비구조화 P2P 시스템에서 이동에이전트를 이용한 Peer의 속성기반 계층적 클러스터링☆

Property-based Hierarchical Clustering of Peers using Mobile Agent for Unstructured P2P Systems

마이클 안젤로 살보* 마테오 로미오** 이 재 완***
Michael Angelo G. Salvo Romeo Mark A. Mateo Jaewan Lee

요 약

비구조화 P2P 시스템은 오늘날 인터넷에서 가장 널리 사용되지만, 파일의 배치는 임의로 이루어지며, Peer와 컨텐츠 간에는 어떤 상관관계도 존재하지 않는다. 또한 보낸 모든 질의가 원하는 데이터를 찾았는지에 대한 보장도 없다. 본 논문에서는 비구조화된 P2P시스템에서 군집형 계층 클러스터링을 사용하여 노드들을 클러스터화함으로써 검색을 향상시키는 방법을 제시한다. 제안한 기법과 k-means를 사용한 기법 간에 노드 클러스터링을 위한 지연시간을 비교하였다. 또한 제안한 알고리즘, k-means 클러스터링, 클러스터링을 사용하지 않은 방법간에 한 네트워크 토폴로지에서 데이터를 찾기 위한 지연시간에 대해 시뮬레이션을 수행하였다. 시뮬레이션 결과 제안한 기법의 지연시간이 다른 방법들보다 짧았음을 알 수 있었다

Abstract

Unstructured peer-to-peer systems are most commonly used in today's internet. But file placement is random in these systems and no correlation exists between peers and their contents. There is no guarantee that flooding queries will find the desired data. In this paper, we propose to cluster nodes in unstructured P2P systems using the agglomerative hierarchical clustering algorithm to improve the search method. We compared the delay time of clustering the nodes between our proposed algorithm and the k-means clustering algorithm. We also simulated the delay time of locating data in a network topology and recorded the overhead of the system using our proposed algorithm, k-means clustering, and without clustering. Simulation results show that the delay time of our proposed algorithm is shorter compared to other methods and resource overhead is also reduced.

☞ keyword: hierarchical clustering, mobile agent, P2P system, 계층적 클러스터링, 이동에이전트, p2p 시스템

1. Introduction

With the rise of the widely popular peer-to-peer file sharing applications such as BearShare [1], and KaZaA [2], users can now download and share important files and huge amounts of data. Peer-to-peer architecture is a type of network in which each workstation has equivalent capabilities and responsibilities. This differs from client/server architectures where some computers are dedicated to serving the others. Peer-to-peer networks are generally simpler but they usually do not offer the same performance under heavy loads. The P2P network itself relies on computing power at the ends of a connection rather than from within the network itself [3]. Generally, P2P networks are used for

* 준 회 원 : 군산대학교 전자정보공학부 석사과정
masalvo@kunsan.ac.kr
** 정 회 원 : 군산대학교 전자정보공학부 박사과정
rmmateo@kunsan.ac.kr
*** 종신회원 : 군산대학교 전자정보공학부 교수
jwlee@kunsan.ac.kr(교신저자)

[2008/10/24 투고 - 2008/11/06 심사(2009/01/30 2차) - 2009/02/27 심사완료]

☆ This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation

sharing files, but a P2P network can also mean Grid Computing or instant messaging. Instant messaging (IM) is one very common form of P2P networking where software applications, such as Windows Live Messenger or Yahoo Messenger, allow users to chat via text messages in real-time. While most vendors offer a free version of their IM software others have begun to focus on enterprise versions of IM software as businesses and corporations have moved towards implementing IM as a standard communications tool for business. Currently, IMs make use of the Voice-over-IP (VoIP) service and there are also studies related to this service [4] which focuses mainly on multiparty VoIP system.

