Hierarchical Structured Multi-agent for Distributed Databases in Location Based Services

Romeo Mark A. Mateo[a], Jaewan Lee[a] and Oh-Hyun Kwon[b]

[a] *School of Electronoc and Information Engineering, Kunsan National University*
68 Miryong-dong, Kunsan Chonbuk 573-701, South Korea
Tel: +82-63-469-4696, Fax: +82-63-469-4696, E-mail:rmmateo@kunsan.ac.kr

[b]*Department of Computer Engineer, Tongmyong University*
E-mail: ohkwon@tit.ac.kr

Abstract

Location management is very important in location-based services to provide services to the mobile users like banking, city guides and many more. Ubiquitous and mobile devices are the source of data in location management and its significant operations are update and search method. Some studies to improve these were presented by using optimal sequential paging, location area scheme and hierarchical database scheme. In addition, not all location services have the same access methods on data and it lead to difficulties of providing services. A proposed location management of multi-agent architecture is presented in this study. It shows the coordination of the agents on the distributed database of location-based services. The proposal focuses on the location management of the mobile object presented in a hierarchical search and update. Also, it uses a nearest neighbor technique for efficient search method of mobile objects.

Keywords:

Location-Based Services; Distributed Database; Location Management; Multi-agent

Introduction

There are a lot of information services that can be obtained by using mobile and ubiquitous devices. In cellular phones, we have services like banking, city guides, games and many more. Ubiquitous devices like smart cards have less resources but powerful to provide information like location monitoring and awareness because of its ease of use. These services are provided by the location-based service or LBS [1], [2], [3]. The services that are available with the location based services are becoming more sophisticated but also can be acquired at any place and in any time.

Information about the moving object is important and not all location services have the same database access schemes. Different location services simply mean different deployment of the databases. Using software agent [4] solves the problem in the distributed location services. The agent shares its data with the other parts of the system so that information like mobile location is visible to other location service.

One of the important functions in mobile computing environment is to determine the mobile user's current location. Location management in cellular networks [5] deals with how to track mobile users within the network provided by interactive components like base station, mobile switching center and public switched telephone network.

In this paper, we proposed an approach of using multi-agent for the location management within the distributed environment. The architecture shows the design of a multi-agent for the location management that is manages by location agent manager to process the queries. These software agents are developed using CORBA technology to handle mobile objects. The location management of the proposed system presents a process of searching, updating and other concerns to provide an efficient location based service. We use the nearest neighbor algorithm for searching of mobile object within the network of nodes. The result of using the nearest neighbor implies that it provides an efficient search method. This paper is arranged as follows. In Section 2, we discuss some related works of location management. Section 3 explains the architecture of the proposed multi-agent location management. Section 4 presents the performance analysis of the search method using the nearest neighbor and simulation results in Section 5. Section 6 concludes the result and discusses some additional future works on the study.

Related Works

The current location of a mobile object is maintained in distributed databases to support efficient tracking of mobile objects. Two operations are essential: updating the stored location of a mobile object when it moves to another node and searching the location of a mobile object when it is requested for retrieval. The granularity of the information maintained about the location of an object varies. It can be a single site or a set of site. In the latter case, once the set is identified, the object is located by broadcasting a message to all sites in the set. Researches about these operations are discussed in the sub-sections.

Cellular Networks

Location management keeps track of an active mobile station in cellular networks [5]. A base station of a cellular network communicates with mobile station inside the cell. There is always tradeoff between the frequency of location updates and paging costs. If the update cost is zero then paging cost is maximized and if paging cost is zero then the update cost is maximized.

Many research works tackle the methods of locating a mobile object in cellular networks like in [6], [7] and [8]. Some introduce schemes to minimize the cost of paging and updating of locations. Research on location area schemes [5] discussed how the mobile station updates its location whenever it moves into a cell which belongs to new location area. Optimal sequential paging [6] presents a polynomial-time algorithm to solve the problem of minimizing the average paging costs. A predictive distance-based mobile tracking scheme is presented in [7] and a dynamic location update scheme in velocity-based scheme [8] is also presented.

