Scalable Search based on Fuzzy Clustering for Interest-based P2P Networks

Romeo Mark A. Mateo and Jaewan Lee
School of Electronic and Information Engineering, Kunsan National University
68 Miryong-dong, Kunsan, Chonbuk, South Korea 573-701
[e-mail: {rmmateo, jwlee}@kunsan.ac.kr]
*Corresponding author: Jaewan Lee

Received July 9, 2010; revised August 10, 2010; revised October 22, 2010; revised November 27, 2010; accepted December 11, 2010; published January 31, 2011

Abstract

An interest-based P2P constructs the peer connections based on similarities for efficient search of resources. A clustering technique using peer similarities as data is an effective approach to group the most relevant peers. However, the separation of groups produced from clustering lowers the scalability of a P2P network. Moreover, the interest-based approach is only concerned with user-level grouping where topology-awareness on the physical network is not considered. This paper proposes an efficient scalable search for the interest-based P2P system. A scalable multi-ring (SMR) based on fuzzy clustering handles the grouping of relevant peers and the proposed scalable search utilizes the SMR for scalability of peer queries. In forming the multi-ring, a minimized route function is used to determine the shortest route to connect peers on the physical network. Performance evaluation showed that the SMR acquired an accurate peer grouping and improved the connectivity rate of the P2P network. Also, the proposed scalable search was efficient in finding more replicated files throughout the peer network compared to other traditional P2P approaches.

Keywords: P2P system, interest-based P2P, cluster analysis, fuzzy clustering, neuro-fuzzy system, multi-ring topology

This research is supported by Industry and University Research Consortium of Small and Medium Business Administration (SMBA) of 2010.

DOI: 10.3837/tiis.2011.01.009

1. Introduction

The peer-to-peer (P2P) system is a popularly used technique in sharing resources over the Internet because it promotes highly available data by means of replication. The earliest technology of P2P was based on a centralized architecture like the Napster [1] where the system consists of file directory servers to store file information associated with peer location. Later, research studies start infusing P2P with various techniques of distributed system to be more robust against failures in sharing data. Decentralized approaches in P2P are used to overcome the problem of network congestion observed from centralized techniques where file queries are forwarded to selected peers which have the file location information. A P2P overlay network is a decentralized approach where it creates a logical network prior to the selected routing scheme or topology structure of the physical network which spans a wide spectrum of communication frameworks to build a self-organizing system [2]. Overlay network models are classified into two types which are the structured and unstructured P2P overlays. The unstructured P2P overlay does not have specific rules to control the routing of peers as in Gnutella [3] where peers are formed by joining nodes with loose rules. On the other hand, peers in structured P2P overlay are aware of the logical network topology. A ring topology is most commonly used for scalable P2P networks [4][5]. However, the major shortcoming of a ring topology is having a high dependency on a large number of nodes. Adding more peers in the ring will observe more delays in sending messages. Other than the ring scheme, there are topologies that are used for fast search like hierarchies and distributed trees [6] but these generate high message overheads and are more vulnerable to failures.

The overlay network is situated between the users and the physical network where the interactions of routing and topology construction in the physical network are hidden to P2P users. Another technique of connecting peers is using the interest of peers which is a user-level approach. The connections can be presented by communities of users in a P2P system where the peer network is automatically or manually formed based on the interest of peers. Interest-based communities are a natural arrangement of distributed systems that prune the search space and allow for better dissemination of information to participating peers [7]. Nowadays, the assumptions from the interest based P2P are applied in social networking. Studying the relationship between the interests of users and their logical connections is necessary to analyze the peer connectivity. In [8] simply analyzes the connectivity of a user to its friends and followers based on the message posts as data. A more complex work is to process a multiple set of data (e.g. favorites, hobbies, etc.) to find similarities with among users in the network and use this to construct the peer connections. Also, some limitations from previous interest-based P2P are as follows:

- Weak relativeness of peer connectivity.
- Scalability is ignored where users are not aware of other relevant communities.
- Routing of queries in the physical network is inefficient because topology-aware technique is not considered.

Peers are connected by similar properties; however, the relativeness of peer connections is weak if most of the interests of peers are not similar. There are various methods to measure the relativeness of peer connections and a simple way is to get the ratio of similar properties to all properties between two peers. Processing a cluster analysis to the interests of peers acquires a high accurate peer grouping but is more complex. Intelligent data analyses [9] are usually used to discriminate data and then cluster or group the data based on their similarities. Providing network boundaries among peer clusters where relevant peers are grouped is a good approach to minimize the message overheads. However, only relevant peers are grouped and thus the connectivity rate of peers in the network lowers. The scalability is also necessary to provide more peers to answer a query and should be handled efficiently. Moreover, infusing a topology-aware technique on the interest-based peer grouping improves significantly the routing of messages which must be considered.

The scalable multi-ring (SMR) uses a fuzzy clustering to perform the scalable groupings of peers which were our previous work [10]. This paper proposes a scalable search utilizing the SMR for the interest-based P2P system and improves the SMR with a topology-aware technique. The construction of the fuzzy system and assignment of groups to peers are processed by the distributed coordinators. After the group assignment, each peer group forms a P2P multi-ring network using the proposed minimized route function to locate the nearest peer neighbors of a peer node for the topology-aware technique. The proposed scalable search is used to query more peers using the SMR. Load balancing is also considered in the proposed scalable search where in a collection of peers with the queried resource, a peer is selected using the least loaded node selection. The result of using the proposed scalable search for queries efficiently finds peers with the file replicas.

2. Related Works

2.1 Topology-Aware and Semantic-Aware Techniques in P2P

Without a prior knowledge of shortest routes in forming the overlay connection, nodes may select a longer route which generates unnecessary message overheads and apparently the response time to queries is more delayed. A topology-aware technique for a P2P system significantly improves the message routing on a large network of nodes by automatically choosing the nearest routes. HRing [11] improves a ring topology where it constructs routing tables based on the distance between nodes instead of node IDs to eliminate the effect of long link distributions of node IDs. Microsoft research proposes a topology-aware routing method for P2P using proximity neighbor selection [12] to reduce the cost of overlay construction that improves Pastry. The proximity neighbor selection uses the routing table of a node to select the nearest hop to a destined node which is the same as in [11]. A multi-ring network topology [13] is proposed to provide high performance group communication in large-scale P2P networks. The multi-ring topology balances the advantage of a ring topology and a hierarchical topology. Similar to Pastry [4], each ring is connected to two neighbor nodes, left and right, in order to form an outer ring. Nodes having higher bandwidth are selected to be assigned as inner nodes that form the inner ring to provide shortcuts in all ring sections.

Most practices on Web applications and P2P networks utilize the interest of a client as an input in selecting appropriate data or resources. The routing of queries on a typical P2P network, which is not aware of peers' interest, is dependent on the message complexity of the network topology. On the other hand, the interest-based or semantic-based P2P network establishes main links to the peers that have similar interests. Using this design, a peer has a higher possibility to answer another user's query if that peer has a similar interest with the user. Shared files or interest profiles are usually used as inputs to determine similarities among peers in establishing peer overlay connections [14][15]. However, the multiple links of peers produce high overheads caused by message overflows throughout the network in performing a query, especially in a mesh network where connections are mostly done in multi-hop links. Interest profiling exploits semantic approaches [16][17][18] to implement the intelligent search in P2P which improves the peer query. In [16] shows a distributed RDF (resource description framework) query architecture built on P2P where the semantic topology is formed by using a distributed interest-index to route the query. A proximity measure to capture and exploit the semantic relationships between peers in a file sharing system is shown in [17]. A containment-based proximity metric to compare each peer interest is provided in [18]. The approach of publish/subscribe method simply forwards messages to the interested peers and stops forwarding if a peer is not interested with the message subscription. The connection of peers forms a hierarchical structure based on their interest contents.