There are methods of distributing large amounts of data widely without the original distributor incurring the entire costs of hardware, hosting, and bandwidth resources. When data is distributed using this protocol, each recipient supplies pieces of the data to newer recipients, reducing the cost and burden on any given individual source, providing redundancy against system problems, and reducing dependence on the original distributor. One of these examples is BitTorrent [5], a peer-to-peer file sharing (P2P) communications protocol. Though this application is already widely used, the common problems that most users experience are in the speed and lack of peers that provide a specific service.

An important goal in peer-to-peer networks is that all clients provide resources, including bandwidth, storage space, and computing power [6]. Thus, as nodes arrive and demand on the system increases, the total capacity of the system also increases. There are mainly three different structures for P2P systems; centralized, decentralized structured, and decentralized unstructured [7]. This study mainly focuses on the decentralized unstructured P2P systems which are most commonly used in today's internet.

Hierarchical clustering algorithms organize data into a hierarchical structure according to the proximity matrix. They are mainly classified as agglomerative methods and divisive methods. Agglomerative clustering starts with N clusters and each of them includes exactly one object. A series of merge operations are then carried out that finally lead all objects to the same group. Divisive clustering proceeds in an opposite way. It starts with the entire data set belonging to a cluster and a procedure successively divides it until all clusters are singleton clusters. In this paper, we used the former, the agglomerative clustering method. We propose a P2P overlay network where nodes are clustered based on their properties. These properties are from the values of several services a peer provides. The values are decomposed into a single value thus becoming the property of a peer.

2. Related Works and Background

2.1 P2P Overlay Network

A peer-to-peer (P2P) network uses diverse connectivity between participants (or peers) in a network and the cumulative bandwidth of network participants rather than conventional centralized resources where a relatively low number of servers provide the core value to a service or application. P2P networks are typically used for connecting nodes via largely ad hoc connections [8]. Regardless of the current location of the node, P2P networks can be connected via an overlay network. There are several studies which makes use of the overlay architecture. In [4] a multiparty VoIP system adopts an overlay structure mainly for the mixing and distribution trees which is constructed on top of the overlay rather than doing it directly. Pippon connects peers via overlay but is aware of the physical infrastructure as proposed in [9]. Overlay architecture was also utilized in [10] to enhance internet quality of service. And finally overlay was also used as a means of preventing denial of service attacks on web servers in [11].

An overlay network is a computer network which is built on top of another network. Peer-to-peer networks are overlay networks because they run on top of the Internet [12]. A P2P system connects peer hosts into an overlay mesh on top of the existing IP network. Each peer is connected with a number of peers via application-level virtual links called overlay links.

2.2 Cluster Analysis

Cluster analysis is used as a tool for knowledge discovery. It may show structure and associations in data, although not previously evident, but are sensible and useful once discovered. The results of cluster analysis may contribute to the definition of a formal classification scheme, such as in taxonomy for related animals, insects or plants; suggest statistical models to describe populations; indicate rules for assigning new cases to classes for identification and diagnostic purposes; provide measures of definition, size and change in what previously were only broad concepts; or find patterns to represent classes [13]. The objective function depends on the distances between vectors uk and cluster centers cm, and when the Euclidean distance is chosen as a distance function, the expression for the objective function will be as shown in Equation 1.

$$J = \sum_{k=0}^{K} \sum_{m=0}^{M} \left\| u_k - c_m \right\|^2 \tag{1}$$

Ji is the objective function within cluster i. The partitioned clusters are typically defined by a c × K binary characteristic matrix M, called the membership matrix, where each element mik is 1 if the kth data point uk belongs to cluster i, and 0 otherwise. The Ji is minimized by several iterations and stops if either the improvement over the previous iteration is below a certain tolerance or Ji is below a certain threshold value. A multi-relational hierarchical clustering algorithm was implemented in [14] using similar principles in this study.

2.3 Nearest Neighbor Algorithm

The k-nearest neighbor algorithm (k-NN) is a method for classifying objects based on closest training samples in the feature space mostly used in pattern recognition. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The calculation of the k-NN is based on the Euclidean distance which calculates the absolute value of the two vectors. Distances from the new vector to all stored vectors are computed and k closest samples are selected. Equation 2 represents the formula for the Euclidean Distance which is commonly used in k-NN.