Hierarchical Location Management

Hierarchical scheme of search and update method is another approach for location management discussed by Pituora, et. al. [11]. The study uses a hierarchical distributed location database to track mobile object in which the nodes are networked in a tree-like structure. It extend two-tier schemes by maintaining a hierarchy of location databases, where a location database at a higher level contains location information for objects located at levels below it. A location database at a leaf serves a single l-cell and contains entries for all objects currently in this l-cell. A database at an internal node maintains information about objects residing in the set of l-cells in its sub tree. For each mobile object, this information is a pointer to an entry at a lower database. As shown in Figure 1, for a mobile object x residing at l-cells 18, there is an entry in the database at node 0 pointing to the entry for x in the database at node 2 points to the entry for x in the database at node 6, which in turn points to the entry for x in the database at node 18.

In addition, a research of hierarchy of agents in the network nodes that acts as mobile location databases [10] can be a model for location management using agents. Similar hierarchical scheme is used in this research except that it uses mobile agents to store the replication of data and applied for the location management.

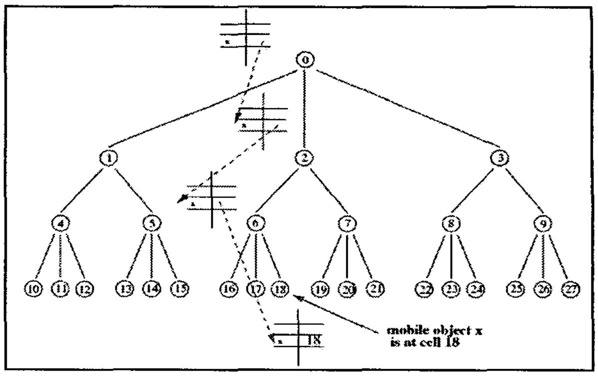

Figure 1. Hierarchical location schema

Proposed Multi-agent Location Management Architecture

The goal of the proposed system is to provide an efficient location management for distributed databases on location-based services. Figure 2 shows the architecture of the proposed system. Each LBS has a location management agent which accesses the database and communicates with other agents like the location agents and collector agents. It simply manages all the task of every agent connected with it as shown in Figure 4. The collector agent updates and deletes the mobile object collected from sensors or embedded systems. Location agents are the medium of communication of each location management agent in a LBS or nodes. They are arranged in hierarchical tree-like structure of networks as shown in Figure 3.

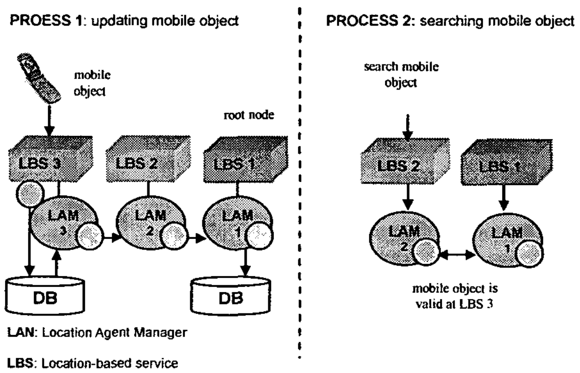

Figure 2. Proposed location management of multi-agent

Figure 2 shows the process of the proposed location management. In process 1, mobile objects are stored in the database of LBS. The location agent manager 1 (LAM 1) copies the mobile object and passes the value to LAM 2. The update method continues until it reaches the root node. Figure 2 shows the update method of location agent. Each mobile ID from the lower-level passes the value to higher-level location agent. The mobile object is replicated to higher-level location agents. In process 2, a search method is executed to find a mobile object. A query in LBS 2 searches for a mobile object in the local node. If the data is not found in the local node then the location agent LA 2 communicates with LA 1 to verify the mobile object's availability. If the mobile object is not registered in the LBS then the search continues until it reaches the root node. If a mobile object is found on LBS then it returns the value of the node's location where the mobile object is registered.