Research studies also considered both topology-aware and semantic-aware techniques in a P2P system to exploit the advantages of both techniques. An unstructured P2P interest-based grouping is proposed [19][20] to improve the flooding scheme of the search procedure. The interest-based shortcuts in [20] are used by the peers to avoid unnecessary overheads from flooding. The study in [21] uses a Distributed Hash Table (DHT) for the location information of services and a deterministic strategy performs the semantic matching between service profiles and peer interests. The group-based P2P network model in [22] constructs a peer group based on interest and uses a value-headed walk search algorithm where a peer forwards query to a subset of its neighbors based on their peer value. A topic-group approach constructed by hierarchical layers is presented in [23] which performs dynamic grouping using the shared resources of peers. However, most researches use deterministic approaches and do not consider the uncertain properties which can improve the scalability of peer grouping. In our work, we use an unsupervised clustering to establish the interest-based P2P network and extend the multi-ring topology into a scalable multi-ring by using fuzzy clustering to promote scalability to the interest-based peer grouping.

2.2 Clustering and Classification

The interest-based P2P is improved by clustering using the interest data of peers. Calculating the similarity of each peer and ranking the best peers for clustering methods are commonly used for fast search [24]. In the research of a cluster-based P2P system [25], all participating computers are grouped into various interest clusters based on hypercube topology. A hypercube topology employs an *n*-cube network to map data which is a well-known method in parallel computing environments. The hierarchical clustered structure of a hypercube is used to route the query based on the peer's interest. The architecture of our interest-based P2P system adopted a knowledge-based peer grouping which connects the most relevant peers and provides a structure for scalable peer grouping. We refer the term knowledge-based peer grouping as the use of any knowledge-based method to group or cluster the nodes in a P2P network. Relevant peers are identified if they have near distance values in processing the scalable peer grouping.

A cluster analysis divides data into groups such that similar data objects belong to the same cluster and dissimilar data objects to different clusters [26]. Partitioning methods construct c partition of data, where each partition represents a cluster and $c \leq k$, k is the number of data. A classical objective function is shown in Equation 1 which is used to optimize the compactness of each group i to minimize J.

$$\min \sum J = \sum_{i=1}^{c} \left(\sum_{k=1, \mathbf{u}_k \in c_i} \left\| \mathbf{u}_k - \mathbf{c}_i \right\|^2 \right) \tag{1}$$

The compactness value is acquired by calculating each distance of data or vector ($\mathbf{u}_k$) in a group to a cluster center ($\mathbf{c}_i$) and then summing all distance values in each group. The compactness of a group depends on the distances between vectors $\mathbf{u}_k$ and cluster centers $\mathbf{c}_i$. If the data grouping is not optimal then the procedure adjusts the groups by a routine or a technique which assigns data to the appropriate group to produce a more compact group. In Equation 1, J represents the summation of the total distance values. The objective function minimizes J by several iterations and stops if either the improvement over the previous iteration is below a certain tolerance or J is below a certain threshold value. The result from this method only represents crisp membership and limits the scalability in classifying. In strong similarity to the hard clustering algorithm, fuzzy clustering was derived [27]. Fuzzy clustering employs fuzzy partitioning, where a point can belong to several clusters based on membership degrees. However, it observes poor classification accuracy and can be improved by adjusting the fuzzy system based on some rules extracted from data. In our proposed scalable multi-ring, the peer grouping is based on fuzzy clustering and the fuzzy system is trained in the fuzzy set learning algorithm of neuro-fuzzy classification [28]. Neuro-fuzzy classification is a 3 layered perception which consist of the following; first layer is for inputs, $U_1=\{x_1, x_2, x_3 \ldots x_i\}$, second layer is for generating rules, $U_2=\{R_1, R_2, R_3, \ldots R_k\}$, and third layer is for output layer, $U_3=\{y_1, y_2, y_3 \ldots y_m\}$. The classes for U_3 are labeled with c which can represent the identification of a peer group. The values from input and rule layers are evaluated in the connection of hidden and output layer by minimizing errors of the network. For all output units, $c \in U_3$, the net input net_c is calculated by:

$$net_c = \frac{\sum_{R \in U_2} W(R,c) \cdot o_R}{\sum\limits_{R \in U_2} W(R,c)} \tag{2}$$

where $W(R, c)$ is a function to calculate the output of R and c, and O_R is the current output value of R.

3. Architecture of the Fuzzy-based Multi-Ring P2P System using the Scalable Search

The scalable multi-ring (SMR) [10], which is an interest-based peer grouping and an improvement of a multi-ring topology [13], is used for scalable resource sharing in P2P networks. The scalability is achieved through the connectivity of peers by having links to other peer groups and these links are identified as relevant peers based on the approximation of the fuzzy system. The assumption is that multi-hop linked peers have higher possibility of having same file. The interest data profile (P_x) from a node is used as input data for the fuzzy system. We refer a node as a peer after processing the SMR. We assume that the values from P_x are data associated with the resources that a peer shares. For example, if a peer shares several music files then its music profile will have a higher value. The peer grouping is managed by the distributed coordinators to group the peers and coordinate the construction of the fuzzy system. At first, distributed coordinators are manually assigned to selected nodes in the physical network mainly to handle the collection of interest profiles from nodes. Distributed coordinators also establish its communication to other distributed coordinators. Then, all distributed coordinators will select a coordinator node to process all the interest profile. The selected node gathers the data from all coordinators to construct the fuzzy system for peer grouping. **Fig. 1** illustrates the architecture of scalable grouping used by SMR where the overlapped peer grouping is a result of processing the scalable peer grouping. The overlapped peer grouping is translated to fuzzy sets (A, B, C) shown at the bottom of figure. x axis indicates the profile value while y axis maps the membership value to each fuzzy set. Some peers in a group are members to other groups. The peer with index 5 is classified in group A

and it is also classified in group *B* and *C*. After determining the peer groupings, each group forms the multi-ring structure.

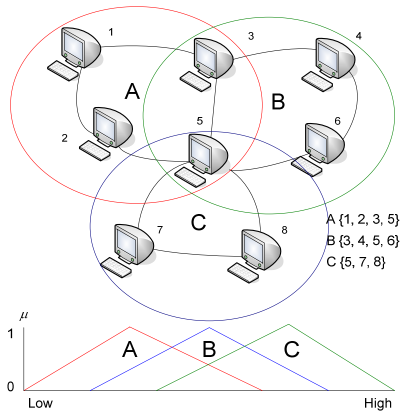


Fig. 1. The scalable peer grouping of the SMR where some peers overlap their membership to another cluster showing at the top and the membership is represented by fuzzy sets illustrated at the bottom.

We define a set of user profiles as P_x in a collection of **P** data points, ($P_x = \{p_1, p_2, \ldots p_i\} \mid P_x \in \mathbf{P}$), and each p_i is a numerical value to measure a user interest. A node hosts several files that are used for resource sharing. The P_x is sent to a coordinator node to assign the peer groups. SMR is processed into two steps; (1) initialization of fuzzy system for scalable peer grouping, and (2) construction of multi-ring using the minimized route function, which is discussed in Section 3.1 and Section 3.2, respectively. The number of peer groups is determined by processing the objective function in Equation 1 with different numbers of *c* which is less than the number of nodes (*n*) but not equal to one ($n > c \neq 1$), and then selecting a *c* index with the smallest *J*. For example, having n=10, we process the objective function in 10 cases where $c = \{1, 2\ldots 10\}$ and find the smallest J_c. In this paper, we use the number of nodes as the initial number of *c* and the next iterations will decrease the *c* to 1, (c=c-1). The summary of building and maintaining the SMR are as follows:

1. Collect the interest profiles in all nodes to **P**.
2. Initialize the fuzzy system by transforming **P** data points to *k* array values based on Eigen decomposition and use the heuristic approach to construct the fuzzy sets.
3. Adjust the fuzzy system using the fuzzy set learning algorithm.
4. Assign the group to peers using the fuzzy system.
5. Process the multi-ring using the minimized route function.
6. Calculate the classification error when a new node joined the peer network. If the classification error value is greater than the classification threshold ($\gamma > \Phi_e$) then process step 1.

