQTO is locally built and does not require any communications. Additionally, due to the simple stateless and infrastructure nature of the DQTO, it shows a superior flexibility against node failures. The DQTO provides a solution by offering answers to multifaceted range queries and a "proactive caching" optimisation scheme for developing the best queries. The stateless nature of the DQTO makes it simple for the topology alteration. It is possible to enlarge the DQTO in order to provide a location service for different mobile ad-hoc networks. The suggestion is to attempt at retrying a query until this query joins the targeted mobile. Despite that a target node may produces moves when the query is being implemented causing a miss, the query that is appealed from the current location, and which is closer to the destination node will have the opportunity to connect through to the destination node. The reason behind this connection refer to the property of the distance-sensitivity within the DQTO structure. Consequently, it can be inferred from the obtained results that are derived from the simulation that the number of messages and communication time hops is reduced. Finally, the DQTO successfully reduces the number of communication time, hops, and messages. Except the DQTO, which needs a modest additional memory for storing the values of a resource compared to the Chord. Further, the DQTO will not be a subject nowadays as memory is measured in provisions of terabyte.

## REFRENCES:

[1] G. Strawn, 'Masterminds of the Arpanet', IT Prof., no. June, pp. 66–68, 2014.

[2] R. Tedrake and M. Fallon, 'A summary of team MIT's approach to the Virtual Robotics Challenge', … (ICRA), 2014 IEEE …, p. 2087, 2014.

[3] S. S. Sehra, J. Singh, and H. S. Rai, 'Dynamic Scheduling in Grid Environent with the Improvement of Fault Tolerant Level', Indian J. Sci. Technol., vol. 8, no. November, pp. 507–515, 2015.

[4] Q. Chen et al., 'ISPRS Journal of Photogrammetry and Remote Sensing Local curvature entropy-based 3D terrain representation using a comprehensive Quadtree', ISPRS J. Photogramm. Remote Sens., vol. 139, pp. 30–45, 2018, doi: 10.1016/j.isprsjprs.2018.03.001.

[5] S. M. B, A. Sivasubramanian, and S. P. K. Babu, 'A Review of MAC Scheduling Algorithms in LTE System', Int. J. Inf. Technol. Adv. Sci. Eng., vol. 7, no. 3, pp. 1056–1068, 2017, doi: 10.18517/ijaseit.7.3.2059.

[6] W. Alrawabdeh, 'Internet and the Arab World: Understanding the Key Issues and Overcoming the Barriers.', Int. Arab J. Inf. …, vol. 6, no. 1, 2009.

[7] A. Jain and R. Singh, 'An innovative approach of Ant Colony optimization for load balancing in peer to peer grid environment', Issues Challenges Intell. …, pp. 1–5, 2014.

[8] A. Singh and P. Chakrabarti, 'Ant based resource discovery and mobility aware trust management for Mobile Grid systems', in Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013, 2013, vol. 25, no. 4, pp. 637–644, doi: 10.1109/IAdCC.2013.6514301.

[9] B. G. Batista, F. C. Teixeira, M. J. Santana, and R. H. C. Santana, 'Scheduling algorithms for opportunistic computational grid based on television digital receiver', 2013 Int. Conf. High Perform. Comput. Simul., pp. 584–591, Jul. 2013, doi: 10.1109/HPCSim.2013.6641473.

[10] R. Myung, H. Yu, and D. Lee, 'Optimizing Parallelism of Big Data Analytics at Distributed Computing System', Int. J. Adv. Sci. Eng. Inf. Technol., vol. 7, no. 5, p. 1716, 2017, doi:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

10.18517/ijaseit.7.5.2676.

[11] J. Perez, J. Diaz, C. Vidal, D. Rodriguez, and D. Fernandez, 'Self-Balancing Distributed Energy in Power Grids: An Architecture Based on Autonomic Computing', 2014 47th Hawaii Int. Conf. Syst. Sci., pp. 2398–2407, Jan. 2014, doi: 10.1109/HICSS.2014.301.

[12] E. B. Lakew, F. Hernandez-Rodriguez, L. Xu, and E. Elmroth, 'Management of distributed resource allocations in multi-cluster environments', 2012 IEEE 31st Int. Perform. Comput. Commun. Conf., pp. 275–284, Dec. 2012, doi: 10.1109/PCCC.2012.6407768.

[13] H. Viswanathan, S. Member, and E. K. Lee, 'Uncertainty-aware Autonomic Resource Provisioning for Mobile Cloud Computing', IEEE Trans. Parallel Distrib. Syst., vol. 9219, no. c, pp. 1–1, 2014.

[14] S. Arif, M. S. Alzboon, and M. Mahmuddin, 'Distributed quadtree overlay for resource discovery in shared computing infrastructure', Adv. Sci. Lett., vol. 23, no. 6, pp. 5397–5401, 2017.

[15] M. Mahmuddin, M. S. Alzboon, and S. Arif, 'Dynamic network topology for resource discovery in shared computing infrastructure', Adv. Sci. Lett., vol. 23, no. 6, 2017.