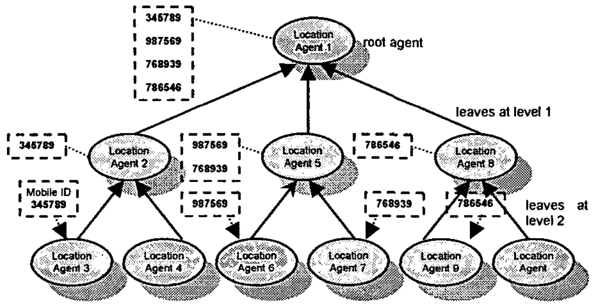

Figure 3. Hierarchical update method of the location agent

Location Agent

One of the components of the multi-agent in the location management architecture is the location agent. Location agent's task is to communicate with the parent agent on the hierarchical structure of the location agents. Figure 3 presents the hierarchical update method of the location agent. The mobile ID gathered by the lower-level agents update the value to the higher-level agent. The location agent has no access to the database. One-way communication with location agents minimizes the load of the system. In Figure 4, the location agent is illustrated as a medium of communication between each node. Also, location agent handles the proposed search method with nearest neighbor algorithm. Further explanation of the algorithm is discussed Section 4.

There are in some cases that the location service process queries which look up for more than one mobile object or other cases of queries. There are many types of queries that are cited in [13]. An example is graphical multitasking where a user request to search for all mobile objects in the current specified region and a nearby search can be answered by hierarchical networks of location agents.

Collector Agents

The main functions of location services are to collect and process the mobile objects to serve the mobile user subscribers. Location service is represented by nodes and also handles the process of queries to client's request. Active badge locating system [12] is one example of location service which provides information of staff members wearing the active badge within the establishment. The proposed system uses collector agents to collect data or mobile objects. These agents communicate with the location agent manager to update mobile objects. In Figure 4, the collector agent presented between the input device and location agent manager. All input data like mobile object is processed to the collector agent and send it again to the location agent manager.

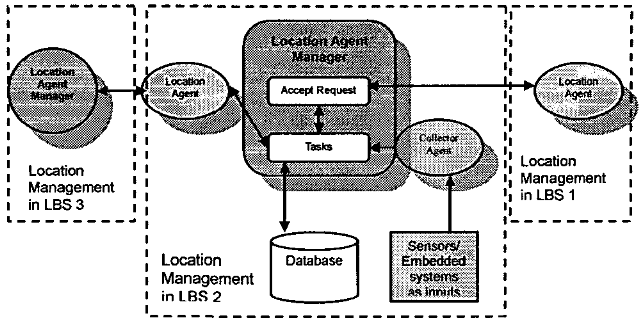

Figure 4. Multi-agent components

Location Agent Manager

There are different types of databases used for location services and the structure of the systems also varies. In [4] the location service uses software agents to solve data management problem. The location agent listens to all data at every available location services. The proposed system's main component is the location agent manager or LAM. The LAM is installed in LBS. The LAM function is to access the database of location service and communicates to other agents within the LBS. Also, this location management agent manages all the task of the agents communicating through it. It is the main processor of the location management. Figure 4 shows the LAM as the main component of the proposed system.

Data caching is important in mobile environments. Besides improving system performance, it saves power due to less data transmission and improves data availability in case of disconnection. In the proposed system, LAM uses cache to store replicated mobile object. However, the problem about data caching is maintaining the cache consistency. In general, cached data items become invalid if they are updated at the data server.

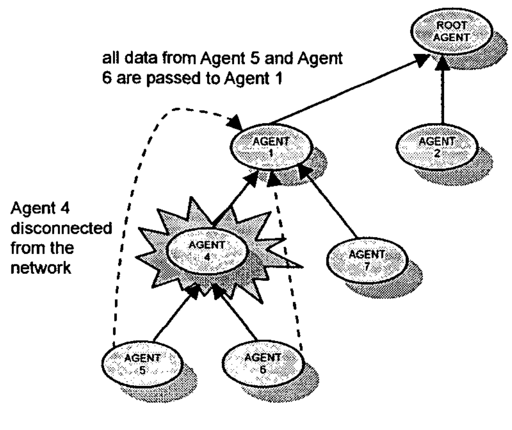

Figure 5. Fault management of agents

Fault Management

The location agent of the proposed system has location agent manager that analyzes the tasks to execute. Figure 4 shows the components of multi-agent. One of the tasks of the location agent manager is the fault management. In the case of leaf nodes, the agents know its parent's parent or least ancestor so that if there is no communication between the parents, it searches for the least ancestor to establish its communication. If the root node fails the connection with its sub nodes then the sub nodes search for the available root and pass the data to the new root. If the node recovers then the sub nodes establish the link to its original parent. The node that recovers the link gathers all the updated data on its sub nodes and deletes all the obsolete data. Figure 5 illustrates the fault management of the agents in the hierarchical structure.