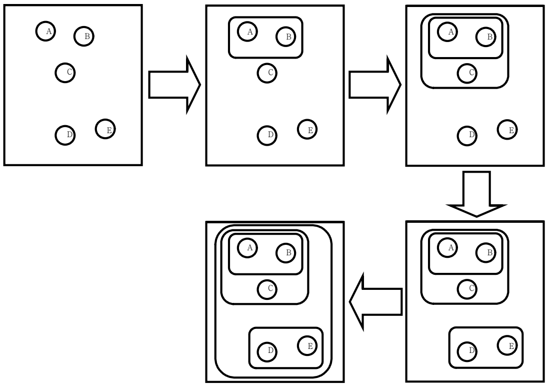
$$EuclideanDist = \sum_{n}^{N} \left\| x - y \right\| \tag{2}$$

There are a number of ways to classify the new vector to a particular class. One of the most used techniques is to predict the new vector to the most common class amongst the k-nearest neighbors. A major drawback to using this technique to classify a new vector to a class is that the classes with the more frequent examples tend to dominate the prediction of the new vector, as they tend to come up in the k-nearest neighbors when the neighbors are computed due to their large number. One of the ways to overcome this problem is to take into account the distance of each k-nearest neighbor with the new vector that is to be classified and predict the class of the new vector based on these distances. In this study, our algorithm used the minimum distance, dmin (Ci , Cj) to measure the distance between clusters hence may also be called the nearest-neighbor clustering algorithm. The k-nearest neighbor clustering algorithm was used in [15] to cluster applications with noise which represents outlier data.

3. Property-based Hierarchical Clustering of Peers

The P2P overlay network consists of all the participating peers as network nodes. There are links between any two nodes that know each other: i.e. if a participating peer knows the location of another peer in the P2P network, then there is a directed edge from the former node to the latter in the overlay network. Based on how the nodes in the overlay network are linked to each other, we can classify the P2P networks as unstructured or structured.

An unstructured P2P network is formed when the overlay links are established arbitrarily. Such networks can be easily constructed as a new peer that wants to join the network can copy existing links of another node and then form its own links over time. In an unstructured P2P network, if a peer wants to find a desired data in the network, the query has to be flooded through the network to find as many peers as possible that share the data. The main disadvantage with such networks is that the queries may not always be resolved. Popular content is likely to be available at several peers and any peer searching for it is likely to find the same thing. But if a peer is looking for rare data shared by only a few other peers, then it is highly unlikely that search will be successful. Since there is no correlation between a peer and the content managed by it, there is no guarantee that flooding will find a peer that has the desired data. Flooding also causes a high amount of signaling traffic in the network and hence such networks typically have very poor search efficiency.



(Figure 1) Agglomerative Hierarchical Clustering

In order to improve search, we approach the P2P overlay network using the agglomerative hierarchical clustering method. This clustering is a bottom-up strategy that starts by placing peers as N distinct clusters. It then merges similar peers to form a cluster and merges other clusters into larger and larger clusters until all of the objects are in a single cluster or until certain termination conditions are satisfied. These conditions are not discussed further in this paper but can be a course of future study. The main contribution of this paper is mainly the clustering of nodes in an unstructured P2P environment.