The query messages from peers are sent in multi-hops using the peer network created from SMR. A copy of the query message is forwarded to every linked peer. If the query message verifies that the linked peer was already queried then it skips that peer. Also, to prevent long hops, a query is set with maximum hops.

3.1 Establishing the Fuzzy System for SMR

3.1.1 Initialization of Fuzzy System

The collection of **P** data points is transformed into $k=\{k_1, k_2, k_3 \ldots k_x\}$ using the Eigen decomposition where k_x is a summarized value of P_x and then k is used in finding the initial cluster centers before the fuzzy clustering procedure. Eigen decomposition is the factorization of a matrix into a canonical form where the matrix is represented by eigenvalues and eigenvectors. The process includes finding the appropriate eigenvectors and eigenvalues for the given matrix. Equation 3 shows that the result on finding the Eigen values and eigenvectors is equal to the given matrix. A is a square matrix from $\mathbf{P}^*\mathbf{P}^{\mathrm{T}}$ with N linearly independent eigenvectors, e_i (i=1,2,3,… N). E is a matrix whose i^{th} column is the eigenvector e_i of A. D is a diagonal matrix with the corresponding Eigen values. We move the inverse E to the other side of the equation and get the product of A and E which is also equivalent to the product of E and D where the values are assigned to the arrays of k.

$$A = EDE^{-1} \tag{3}$$

$$AE = ED = k \tag{4}$$

k values are sorted from lowest to highest preserving its information about the **P** index. The fuzzy system for the fuzzy clustering is initialized by:

$$FuzzySetLength = d(A,B) = \frac{k_{\max} - k_{\min}}{c} + o \tag{5}$$

which is a heuristic approach using the sorted k. Equation 5 is the function to determine the distance of each fuzzy set by subtracting k_{max}, the maximum value of k, and $k_{\min}$, the minimum value of k, divided by c. The overlapping of each fuzzy set is determined by o and additional to the current distance. The result from Equation 5 is applied to each fuzzy set by $d(FS)$ function where FS is a collection of fuzzy sets (q_n), $FS=\{q_1,\ldots q_n\}$. The fuzzy set length in Equation 5 is also used to calculate the value of minimum ($Cmin_n$), maximum ($Cmax_n$) and center ($Ccen_n$) values of each fuzzy set. These values are calculated in the following: $Cmin_n$ = min(q_n), $Cmax_n$ = min(q_n) + *fuzzylength*, and $Ccen_i$ = min(q_n) + (*fuzzylength*/*2*). In next iterations, the values of each fuzzy set are the addition of previous value m (m=n-1) and *fuzzylength*/*2*. Also, every cluster center is initially set to $c_n = Ccen_n$. The objective function of the fuzzy clustering is:

$$\min \sum_{n=1}^{c} J_n = \sum_{n=1}^{c}\sum_{k=1}^{X} m_{nk}^{q} d_{nk}^{2}, \tag{6}$$

where the membership function (m^q_{nk}) maps the value from d^2_{nk} using the fuzzy system initialized from Equations 3 to 5. In Equation 6, the d^2_{nk} is the Euclidean distance of a k value to c_n. The current structure of the fuzzy system built from heuristic approach can easily extract rules and this also provides customized parameters for the fuzzy set training.

3.1.2 Adjusting the Fuzzy System

We use the learning method of a neuro-fuzzy algorithm [28] to train our fuzzy system for accurate peer grouping. The initialized fuzzy sets for the fuzzy weights indicated by $\mu_{qn}^{(\mathrm{i})}$ from U_1 to U_2 is used. The classes for U_3 are labeled with cluster indexes indicated by cl_n which is the identification of a peer group. Each connection between units $x_i \in U_1$ and $R_k \in U_2$ is labeled with a linguistic term, $A_{jr}^{(i)}$ ($jr\in\{1,\ldots q_n\}$). Patterns ($\mathbf{p}_{\mathrm{x}}$, $\mathbf{t}_{\mathrm{x}}$) for training the network are the input interest profiles ($\mathbf{p}_{\mathrm{x}}\in P_x$) and target values ($\mathbf{t}_{\mathrm{x}}\in\{0,1\}$). Initially, rules are generated from the combination of p_i for the antecedents associated with $A_{jr}^{(i)}$ as an outcome of a p_i, and cl_n for the rule outcome. These rules are pruned by selecting the rules with the highest degree of membership using $\mu_{qn}^{(\mathrm{i})}$. After pruning, the best rules are determined from the training by computing the performance values of each rule. Rules (R_k) are stored and used for fuzzy set learning. The next pattern from **P** propagates through the procedure and we get the output value ***cl*** in final iteration. In our paper, the iteration of the algorithm stops until the end criterion which is the *max*. The calculation of delta value (δ_n) from each output unit is given in Equation 7 by subtracting target value to the output value from its activation function,

$$\delta_n = t_i - activation(cl_n) \tag{7}$$

$$e_R = o_R(1 - o_R)\sum\nolimits_{n \in U_3} W(R, y)\delta_n \tag{8}$$

where δ_n is a factor for the summation of weights from U_3 to U_2 . The total weight is used to calculate the error value of the rule (e_R) shown in Equation 8 where O_R is the current output value from U_2. Then, find the $\mu_{qn}^{(i)}$ with the minimal weight value from the set of inputs and rules in Equation 9.

$$W(x', R)(o_{x'}) = \min_{x \in U_1}\{W(x, R)(o_x)\} \tag{9}$$

The e_R is the error value which is used for calculating the new values for the fuzzy sets with n index determined by $W(x', R)$. Fuzzy sets are adjusted using the parameters from:

$$\begin{aligned} \delta_b &= \sigma \cdot e_R \cdot (C\max{}_i - C\max{}_i) \cdot \operatorname{sgn}(p_i - Ccen_\mu); \\ \delta_a &= -\sigma \cdot e_R \cdot (C\max{}_i - C\max{}_i) + \delta_b; \\ \delta_c &= \sigma \cdot e_R \cdot (C\max{}_i - C\max{}_i) + \delta_b; \end{aligned} \tag{10}$$

The adjusted fuzzy sets are used once more for the rule learning. Every time an update occurs in the fuzzy set training, the value of c_n, also adjusts. The result of the trained fuzzy system is then used to classify peers.

3.1.3 Handling Classification Error

Whenever a new peer joins a group, classification error may arise. This error is observed when there is a large difference of values among the existing profile patterns and the profile pattern from a new node. This can be verified by processing a cross validation to the classification method with the new pattern. To overcome this problem, we provided a procedure to handle the classification error. Equations 3 to 10 are executed by the selected coordinator node every time the peer grouping is initialized or after reaching a certain threshold value of classification error. The threshold value for classification error, represented by Φ_e, is manually configured by an expert. A higher value of Φ_e will tolerate classification errors while a lower value will regrouped the peers frequently. The calculation of classification error (γ) is:

$$\gamma = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{P}\sum\nolimits_{x=1}^{P} f(P_x, cl_x) \tag{11}$$

This is done by processing the input patterns including the new pattern to the classifier. We define **x** as the collection of interest profile patterns and each pattern P_x has a group class label cl_x. The patterns are processed in $f(\mathbf{x})$ which is the fuzzy classifier model. If the function classified a pattern correctly, then the value is incremented to 1. All patterns are processed in the classifier and the total result is divided into P, where P is the total number of profile patterns. Equation 11 has N folds of cross validating the classifier and then calculates the average. The process from Equations 3 to 10 is repeated if $\gamma > \Phi_e$. Each coordinator uses Equation 11 every time a new node joins the peer network. The coordinator with the newly joined node informs other coordinators about the event via multicast message. If there is no previous information about nodes that joined the peer network then the coordinators reply with no information to the coordinator and that coordinator starts classifying the data from the new node. If a coordinator verifies that there were nodes joined the peer network before the new node then the coordinator requests the previous data of those nodes to the coordinator storing the information. After collecting the previous data, it executes Equation 11. If the process determines $\gamma<\Phi_e$ then the information of the new node is stored in the coordinator, else, the distributed coordinators starts selecting a coordinator to execute the Equation 3 to 10.