Performance Analysis

The proposed multi-agent location management includes a search method using a nearest neighbor algorithm. This is handled by location agents. We assume a hierarchy of location database appropriately placed at each node in a mesh network. To allow maximum flexibility in the design of the proposed location management, we consider hierarchies with a variable number of levels. The region is covered by each leaf database corresponds to a unique physical address. Figure 6 shows the design of the nodes.

Nearest Neighbor Algorithm

The nearest neighbor (NN) feature method is defined as the distance between the closest pair of objects. It involves pairs consisting of one object from each group. A research of using nearest neighbor on location based system techniques like in [14] is applied on query processing for mobile object. In our case, we have a set of nodes in a mesh network and each node has specific distance to each other. We calculate the distance from a source node to the other node by using the Equation 1 below:

$$\min d(n_i, n_x) = d(n_i, n_x) \text{ where } x = 1,2,\ldots,n \quad (1)$$

The n represents a node and n_i is the source node. The algorithm used to locate the nodes which have the least distance. The calculation of determining the distance of all nodes which is less than the distance from the source node to its parent is given by Equation 2. The p_i indicates the parent node of n_i. Once the nearest nodes are determined then process the query from each node starting from the nearest node.

$$\sum dist(n_i, n_x) < dist(n_i, p_i) \quad (2)$$

Search Method using NN

The proposed search method uses a NN algorithm which detects a near location node to execute the query. First, it executes the NN algorithm to determine the nearest nodes then, sends the query. If the query is not valid in the near nodes, it sends the query to the parent node. If the query is valid then it returns the value of the node's location back to the source node. This method minimizes many lookups on the nodes if the search for the mobile object is found on the nearest node. Figure 10 shows the average hop counts by searching on the nearest node and otherwise.

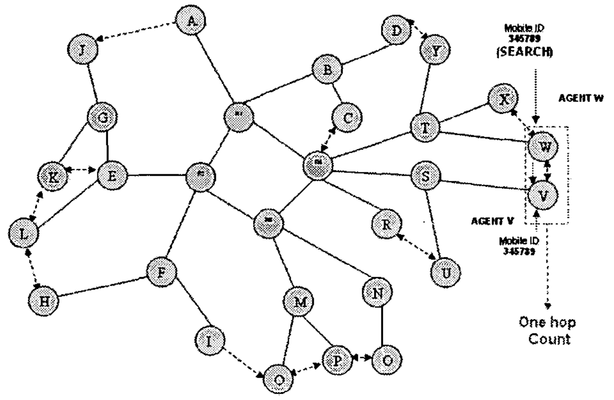

Figure 6. Search method using NN algorithm

In Figure 6, node V is the nearest node of W (not a parent). Here, agent W communicates with the agent V to process the query. It does not need to go on other agents which are at the higher-level to process the query. There is only one hop count in this point. The algorithm is presented in Figure 7.

```
//search the nearest neighbor node
List all Node Which dist(Node, SourceNode)
< dist(ParentNode, SourceNode)
i = 0
For all List Do
        AgentSearch (MobileID, Node(i))
        If BackOriginalNode = True
        Then End
        i = i + 1
Loop
//search the parents up to the root
GoTo ParentNode

Function AgentSearch (MobileID, Node)
{
GoTo Node
Search all List mobileID in Database
        If MoblieID(List) = MobileID
        Then Return Node Location
        BackOriginalNode = True
}
```

Figure 7. Nearest neighbor algorithm

Database Operation Analysis

We compare the database operation estimation of using the proposed search method with NN algorithm. The number of hops is determined by the level of the nodes and the search method is done from the lowest-level node up to the highest-level. Having inputs of 8 mobile objects per nodes and considering the total number of database operation to process the mobile object as outputs, we calculate the result of the average database operation. The formula is shown in Equation 3.