In Figure 1, Node A and B are clustered based on their values. Next, node C is clustered with cluster AB. Then, nodes D and E are clustered. The last phase is where both clusters ABC and DE are grouped altogether. In this study, as already mentioned, we propose an architecture that clusters peers according to their properties. Properties are derived from each service a peer can provide. These services are given numerical values and are translated into vectors using the Eigen decomposition [15] in Equation 3.

$$eM = PDP^{-1} \tag{3}$$

Let P be a matrix of eigenvectors in a given square matrix eM and D be a diagonal matrix with the corresponding eigenvalues. As long as eM is a square matrix, Eigen decomposition can be processed in Equation 3. After calculating the property of each peer, we begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0. All nodes are individual clusters in this level. We find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to Equation 4.

$$d[(r),(s)] = \min d[(i),(j)] \tag{4}$$

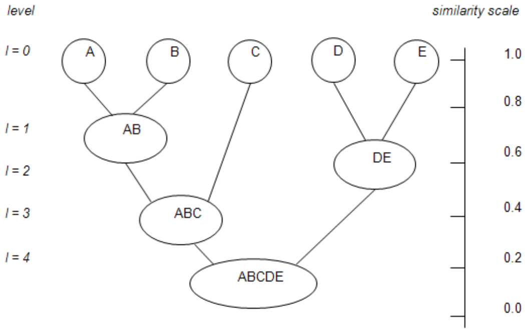
The minimum value among all pairs of clusters in the current clustering will be recorded. Next we increment the sequence number: m = m + 1. Clusters (r) and (s) will be merged into a single cluster to form the next clustering m. The level of this clustering will be set according to Equation 5.

$$L(m) = d[(r),(s)] \tag{5}$$

The proximity matrix, d, will be updated by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. This is to calculate the default values of the peers and not the current computation of the new cluster. The proximity between the new cluster, denoted (r,s) and old cluster (k) is defined in Equation 6.

$$d[(k),(r,s)] = \min d[(k),(r)], d[(k),(s)] \tag{6}$$

The process of finding the least dissimilar pairs will be repeated until all peers are in one cluster. Figure 2 shows a dendrogram representation of the clustering of 5 peers {A, B, C, D, E}.

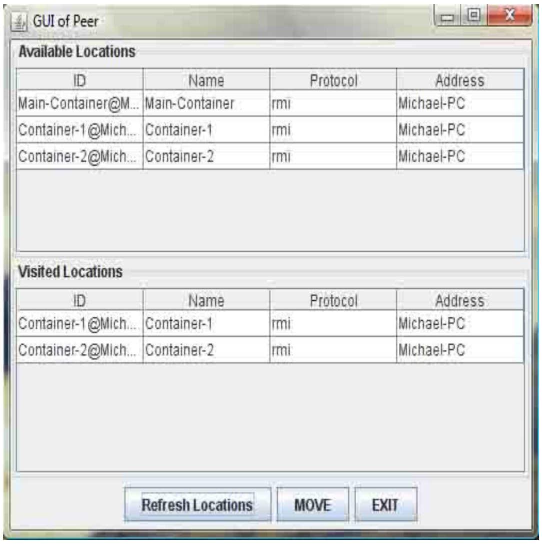


(Figure 2) Dendrogram representation for agglomerative hierarchical clustering of peers

This dendrogram shows which peer ended up in each cluster. By using this method, peers can efficiently search for services by flooding query requests starting from the lowest level in the hierarchy up to the last level until it finds its desired data.

4. Mobile Agent Implementation

In the proposed architecture, we used Jade 3.5 to implement a mobile agent to circulate within an intrasystem of 10 nodes to collect the property of each node. The OS platform used was Windows Vista. A mobile agent, named here as “Peer”, is initialized and migrated from one node to the next. This agent collected the properties of each peer connected to the system. The properties are then used to cluster the peers.