3.2 Scalable Multi-ring based on Fuzzy System

Each coordinator uses the fuzzy system to assign the groupings of peers. After the group assignment, each peer is processed to find the nearest neighbor peers using the minimized route function in constructing the multi-ring topology. The structure built from SMR is utilized by the proposed scalable search to implement the load balancing for the P2P system.

3.2.1 Minimized Route Function

The result of scalable peer grouping is having more peers in every group that also provides more links in establishing the multi-ring. To overcome long routes, a minimized route function, shown in Equation 12, is formulated to find the nearest route to a destination node in the physical network.

$$\min(r) = R = \sum_{i=1}^{k=\infty, n_x} f(n_i, n) \tag{12}$$

This function stores the possible routes in R and then chooses the nearest route. The n, which is the destination node, is compared with n_i, which is a neighbor of the source node n_x. The source node has k number of neighbors to check the destination node. After checking all neighbors, if n is not found then a sub procedure will do the same calculation in Equation 12 to each neighbor of n_x. The index of n_x is changed to n_i and n_i becomes the neighbor node of the current n_x. If n is found then it returns the route information r_x to the source node and collect to R, where $R = \{r_1, r_2, \ldots r_x\}$. The sub procedure determines the number of hops of each route in the physical network. The process continues until the destination node is searched, or it returns to the source node, or it determines an edge node. Also, if nodes are connected to a router, which is the case in our simulation, then the node information in the router are queried. In this case, the function in Equation 12 is changed to $f\,(rt_i\,(n),\, n)$ where $rt_i(n)$ is the function of a router to query the nodes in its list of nodes. After collecting R, the least hop route is chosen.

3.2.2 Scalable Multi-Ring Topology

A node sends P_x to a coordinator node by *send_reg*(P_x) interface and then the coordinator node processes the node's profile through the fuzzy system in deciding its peer group. A threshold value for membership, represented by Φ_m, controls the membership of the nodes, where $\Phi_m <$ 1. Equation 13 decides if a node belongs to a peer group n where a value of 1 indicates that a node is allowed to join peer group n. P_x is the profile of n_x which is changed to k and this is processed in the membership function $q_n(k)$ of the peer group n.

$$peerGroup_n(n_x \rightarrow P_x) = \begin{cases} 1 & if\ q_n(k) > \Phi_m, \\ 0 & otherwise. \end{cases} \tag{13}$$

Fig. 2 illustrates the classification procedure where node x requests to join the peer network by sending its profile properties indicated by *send_reg*(P_x) to the coordinator node. The coordinator node classifies P_x using the scalable peer grouping. After the grouping process, a node joins the peer group which will be processed in the multi-ring connection. The calculation of classification error is also executed at this point. We identify the inner peers as the nodes inside the ring and sub peers as the nodes that form the outer ring. After all nodes performed Equation 13, inner peers are selected based on higher bandwidth to serve numerous queries from sub peers where the same approach in [13] is used. The number of inner peers the ratio of inner peers (i) and sub peers (s) which is, $q=i/s$. Subsequently, excluding inner peers, sub peers form the outer ring. A peer node starts selecting the right peer within the peer group using the function in Equation 12. This procedure continues until the last peer connects to the first peer that completes the ring connection.

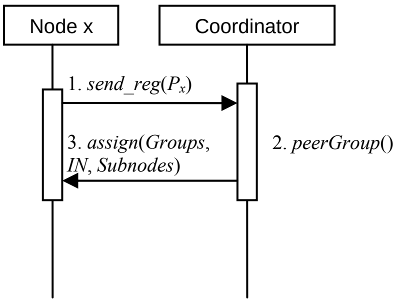


Fig. 2. Classification method of a coordinator node in assigning a node to peer groups.

If a selected node is an inner peer then the procedure skips it and selects the next nearest node to connect. Then, sub peers for an inner peer are determined by dividing the ring into i where first peer (fn_i) and last peer (ln_i) of a division disconnect on its left peer and right peer, respectively, and then fn_i and ln_i connect to the assigned inner peer. The assigned inner peer in a sub group becomes a peer leader and obtains the routing and resource information of its sub peers. A peer leader is a node that contains the file information from its sub peers and uses this information in file queries of its sub peers and inner peers.

```
for each $g_k \in$ GROUP do
 for each $in_i \in$ INNERPEER$_k$ do
  for each $in_j \in$ INNERPEER$_k$ where $in_j$ is not $in_i$ do
   if $leftPeer(in_j)$ == FALSE and $hopDist(in_i, in_j) < hopDist(in_i, np)$ then
    $np = in_j$
   end if
  end for
 $assignRight(in_i, np)$ and $assignLeft(np, in_i)$
 end for
end for
```

Fig. 3. Algorithm for assigning the left and right peers to construct the ring of inner peers.

Similar to the procedure of building the outer ring, each inner peer selects a right inner peer until it reaches the last inner peer shown in **Fig 3**. Because of the scalable grouping, an inner peer can be a peer leader more than once or it can be an inner peer to several peer groups which creates a bottleneck to queries of sub peers. In our method, a node that is already assigned as an inner peer in other peer group is skipped and another candidate node will be chosen. Also, a new node that will connect the multi-ring after determining its peer group starts finding the nearest node. The chosen nearest node disconnects to its left peer and connects to the new node, and also the left peer connects to the new node.

All inner peers have a routing list of its sub peers and neighboring inner peers, but sub peers only have links, right and left peers, to forward the query. A query from a sub peer is sent in both directions to its peers and this will continue until it reaches the inner peer. The messages are efficiently relayed within the peers because of the minimized route function. After reaching the inner peer, if a query was not successful on finding the resource in the sub peers then the inner peer forwards the query to its neighboring inner peers. Queries are directed to inner peers and this provides fast response in sending the query compare to a ring topology.

3.2.3 Scalable Search with Load Balancing

The clustered peers are assumed to contain same resources or files. Similar to the multi-ring topology [19], the query flow of a peer is from left and right until it reaches its inner peer (in_x). In our approach, a message query from a peer source (p_x), a sub peer of in_x, contains information of $m\{n, l, g, p_x, in_x\}$ where n is for node index, l is for load, and g is for the group which is used to append information in each peer that satisfied the query.

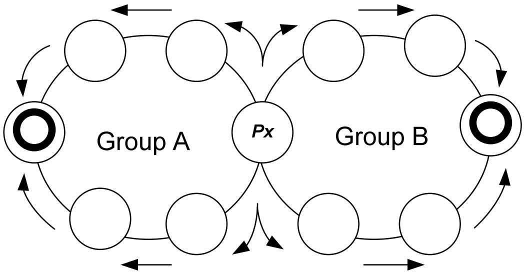


Fig. 4. A peer is classified in two groups (A and B) and performs the scalable search.

Fig. 4 illustrates the procedure of scalable search where p_x which has been classified into two groups (A and B) sends the same query to those groups. Peers of p_x only relay the query to the next peer inside the sub peer group after receiving m. Every peer that satisfied the query adds its information to m. When m reaches in_x, in_x immediately sends m back to p_x and then p_x decides the appropriate node using Equation 14. However, if resource was not found within the sub peers, then the inner peer will query its neighboring inner peers using m. Inner peers also do the same sequence of finding the resource inside the inner ring using their resource list information. On processing the query, if an inner peer finds the nodes with the resource then it requests those nodes to send load information. After receiving, the inner peer appends the information to m and m is returned to in_x.