$$\frac{1}{x}\sum n_l D_{cost} + D_{cost}(cn_{left}, cn_{right}) \quad (3)$$

Equation 3 calculates the total average database operations of search in the hierarchical structure of nodes. The n indicates the nodes on level l and x is the total number of nodes. The total database operation is indicated by D_{cost} is increasing as the level of the nodes increases because it has the data of every child nodes and so its local database cost is added by the $D_{cost}(cn_{left}, cn_{right})$ which indicates the database cost of its left and right child nodes. The analysis result of the equation is shown in Figure 9. The graph shows that the search using nearest node minimizes the database operation while the normal search increases dramatically.

Experimental Evaluation

The simulation used a database consisting of 34 nodes, 80 mobile ID and 372 data replicated by agents were stored within the network. The simulation platforms used in the research were IBM compatible PC with Windows and Linux operating systems for the nodes, Borland Visibroker, C/C++ and Java as development of software agents, MySQL for the database and HBE-EMPOS II as collector of mobile objects.

Results of the Proposed Search Method

The experiment used a network of computers to implement the simulation. The CORBA was used for the simulation to perform interoperability on the operating systems. First, the experiment used the hierarchical search method of a mobile ID in the network. Then, the proposed search method was simulated. The proposed technique provides fast search if the mobile object is near the node being searched and also enhances the efficiency in the database operation. The simulation result is shown in Figure 8 which searches a mobile ID within the network.

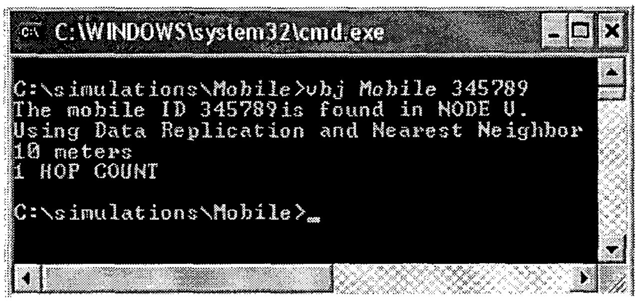

Figure 8. CORBA Implementation

In addition, it was observed that there were additional numbers of hop count when implementing the proposed search depending on the number of nearest nodes but the search was more efficient. Figure 9 shows hop count results. There are 34 nodes and each node can have two children nodes. There are 5 levels of nodes in the hierarchical structure and the root is at the level 0. Two cases were described in the situation. First, the mobile object is found on the nearest node and second when it is not found on the nearest node. The hop counts would minimized if the search for the mobile object is found on the nearest node.