Available Locations

ID	Name	Protocol	Address
Main-Container@M...	Main-Container	rmi	Michael-PC
Container-1@Mich...	Container-1	rmi	Michael-PC
Container-2@Mich...	Container-2	rmi	Michael-PC

Visited Locations

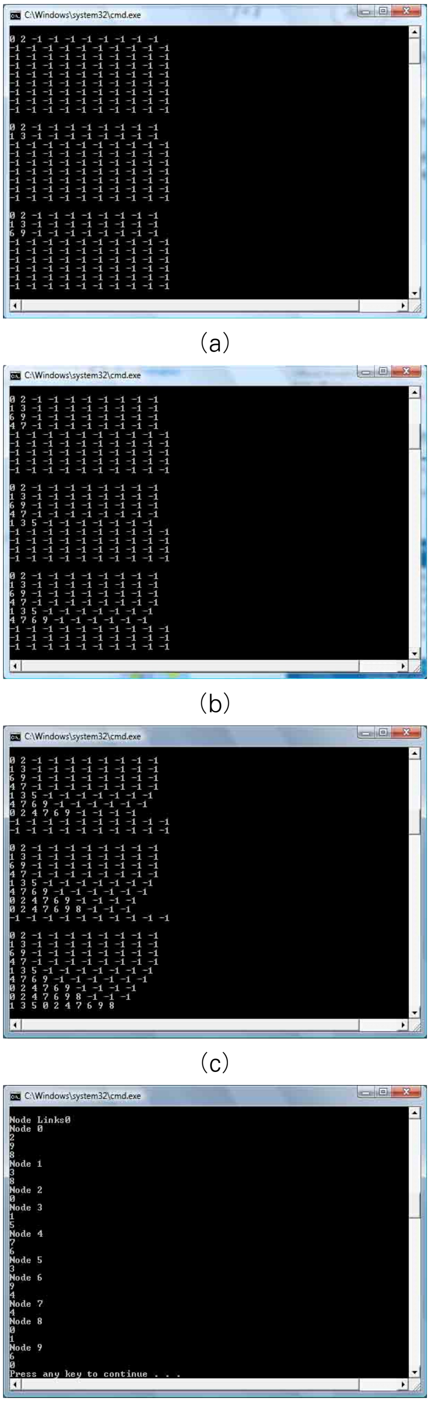
ID	Name	Protocol	Address
Container-1@Mich...	Container-1	rmi	Michael-PC
Container-2@Mich...	Container-2	rmi	Michael-PC

(Figure 3) GUI of the mobile agent

Figure 3 shows the GUI of the mobile agent. It displays the peers connected to the system in “available locations” and the containers already visited in “visited locations”. The “refresh locations” button allows newly initialized or terminated containers to be shown or removed respectively from “available locations”. The “move” button moves the agent to the selected container in “available locations”.

The proposed agglomerative hierarchical clustering algorithm was encoded using Java. After visiting all locations, the mobile agent gathered all properties of 10 peers connected to the system.

The properties of the nodes are compared to the property of each other node and the nodes that have the nearest values are clustered together. If n1 and n2 have the closest properties to each other compared to other peers, then they are clustered together.



```
0 2 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1

0 2 -1 -1 -1 -1 -1 -1 -1 -1
1 3 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1

0 2 -1 -1 -1 -1 -1 -1 -1 -1
1 3 -1 -1 -1 -1 -1 -1 -1 -1
6 9 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

(a)

```
0 2 -1 -1 -1 -1 -1 -1 -1 -1
1 3 -1 -1 -1 -1 -1 -1 -1 -1
6 9 -1 -1 -1 -1 -1 -1 -1 -1
4 7 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1

0 2 -1 -1 -1 -1 -1 -1 -1 -1
1 3 -1 -1 -1 -1 -1 -1 -1 -1
6 9 -1 -1 -1 -1 -1 -1 -1 -1
4 7 -1 -1 -1 -1 -1 -1 -1 -1
1 3 5 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1

0 2 -1 -1 -1 -1 -1 -1 -1 -1
1 3 -1 -1 -1 -1 -1 -1 -1 -1
6 9 -1 -1 -1 -1 -1 -1 -1 -1
4 7 -1 -1 -1 -1 -1 -1 -1 -1
1 3 5 -1 -1 -1 -1 -1 -1 -1
4 7 6 9 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