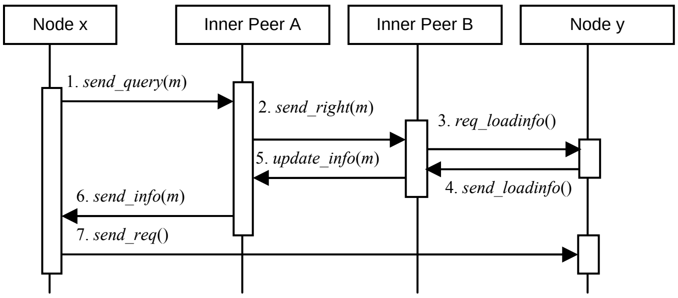


Fig. 5. A query procedure of a node to its inner peer where the resource is found in other inner peer.

Fig. 5 illustrates the query within the inner peers. A node x requests a resource and it is found in node y, but they were assigned to different inner peers. If a query reaches the inner peer A and the resource was not found in its sub peers then the inner peer A sends the query to its neighboring inner peers. After finding the resource on one of its inner peers which is the inner peer B, it returns m with the address of the node and load information to the inner peer A.

Then, the inner peer *A* sends the *m* with the appended information to node *x*. Finally, the node *x* requests node *y* for a resource needed by node *x*.

The distributed characteristic of a P2P system makes load balancing difficult to implement. A load balancing technique tries to find more nodes with the right resource and chooses the least loaded among those nodes. Eventually, gathering such load information in a ring topology will result to high network latency and query delays. However, without load balancing, a peer could suffer on task overloading which apparently affects the stability of a node's system that also causes delays on serving peers. On top of that, the nodes in a P2P system can have large load variations throughout the network which means resources are not used efficiently. Other researchers consider random selection [29] and optimization [30] techniques in P2P but these are rather difficult to implement. The SMR adopts the least load selection for the load balancing. After p_x receives *m*, it uses the load information from nodes represented in Equation 14. The nodes having the queried resource are collected in *S*. It is possible that some nodes collected in *m* are duplicated when performing the scalable search. This is handled by skipping the same node in collecting the node information. In Equation 14, σ_i indicates a load of node *i* in *S* which is used to compare the loads.

$$CandidatePeer = getindex(\min\{\sigma_1, \sigma_2, ..., \sigma_i\}) \tag{14}$$

The minimum σ_i is selected from $S = \{\sigma_1, \sigma_2, \ldots, \sigma_i\}$ and its node identification is used to route the node to access the resource. The access procedure also uses the minimized route function when using a resource or downloading a file.

4. Experimental Evaluation

4.1 Simulation Environment

The simulation environment used 10,000 nodes for the peer network and 1,000 unique files to process the scalable peer grouping. In the physical network, nodes were divided equally into 100 domains where each domain serves 100 nodes. In a domain, nodes were connected to a router with a latency of 10 milliseconds (ms) to send a message. Routers were connected in a ring topology having two router neighbors in each router with a latency of 100 ms in each connection. The 1,000 files were divided into 4 classes; music, movies, documents and installers where each class represents the 250 files. These files were replicated throughout the nodes and a node has 10 files which were randomly assigned. Nodes used the files for their interest profile where each profile value was determined by counting all files with a class associated to that profile. We assigned a coordinator in every domain to collect the profile values and to handle the peer grouping after constructing the fuzzy system. After grouping, peers in a group formed the multi-ring where *q* was set to 0.10 or 1 inner peer per 10 sub peers which is similar to the ratio in [13]. The peer neighbor connections in SMR can consist of several hop links in the physical network. If a neighboring peer is at the other domain which has 1 physical hop link using routers then the message is routed by 3 hop links from the source node to the destination node where the latency is calculated by the total latencies among the routers (1*hop* x 100*ms*) and the total latencies of the source and destination nodes to its router (2 x 10*ms*) which is totaled to 120*ms*. We didn't consider the delay time produced by the network congestion from simultaneous sending of messages and it was assumed to be 0. The flow of a query starts from right and left peers, and the query continues until its maximum hops. Nodes that were visited by the query message were listed in the node information list. In forwarding the message, a query message is copied and multiplied based on the number of linked peers and it does not forward to a peer that was already in the node information list. After it reaches its maximum hops, it returns to the peer that is the source of query.

4.2 Performance Measure

k-means is a knowledge-based clustering which uses the Euclidean distance to discriminate data and divide the data into *k* groups. We processed the vector values of interest profiles represented by *x* to the *k*-means and *x* were compared to each cluster center c_i represented by $d_i = | x - c_i |$. The class with smallest d_i is the chosen group for the peer with *x*. In SMR, because of the scalable grouping, peers can be also classified in other groups while, in *k*-means, peers are only classified in one group. Normally, a cross-validation is performed to test the accuracy of a classifier. In our experiment, a classification method was evaluated by accuracy of classifying new patterns which simulates the accuracy of assigning the group of a new node. The connectivity rate is determined by connections of a node to all nodes within the network which is also considered for the scalability of searching resources. As discussed previously, we assumed that clustering provides a high probability to answer the queries within a group. However, increasing the number of groups will lower the connectivity rate of nodes. Equation 15 shows the calculation of connectivity rate where *K* is the number of nodes and R_k is a routing function. This evaluates the successful connection of each node (n_i) to all nodes (n_j) where a node tries to route all nodes using the P2P topology.

$$ConnectivityRate_k = \sum_{i=1}^{K} \sum_{j=1}^{K} R_k(n_i, n_j) \qquad (15)$$

In performing file queries, message overheads were observed to measure the efficiency of a P2P topology. The message hops and latencies of a query message were counted. Also, we are interested on the number of file replica found by a query and number of messages distributed throughout the network. In [18] defines a measure of accuracy in disseminating a subscription message. A false positive is determined by a peer received a message but was not interested of that message and a false negative is determined by a peer matches the message subscription but was missed by the message subscription. In our experiment, we are interested on the number of replica found by the query process; therefore, we changed the variables of the mentioned accuracy measures. The false positive is the ratio of the file replica count to the number of nodes visited by a message while the false negative is the ratio of the number of nodes with the replica which were not queried to the number of all nodes with the replica.

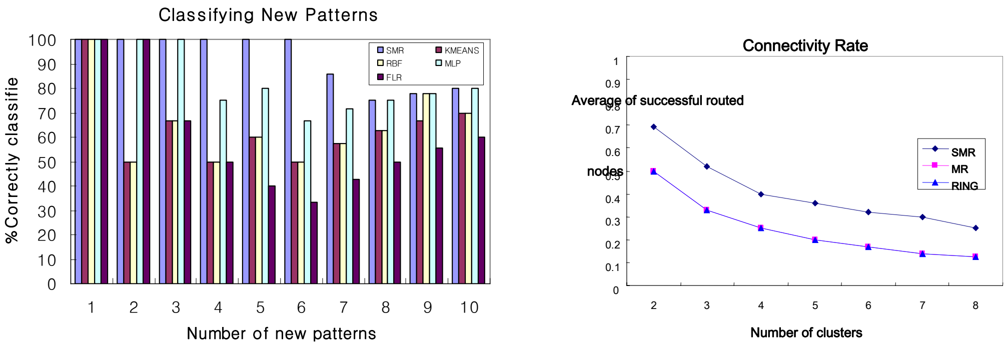
4.3 Experiment Result

We simulated the P2P environment using the physical network topology in Section 4.1. A proper configuration of SMR is needed to overcome unnecessary overheads. We determined that the smallest compactness of the objective function (*J*) is having four (4) clusters based on the data. In **Table 1**, the peer count of SMR and *k*-means are shown where SMR uses different threshold values of group membership. Using Φ_m=0.1, there were a total of 4,765 additional peers. Setting a smaller threshold value will increase the number of peers in a group like shown in Φ_m=0.01. This does not affect the accuracy of classification because the parameters of the fuzzy system were intact. However, having more peers in a group will also increase the message overheads in performing queries. Therefore, setting an appropriate value for Φ_m is necessary to consider. We chose Φ_m=0.1 to construct the SMR because it produces a lesser overheads compare to Φ_m=0.01, Φ_m=0.025 and Φ_m=0.05. To compare the classification accuracy performance of SMR, we used tools like NEFCLASS-J [31] for the fuzzy system of SMR and Weka 3.5 [32] for other algorithms such as *k*-means, radial basis function (RBF) network, multilayer perception (MLP) and fuzzy lattice reasoning (FLR).