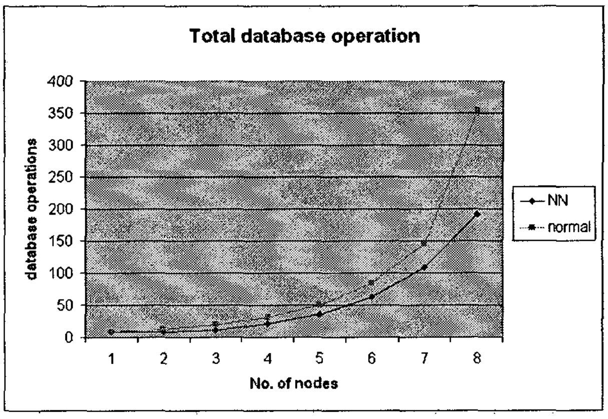

Figure 9. Database operation performance

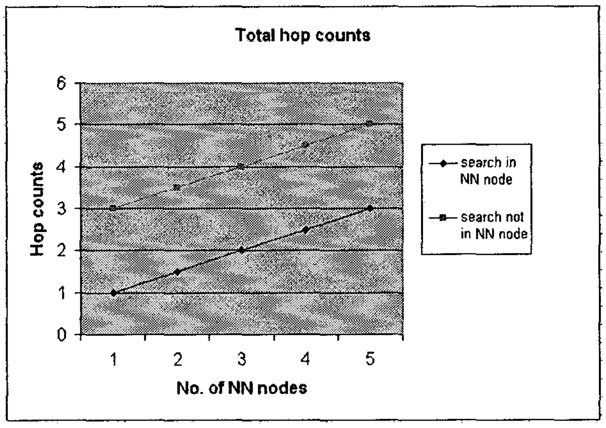

Figure 10. Hop counts result using NN algorithm

Conclusion and Future Works

In this paper, we proposed an approach of using multi-agent for the location management within the distributed environment. It used agents to perform data replication by storing the mobile information on the child agents. A hierarchical process of searching and updating mobile objects was also presented. There was an improvement in location management by implementing the nearest neighbor algorithm. The searching does not need to go to all of the parent nodes when the mobile identification was found on the nearest nodes of the source node, which implies efficient search. The simulation result implies that the technique using NN algorithm minimizes the database operations.

The research is only limited on mobile tracking and does not include physical and logical mapping of the location. The additional functions of the LBS should be studied in the future. The cache and other auxiliary storage were not given emphasis but will be the subject of our future works.

Acknowledgement

The authors thank the Korean Association of Information System, and Dr. Haeng Kon Kim. Also, the authors would like to appreciate Bobby D. Gerardo who exerted his efforts on editing the paper to improve the quality of the contents and other laboratory members of the Distributed System Laboratory, Electronic and Info. Engineering, Kunsan National University for their help.

References

[1] Jensen, C., Christensen, A., Pedersen, T., Pfoser, D., Saltenis, S. and Tryfona, N. (2001). "Location-Based Services: A Database Perspective," *Proceedings of Scandinavian GIS.*

[2] Jensen, C. (2002). "Research Challenges in Location-Enabled M-Services," *Proceedings of the Third International Conference on Mobile Data Management,* pp. 3-7

[3] Roth, J. (2004). "Flexible Positioning for Location-Based Services," *IADIS International Journal on WWW/Internet,* Vol. 1, No. 2, pp. 18-22.

[4] Jaric, P. (1999). "An Agent-Based Location System", Master's thesis, Computing Science Department, Uppsala University.

[5] Zhang, J. (2002). "Location Management in Cellular Networks," *Handbook of wireless networks and mobile computing,* pp. 27 – 49.

[6] Krishnamachari, B., Gau, R. Wicker, S. and Haas, Z. (March 2004). "Optimal Sequential Paging in Cellular Networks," *Wireless Network,* Vol. 10, pp. 121 – 131.

[7] Liang, B. and Haas, Z. (1999). "Predictive Distance-based Mobility Management for PCS networks", *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies,* pp.1377-84.

[8] Wan, G. and Lin, E. (1997). "A Dynamic Paging Scheme for Wireless Communication Systems" *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking,* pp.195-203.

[9] Thiran, Ph., Hainaut, J-L.,Bodart, S., Deflorenne, A. and Hick, J.-M. (1998). "Interoperation of Independent, Heterogeneous and Distributed Databases Methodology and CASE Support: the InterDB Approach," *Third International Conference of Cooperative Information Systems*, p. 54.

[10] Lee, K., Lee, H., Jha, S., and Bulusu, N. (2004) "Adaptive, Distributed Location Management in Mobile, Wireless Networks" *Proceedings of the 2004 IEEE International Conference on Communications*, pp. 4077 - 4081.

[11] Pitoura, E. and Fudos, I. (2001) "Distributed Location Databases for Tracking Highly Mobile Objects", *The Computer Journal*, Vol. 44, No.2.

[12] Want, R., Hopper, A., Falcao, V. and Gibbons, J. (1992) "The Active Badge Location System", *Proceedings of ACM Transactions on Information Systems (TOIS)*, pp. 91-102.

[13] Seydim, A., Dunham, M. and Kumar, V. (2001) "Location dependent query processing", *Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access*, pp. 47 - 53

[14] Chi, H., Kim, S. and Ryu, K. (2005) "A New Continuous Neighbor Technique for Query Processing on Mobile Environments", *Proceedings of the International Conference on the Computational Science and Its Applications*, pp. 977-987.