[16] B. Gueye, O. Flauzac, C. Rabat, and I. Niang, 'A self-adaptive structuring for P2P-based grid', 2014 14th Int. Conf. Innov. Community Serv., pp. 121–128, Jun. 2014.

[17] R. Wattenhofer, 'Principles of distributed computing', Lect. Notes, www. dcg. ethz. ch/lectures/ss05/ …, vol. 8784, 2012.

[18] J. Zarrin, R. L. Aguiar, and J. P. Barraca, 'Dynamic, scalable and flexible resource discovery for large-dimension many-core systems', Futur. Gener. Comput. Syst., vol. 53, pp. 119–129, 2015.

[19] H. M. N. D. Bandara and A. P. Jayasumana, 'Resource and query aware, peer-to-peer-based multi-attribute Resource Discovery', 37th Annu. IEEE Conf. Local Comput. Networks, pp. 276–279, Oct. 2012.

[20] R. Tavuseh and M. Ahmadi, 'Optimization of resources discovery in grid computing using bloom filter', World Appl. Program., no. April, pp. 169–177, 2013.

[21] D. N. Meivita, M. Rivai, and A. N. Irfansyah, 'Development of an Electrostatic Air Filtration System Using Fuzzy Logic Control', Int. J. Adv. Sci. Eng. Inf. Technol., vol. 8, no. 4, pp. 1284–1289, 2018.

[22] M. H. Hassan, S. A. Mostafa, A. Budiyono, A. Mustapha, and S. Shamini, 'A Hybrid Algorithm for Improving the Quality of Service in MANET', Int. J. Adv. Sci. Eng. Inf. Technol., vol. 8, no. 4, pp. 1218–1225, 2018.

[23] C. Jeyabharathi and A. Pethalakshmi, 'New Approaches with Chord in Efficient P2P Grid Resource Discovery', arXiv Prepr. arXiv1401.2008, vol. 41, no. 8, pp. 3831–3849, Jun. 2014.

[24] E. Tanin, A. Harwood, and H. Samet, 'A Distributed Quadtree Index for Peer-to-Peer Settings', 21st Int. Conf. Data Eng., pp. 254–255, 2005.

[25] K. Zhou, G. Tan, and W. Zhou, 'Quadboost : A Scalable Concurrent Quadtree', vol. 9219, no. c, pp. 1–14, 2017.

[26] I. Al-Oqily, M. Alzboon, H. Al-Shemery, and A. Alsarhan, 'Towards autonomic overlay self-load balancing', in 2013 10th International Multi-Conference on Systems, Signals and Devices, SSD 2013, Mar. 2013, pp. 1–6.

[27] S. S. M, M. Aldeen, S. M. Ieee, W. V Stoecker, R. Garnavi, and M. Ieee, 'Biologically Inspired QuadTree Colour Detection in Dermoscopy Images of Melanoma', vol. 2194, no. c, pp. 1–8, 2018.

[28] S. A. Mowafaq Salem Alzboon M. Mahmuddin, and Omar Dakkak, 'Peer to Peer Resource Discovery Mechanisms in Grid Computing: A Critical Review', in The 4th International Conference on Internet Applications, Protocols and Services (NETAPPS2015), 2015, pp. 48–54.

[29] M. S. Alzboon, M. Mahmuddin, and S. Arif, 'Resource Discovery Mechanisms in Shared Computing Infrastructure: A Survey', in International Conference of Reliable Information and Communication Technology, 2019, pp. 545–556.

[30] V. Doraisamy, S. Alomari, "Video on Demand Caching System using NIPBCS over Mobile Ad Hoc Network, "International Journal of Digital Content Technology and its Applications, Vol. 5, No. 6, pp. 142-154, 2011.

[31] M. S. Alzboon, A. S. Arif, and M. Mahmuddin, 'Towards self-resource discovery and selection models in grid computing', ARPN J. Eng. Appl. Sci., vol. 11, no. 10, pp. 6269–6274, 2016.

[32] M. Demirbas and X. Lu, 'Distributed quad-tree for spatial querying in wireless sensor networks'. ICC'07. IEEE Int. Conf., pp. 3325–3332, Jun.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2007.

[33] D. Wise and J. Frens, 'Morton-order Matrices Deserve Compilers' Support Technical Report 533', pp. 1–18, 1999.

[34] A. Clematis, D. D'Agostino, a. Quarati, a. Corana, V. Gianuzzi, and a. Merlo, 'A Distributed Approach for Structured Resource Discovery on Grid', 2008 Int. Conf. Complex, Intell. Softw. Intensive Syst., pp. 117–125, 2008.

[35] I. Al-Oqily, M. Alzboon, H. Al-Shemery, and A. Alsarhan, 'Towards autonomic overlay self-load balancing', 2013.

[36] N. Jafari Navimipour and F. Sharifi Milani, 'A comprehensive study of the resource discovery techniques in Peer-to-Peer networks', Peer-to-Peer Netw. Appl., Mar. 2014.