Table 1. Peer count of SMR using different membership threshold values and *k*-means.

Groups	**SMR based on Fuzzy Clustering**				***k*-means**
	Φ_m=0.01	Φ_m=0.025	Φ_m=0.05	Φ_m=0.1	
Class 1	2257	1821	1764	1764	1957
Class 2	6717	6010	5620	5610	3208
Class 3	6376	5886	5644	5627	2733
Class 4	2024	1808	1764	1764	2102
Total	17,374	15,525	14,792	14,765	10,000

The structure built from data clustering of *k*-means was used in classification. We compared the learning ability of RBF and MLP to the learning algorithm of SMR. The FLR was used to compare its fuzzy system to SMR in classification. The fuzzy system of SMR was used for the classification test which contained 4 fuzzy sets while *k*-means and RBF used 4 clusters, and MLP was consisted of 4 hidden layers. We provided the same training time for SMR, RBF and MLP where we set 500 epochs.



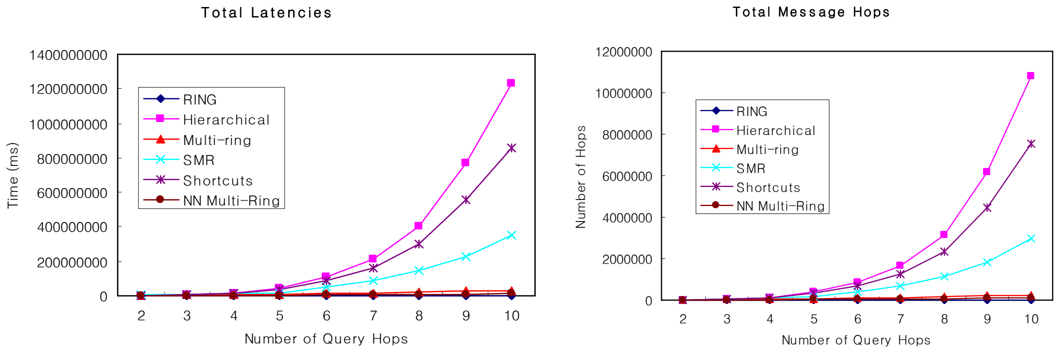
(a) **(b)**

Fig. 6. **(a)** Classification accuracy based on introducing new patterns on SMR, *k*-means, RBF, MLP and FLR and **(b)** connectivity rate of SMR, MR and RING based on the number of clusters.

New interest profile patterns were processed by each algorithm and we determined the percentage of correctly classified patterns. **Fig. 6-(a)** shows the result of classification accuracy using the cases of increasing the number of new patterns. SMR has the highest accuracy among the algorithms in classification test where all patterns were classified correctly until 6 patterns. MLP was second to SMR while *k*-means and RBF showed nearly the same accuracy performance, and FLR has the lowest accuracy. The average accuracy of SMR, *k*-means, RBF, MLP, and FLR were 91.8%, 63.3%, 64.4%, 82.59% and 59.94%, respectively, where SMR outperformed all methods in this test. We compared the connectivity rate of SMR to the clustered multi-ring (MR) and ring (RING) topologies. Equation 15 was used to calculate the connectivity rate of each method based on the number of clusters and the result is shown in **Fig. 6-(b)**. The RING and MR have the same results of connectivity rate. SMR has an average of 16% better connectivity rate compare to other methods because some peers have connection to other groups. The additional connections of peers are considered as relevant connections because these were processed in the knowledge-based clustering.

After the classification accuracy and connectivity tests, we simulated the proposed scalable search using the physical network configuration in Section 4.1. We compared the semantic-aware (ring, hierarchical, multi-ring) and semantic-aware with topology-aware

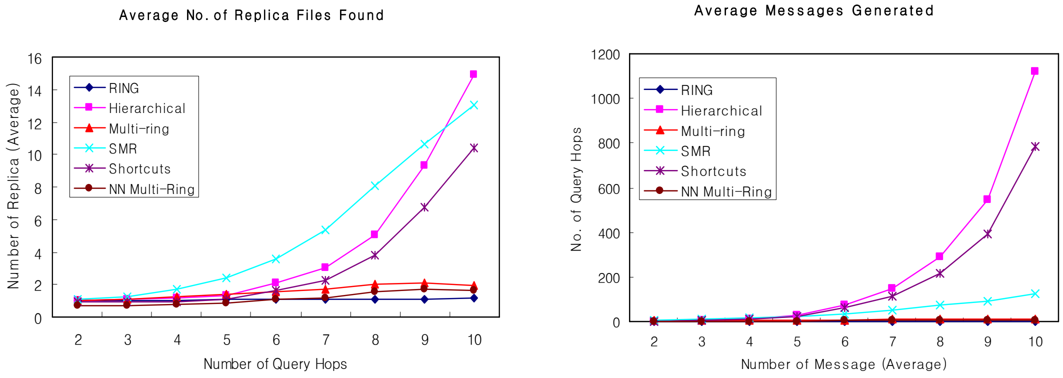
(SMR, shortcuts, NN multi-ring) techniques. In the ring, we used the sorted *k* values in Section 3.1 and connected all peers where p_i has two logical links; p_{i-1} and p_{i+1}. In the case of p_0, p_1 and the last peer (p_m) were its logical neighbors, and in p_m, p_0 and p_{m-1} were its logical neighbors. The hierarchical approach used the hierarchical agglomerative algorithm to construct the peer network. In the first level, the algorithm starts pairing peers with the nearest distance from their *k* values. Successively, each pair merges and establishes its logical link. On the next level, a merged peer set finds another merged peer set with the nearest *k* values where peers from the two merged peer sets are compared and peers with nearest *k* values are selected. After the pair of merged peer sets is determined, peers from the two sets are merged. The logical links are only established to the selected peers. This will continue to the next level until there will be only two pairs of merged peer sets left to establish their logical links. In the multi-ring, we used the *k*-means to group the peers and processed the multi-ring based on nearest *k* values of peers which was similar to the peer ring. Also, we considered a topology-aware technique in multi-ring (NN multi-ring) by connecting the nearest physical distanced nodes instead of connecting using the nearest *k* values. In the interest-based shortcuts, each node is mainly connected to 3 nodes for the unstructured overlay. Initial shortcuts were determined by nearest *k* values and provided each peer with 5 shortcuts. The query in all shortcuts is counted as one query hop. We generated 100 file requests where the source node of the requests and files queried were randomly selected. A message query is forwarded to peers until it reaches its maximum hops. The following are determined; 1) message overheads by the total physical hops and latencies, 2) number of replicas found associated with the volume of generated messages, and 3) accuracy of finding replicas by false positive and false negative measures.



(a) **(b)**

Fig. 7. **(a)** Message overheads by total latencies and **(b)** message hops generated from queries.

In **Fig. 7-(a)**, the latency of SMR is higher than NN multi-ring, multi-ring and ring because peers can have more than two links to forward the query message. Also, it was observed that SMR, shortcuts and hierarchical have a very large value of latency and message hops because the messages were multiplied in forwarding the query to the neighboring peers. The result from **Fig. 7-(a)** only adds all latencies generated by 100 queries but it does not represent the response time of a query. In **Fig. 7-(b)**, the message hop performance of each approach has a similar trend in **Fig. 7-(a)**. In average, SMR was 3 times better in disseminating messages than the hierarchical approach. In **Fig 8-(a)**, SMR has the highest replica count found until 9 maximum hops of a query. In **Fig. 8-(b)**, SMR was fourth to the lowest volume of message generated but better of almost 6 times than the hierarchical. The ring was constant in generating two messages and the multi-ring has an average of 8, and when comparing their numbers of replica found, the multi-ring has 0.80 higher than the ring.