(b)

```
0 2 -1 -1 -1 -1 -1 -1 -1 -1
1 3 -1 -1 -1 -1 -1 -1 -1 -1
6 9 -1 -1 -1 -1 -1 -1 -1 -1
4 7 -1 -1 -1 -1 -1 -1 -1 -1
1 3 5 -1 -1 -1 -1 -1 -1 -1
4 7 6 9 -1 -1 -1 -1 -1 -1
0 2 4 7 6 9 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1

0 2 -1 -1 -1 -1 -1 -1 -1 -1
1 3 -1 -1 -1 -1 -1 -1 -1 -1
6 9 -1 -1 -1 -1 -1 -1 -1 -1
4 7 -1 -1 -1 -1 -1 -1 -1 -1
1 3 5 -1 -1 -1 -1 -1 -1 -1
4 7 6 9 -1 -1 -1 -1 -1 -1
0 2 4 7 6 9 -1 -1 -1 -1
0 2 4 7 6 9 8 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1

0 2 -1 -1 -1 -1 -1 -1 -1 -1
1 3 -1 -1 -1 -1 -1 -1 -1 -1
6 9 -1 -1 -1 -1 -1 -1 -1 -1
4 7 -1 -1 -1 -1 -1 -1 -1 -1
1 3 5 -1 -1 -1 -1 -1 -1 -1
4 7 6 9 -1 -1 -1 -1 -1 -1
0 2 4 7 6 9 -1 -1 -1 -1
0 2 4 7 6 9 8 -1 -1 -1
1 3 5 0 2 4 7 6 9 8
```

(c)

```
Node Links0
Node 0
2
9
8
Node 1
3
8
Node 2
0
Node 3
1
5
Node 4
7
6
Node 5
3
Node 6
9
4
Node 7
4
Node 8
0
1
Node 9
6
0
Press any key to continue . . .
```

(d)

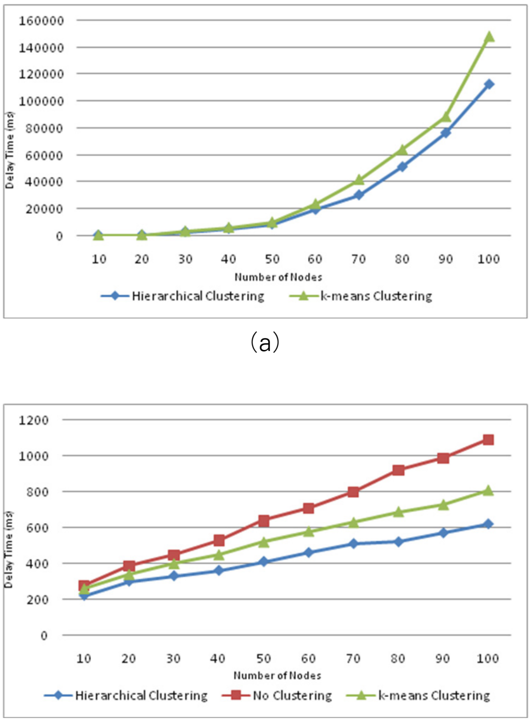
(Figure 4) The clustering (a, b, c) and connection of nodes (d)

If another node n3 has its properties closest to n2 which already belongs to a cluster with n1, n3 simply forms another cluster with nodes n1 and n2. When a node n4, which also belongs to another cluster with node n5, has its property closest to n1, the cluster with n4 and n5 joins the cluster with n1, n2, and n3 to form a final clustering with all nodes in that cluster. Figure 4 (a, b, c) shows which nodes are clustered first and gradually clustering other nodes by either including another similar node to the cluster or by combining the two clusters until all nodes have membership in one cluster. Figure 4 (d) shows which node are connected to each other node. This explains how each cluster is formed through the links of the nodes.

5. Simulation Results

To evaluate the efficiency of the proposed algorithm, we randomly generated coordinates and static connections for each node forming a network topology. We also randomly distributed property values to each of them using normal distribution. Our proposed algorithm and the k-means algorithm were then applied to cluster the nodes. We recorded the delay time of both algorithms in performing this task. We also recorded the delay time in locating randomly generated data on the network using our proposed algorithm, the non-clustered system, and the k-means clustering algorithm.