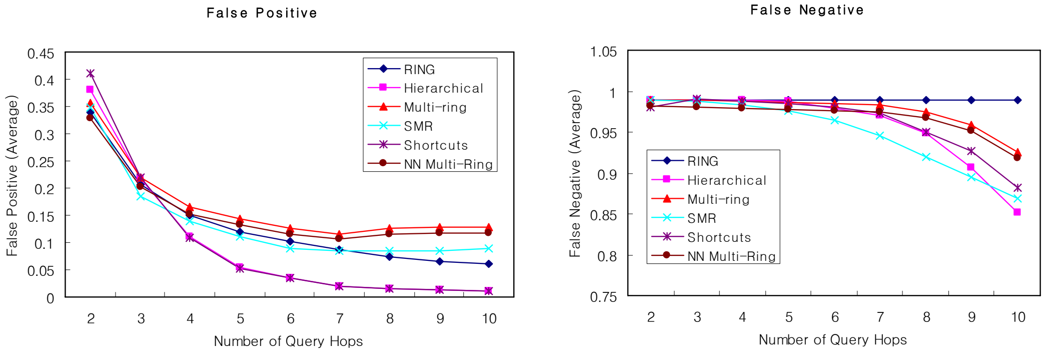


(a) (b)

Fig. 8. **(a)** Average replica file found and **(b)** average message generated from queries.

Even though multi-ring was higher of almost twice, it generated four times the number of messages in finding file replicas compare to the ring. Also, the multi-ring has a tendency that a query will find more replicas, but as the message hop increases, the search will be limited to find more replicas because of the group separation which was observed from 8 to 10 maximum hops in **Fig. 8-(a)**.

NN multi-ring has low message overhead but also has less number of replica found than the multi-ring. The hierarchical found more replicas in 10 hops but generated high message overheads which was less efficient as shown in **Fig. 8-(b)**. The interest-based shortcuts approach has same trend with hierarchical, but because of the shortcuts, the message overhead is low. It was also observed the exponential increase of the hierarchical approach in **Fig. 8** and **Fig. 9** were similar. In average, SMR has still more replicas found with 5.24, while hierarchical, shortcuts, multi-ring, NN multi-ring and ring have averages of 4.32, 3.19, 1.94, 1.13 and, 1.14, respectively.



(a) (b)

Fig. 9. **(a)** Average false positive and **(b)** false negative based on number of replicas found.

The accuracy of the search was shown in **Fig. 9-(a)** and **Fig. 9-(b)**. A higher false positive value means the search has better retrieval of file replicas while a lower false negative value means the search is more scalable in finding the file replicas throughout the peer network. We observed in **Fig. 9-(a)** that from 2 to 3 hops, the shortcuts approach was better; however, from 4 to 10 hops, the accuracy was getting lowered. Multi-ring was more accurate because of the relevant peers connected to a group. SMR, shortcuts and hierarchical approaches are more scalable in searching replicas where SMR can find replica to other peer groups while hierarchical and shortcuts are fully connected network. **Fig. 9-(b)** shows the ratio of queries that missed the nodes with the file replica where the SMR was better than the ring, multi-ring and NN multi-ring because of the scalability of search. In 10 hops, the hierarchical was better but the previous graphs show it was less efficient in handling message overheads. In average, SMR was still better in the false negative result by having a value 0.947 compare to hierarchical, shortcuts, ring, multi-ring and NN multi-ring having 0.956, 0.9617, 0.989, 0.975 and 0.968, respectively.

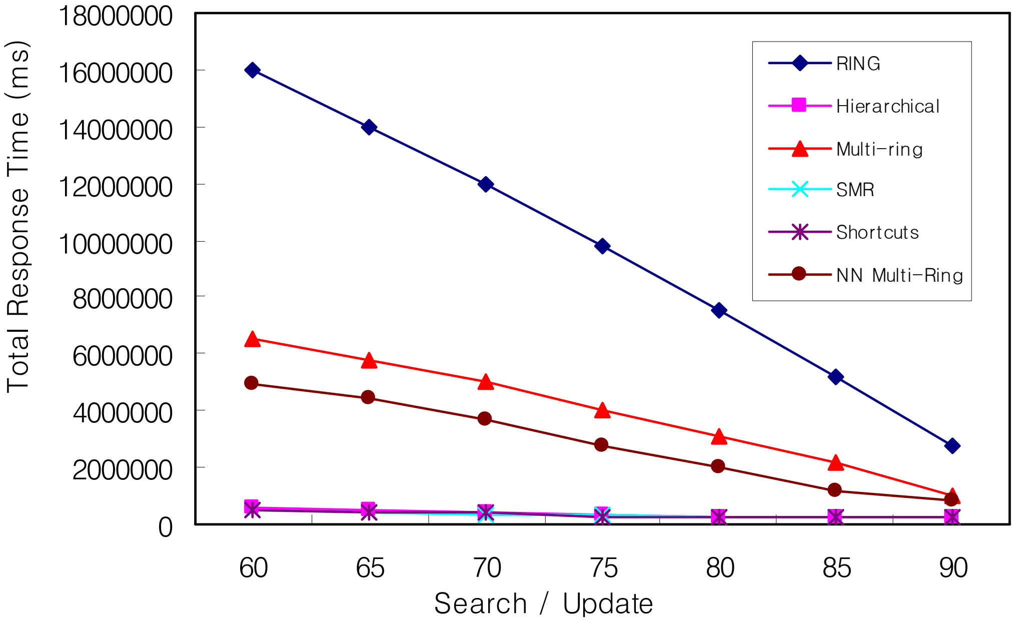


Fig. 10. Total response time of queries considering the search and update rates.

The maintenance of SMR was taken into account by considering the search and update rates in queries. The turn-around time of a search query was determined. In an update query, the turn-around time of updating the file and all of its replicas throughout the peer network was obtained. The number of queries for the simulation was a ratio of the number of search and update messages. There were 100 query messages generated which contained search and update queries. The simulation started from 60% which means the query messages were consisted of 60 search queries and 40 update queries. **Fig. 10** shows the results from query performance considering search and update rates where the total response time was obtained. It was observed that the ring, multi-ring and NN multi-ring topologies provided a very high latency in updating the replicas within the peer network where the latency in search was only a small portion from the total latency. The SMR, shortcuts and hierarchical topologies were more efficient in handling the updates which showed lower latency than the three mentioned algorithms. In the case of SMR, inner nodes were used as distributed file repositories for searching and updating files. Hierarchical and shortcuts were faster because of the hierarchical strategy but it generated a high volume of messages which was not efficient. From 60% until 75%, the hierarchical was the fastest; however, the SMR was better than the hierarchical starting from 80%.

The results from **Table 1** and **Fig. 6** show the scalability and accuracy of the SMR in having high connectivity rate of peers in changing the number of clusters and accurate in classifying new patterns, respectively. The graph results from **Fig. 7** and **Fig. 8** show that the SMR finds more file replicas compare to shortcuts, ring and multi-ring, NN multi-ring, and SMR is more efficient in handling message overheads than the hierarchical and shortcuts approaches. **Fig. 9** shows the accuracy and scalability of the queries in finding file replicas where SMR has better accuracy than the hierarchical, shortcuts and ring approaches, and more scalable compare to all other algorithms. In **Fig. 10**, the SMR showed a good performance in updating file replicas.

5. Conclusion

This paper showed the architecture of the scalable multi-ring (SMR) for accurate and scalable peer grouping of an interest-based P2P system. The SMR uses the fuzzy system to perform a peer grouping and after identifying the groups, peers in each group forms a multi-ring connection. We integrated a scalable search for the interest-based P2P networks and a minimized route function to improve the SMR. The proposed search uses the SMR to find more resources and implements the load balancing by selecting the least loaded node. In performing the search, the minimized route function improved the message routing of SMR. We configured a network topology where we used 10,000 nodes for the network simulation, and then evaluated the performance of SMR in classification accuracy of peer grouping and efficiency of the scalable search. Using 4 groups to divide the peer network, the connectivity rate was increased to 47.5% after setting the Φ_m=0.10. In classifying new patterns, SMR outperformed other algorithms with a difference of 28.5% in *k*-means, 27.4% in RBF, 9.21% in MLP and 31.86% in FLR. In performing peer queries, the proposed scalable search was more efficient in querying peers in terms of number of replicas found and accuracy of search compared to other approaches.

References

[1] Napster's website, *http://www.napster.com*.

[2] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communication Survey and Tutorial*, vol. 7, no. 2, pp. 72-93, Mar. 2004. Article (CrossRef Link).

[3] Gnutella community website, *http://gnutella.wego.com*.

[4] A. Rowstron and P. Druschel, "Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems," in *Proc. of IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pp. 329-350, Nov. 2001.

[5] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for Internet applications," *Transactions on Networking, IEEE/ACM*, vol. 11, no. 1, pp. 17-32, Feb. 2003. Article (CrossRef Link).

[6] C. Zhang, A. Krishnamurthy and R. Y. Wang, "Brushwood: distributed trees in peer-to-peer systems," *Lecture Notes in Computer Science*, vol. 3640/2005, pp. 47-57, 2005. Article (CrossRef Link).

[7] M. Khambatti1, K. D. Ryu and P. Dasgupta, "Structuring peer-to-peer networks using interest-based communities," *Lecture Notes in Computer Science*, vol. 2944/2004, pp. 48-63, 2004. Article (CrossRef Link).

[8] B. Huberman, D. M. Romero and F. Wu, "Social networks that matter: twitter under the microscope," *First Monday*, vol. 14, 2009.

[9] M. Berthold and D. J. Hand, "Intelligent data analysis: an introduction," *Springer-Verlag New York, Inc.*, pp. 1-14, 1999.

[10] R. M. Mateo, H. H. Yang and J. W. Lee, "Scalable grouping based on neuro-fuzzy clustering for P2P networks," *Lecture Notes in Computer Science*, vol. 5559/2009, pp. 813-822, 2009. Article (CrossRef Link).

[11] H. Zhuge, X. Chen, X. Sun and E. Yao, "HRing: a structured P2P overlay based on harmonic series," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 2, pp. 145-158, Feb. 2008. Article (CrossRef Link).

[12] M. Castro, P. Druschel, Y. C. Hu and A. Rowstron, "Topology-aware routing in structured peer-to-peer overlay networks," *Future Directions in Distributed Computing*, Springer-Verlag Berlin, pp. 103-107, 2003.

[13] M. O. Junginger and Y. Lee, "The multi-ring topology-high performance group communication in peer-to-peer networks," in *Proc. of Second International Conference on Peer-to-Peer Computing*, pp. 49, Sep. 2002. Article (CrossRef Link).

[14] W. T. Chen, C. H. Chao and J. L. Chiang, "An interest-based architecture for peer-to-peer network systems," in *Proc. of International Conference on Advanced Information Networking and Applications*, pp. 707-712, Apr. 2006. Article (CrossRef Link).

[15] H. Chiou, A. Su and S. Yang, "Interest-based peer selection in P2P network," in *Proc. of IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing*, pp. 549-554, Jun. 2008. Article (CrossRef Link).

[16] Q. Gao, Z. Qiu, Y. Wu, J. Tian and Y. Dai, "An interest-based P2P RDF query architecture," in *Proc. of International Conference on Semantic, Knowledge and Grid*, pp. 11, Nov. 2005. Article (CrossRef Link).

[17] Y. Busnel and A. M. Kermarrec, "PROXSEM: interest-based proximity measure to improve search efficiency in P2P systems," in *Proc. of 4th European Conference on Multiservice Networks*, pp. 62-74, Feb. 2007. Article (CrossRef Link).

[18] R. Chand and P. Felber, "Semantic peer-to-peer overlays for publish/subscribe networks," *Lecture Notes in Computer Science,* vol. 3648/2005, pp. 1194-1204, 2005. Article (CrossRef Link).

[19] J. Yang, Y. Zhong and S. Zhang, "An efficient interest-group based search mechanism in unstructured peer-to-peer network systems," in *Proc. of International Conference on Computer Networks and Mobile Computing*, pp. 247, Oct. 2003. Article (CrossRef Link).

[20] K. Sripanidkulchai, B. Maggs and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," in *Proc. of IEEE INFOCOM*, pp. 2166-2176, Jul. 2003. Article (CrossRef Link).

[21] M. Amoretti, M. Agosti and F. Zanichelli, "Interest-based overlay construction and message routing in service-oriented peer-to-peer networks," in *Proc. of IASTED International Conference on Parallel and Distributed Computing and Networks*, pp. 152-157, Feb. 2008.

[22] W. Wu, W. Hu, Y. Huang and D. Qian, "Group-based peer-to-peer network routing and searching rules," *Current Trends in High Performance Computing and Its Application*, Springer Berlin Heidelberg, pp. 509-514, 2005. Article (CrossRef Link).

[23] S. Y. Chen, W. H. Tseng and H. Mei, "A multilayer topic-group based P2P network," in *Proc. of International Conference on Advanced Information Networking and Applications*, pp. 702-706, Apr. 2006. Article (CrossRef Link).

[24] X. Bai, S. Liu, P. Zhang and R. Kantola, "ICN: interest-based clustering network," in *Proc. of Fourth International Conference on Peer-to-Peer Computing*, pp. 219-226, Aug. 2004. Article (CrossRef Link).

[25] X. M. Huang, C. Y. Chang and M. S. Chen, "PeerCluster: a cluster-based peer-to-peer system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 10, pp. 1110-1123, Oct. 2006. Article (CrossRef Link).

[26] M. R. Anderberg, *Cluster Analysis for Applications*, Academic Press, New York, 1973.

[27] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York, Plenum Press, 1981.

[28] D. Nauck and R. Kruse, "NEFCLASS - a neuro-fuzzy approach for the classification of data," in *Proc. of ACM Symposium on Applied Computing*, pp. 461-465, 1995. Article (CrossRef Link).

[29] J. S. A. Bridgewater, P. Oscar Boykin and V. P. Roychowdry, "Balanced overlay networks (BON): an overlay technology for decentralized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 8, pp. 1122-1133, Aug. 2007. Article (CrossRef Link).

[30] C. Chen and K. C. Tsai, "The server reassignment problem for load balancing in structured P2P systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 2, pp. 1122-1133, Jan. 2008. Article (CrossRef Link).

[31] NEFCLASS, website *http://fuzzy.cs.uni-magdeburg.de/nefclass*.

[32] J. Han and M. Kamber, "Data Mining Concepts and Techniques", 2nd Edition, Morgan Kaufman, pp. 1-38, 2006.

Romeo Mark A. Mateo received his B.S. degree in Information Technology from West Visayas State University, Philippines in 2004 and M.S. degree in Information and Telecommunications Engineering from Kunsan National University, South Korea in 2007. Currently, he is a Ph.D. candidate in Information and Telecommunications Engineering and working as a research assistant at the Distributed Systems Laboratory (DSL). His research interests include distributed systems, mobile computing, wireless sensors, artificial intelligence and data mining.

Jaewan Lee received his B.S., M.S., and Ph.D. degrees in Computer Engineering from Chung-Ang University in 1984, 1987, and 1992, respectively. Currently, he is a professor at the School of Electronic and Information Engineering in Kunsan National University, Kunsan City, South Korea. His research interests include distributed systems, database systems, data mining and computer networks.