Figure 5 (a) shows the delay time in milliseconds after the algorithms cluster a number of nodes (10, 20 … 100) in the network. The proposed algorithm clustered nodes faster because the k-means algorithm attempts to find the natural center for each cluster and does so to provide the best clustering given k clusters.

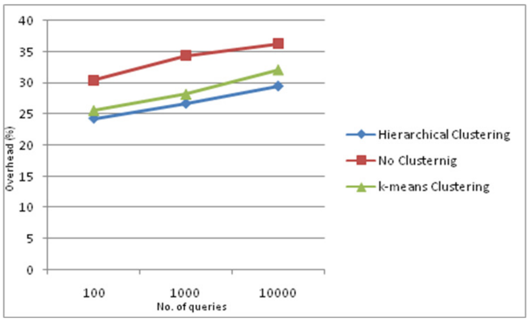


(a)

(b)

(Figure 5) The delay time (ms) for clustering nodes (a) and average delay time (ms) for locating data in a simulated query procedure (b)

Next, we simulated the query procedure by locating randomly generated data on the network with; agglomerative hierarchical clustering, no clustering, and k-means clustering. Figure 5 (b) shows the average delay time for locating data in the system with a number of generated nodes. The results show that the proposed algorithm had the advantage because nodes eventually belong to a single cluster, making search easier. Its structure made locating the node with the data faster because all nodes have connections logically while in the k-means algorithm, only nodes in the same cluster have direct connections to one another. Even though the compactness of each cluster using the k-means algorithm is also relatively high, there are no logical connections to each other cluster. This makes the algorithm do computations on each cluster thus making calculations a bit longer. On the other hand, when locating data on the system without clustering, it will start searching from the next node that is physically connected to the originating node in the network and goes on to the next, one hop at a time.



(Figure 6) Overhead of the algorithms based on number of queries

Figure 6 shows the resource overhead of the network with each of the same three instances that were previously mentioned. As implied by these results, it can be discussed that the system that was clustered using the proposed algorithm has the minimum overhead compared to the system clustered with the k-means algorithm and the system that wasn't clustered. By sending query messages of 100, 1000, and 10000 respectively through the nodes in the system to acquire the desired data, we gather that the percentage of the nodes used in the system were the fewest when using the proposed algorithm which means that resources are efficiently used by involving few nodes to carry out the task. This is because of how the nodes are clustered which provides compactness that allows the desired data to be easily located.

6. Conclusion and Future Works

This research presented a clustering method to provide efficient search for unstructured peer-to-peer systems. The agglomerative hierarchical clustering algorithm was used to cluster nodes according to their properties. These properties are derived according to the services each node can provide. A value is given for each service and these values are decomposed to a single value which becomes the property that varies for each peer. The researchers were successful in implementing, on a single system, a mobile agent that can gather these values, cluster the nodes, show the clustering, and also show the connection or links between the nodes. Simulation results show that the proposed algorithm cluster nodes faster than the k-means clustering and provided the least average delay time for searching data in a network. Also, the proposed algorithm has the least overhead compared to other methods. In the simulations of this study, properties of each peer are pre-generated. Future works may include but is not limited to: the computation of the properties by the mobile agent where the values of the services are shown; the implementation of the method in an intersystem; and the setting of termination conditions on queries.

References

[1] "BearShare", From Wikipedia, the Free Encyclopedia, http://en.wikipedia.org/ wiki/BearShare

[2] "Kazaa", From Wikipedia, the Free Encyclopedia, http://en.wikipedia.org/ wiki/ KaZaA

[3] Vangie "Aurora" Beal, "All About Peer-to-Peer (P2P)", From Webopedia, http://www.webopedia.com/DidYouKnow/Internet /2005/peer_to_peer.asp.

[4] Xiaohui Gu, Zhen Wen, Philip S. Yu, Zon-Yin Shae, “peerTalk: A Peer-to-Peer Multiparty Voice-over-IP System”, Peer-to-Peer Applications, IEEE, Transactions on Parallel and Distributed Systems, Vol. 19, No. 4, April 2008, pp. 515 – 528.

[5] “BitTorrent (protocol)”, From Wikipedia, the Free Encyclopedia, http://en.wikipedia.org/wiki/BitTorrent_%28protocol%29

[6] “Advantages of Peer-to-Peer Networks”, From P2P World, http://www.soly rich.com /p2p-pros-cons.asp.

[7] Jie Wu (Edited by), Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks, Auerbach Publications, 2006.

[8] “Peer-to-peer”, From Wikipedia, the Free Encyclopedia, http://en.wikipedia.org/ wiki/Peer-to-peer

[9] Doan B. Hoang, Hanh Le, Andrew Simmonds, “PIPPON: A Physical Infrastructure-aware Peer-to-Peer Overlay Network”, TENCON 2005 – IEEE Region 10 Conference, November 2005, pp. 1 – 6.

[10] Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, Randy H. Katz, “OverQoS: An Overlay based Architecture for Enhancing Internet QoS”, In proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation, Vol. 1, pp. 71 – 84.

[11] Angelos Stavrou, Debra L. Cook, William G. Morein, Angelos D. Keromytis, Vishal Misra, Dan Rubenstein, “WebSOS: An Overlay-based System for Protecting Web Server from Denial of Service Attacks”, Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 48 , No. 5, August 2005, 781 – 807.

[12] “Overlay network”, From Wikipedia, the Free Encyclopedia, http://en.wikipedia.org/wiki/Overlay_network.

[13] Jiawei Han, Micheline Kamber, Data Mining Concepts and Techniques, Second Edition, Morgan Kaufmann, 2006.

[14] Jing-Feng Guo, Yu-Yan Zhao, Jing Li, “A Multi-Relational Hierarchical Clustering Algorithm Based on Shared Nearest Neighbor Similarity”, In proceedings of the Sixth International Conference on Machine Learning and Cybernetics, August 2007, pp 3951 – 3955.

[15] Qing-Bao Liu, Su Deng, Chang-Hui Lu, Bo Wang, Yong-Feng Zhou, “Relative Density Based K-Nearest Neighbors Clustering Algorithm”, In proceedings of the Second International Conference on Machine Learning and Cybernetics, November 2003, pp. 133 – 137.

◐ 저 자 소 개 ◑

Michael Angelo G. Salvo

2007 West Visayas State University, Philippines
BS in Information Technology
2008 ~ currently Kunsan National University, South Korea
Graduate student in Master's course major in Information and Telecommunications
Research interests: Distributed systems, multi-agents, mobile computing, ubiquitous sensor networks
E-mail: masalvo@kunsan.ac.kr

Romeo Mark A. Mateo

2004 West Visayas State University, Philippines
BS in Information Technology
2007 Kunsan National University, South Korea
Master of Engineering major in Information and Telecommunications
2007 ~ currently Kunsan National University, South Korea
Graduate student in Ph.D. course
Research interests: Distributed systems, data mining, fuzzy systems, multi-agents, mobile computing, ubiquitous sensor networks
E-mail: rmmateo@kunsan.ac.kr

이 재 완 (Jaewan Lee)

1984년 중앙대학교 이학사-전자계산학
1987년 중앙대학교 이학석사-전자계산학
1992년 중앙대학교 공학박사-전자계산학
1996년 3월~ 1998년 1월 한국학술진흥재단 전문위원
1992 ~ 현재 군산대학교 교수
관심분야 : 분산 시스템, 운영체제,실시간 시스템, 컴퓨터 네트워크, 멀티미디어 등
E-mail: jwlee@kunsan.ac.kr