

Improve Mobile Agent Performance by using Knowledge-Based Content

Tarig Mohamed Ahmed

Assoc. Prof., Department of MIS, Prince Sattam Bin Abdulaziz University AL-Kharj, Saudi Arabia
Faculty of Mathematical Science, University of Khartoum, Khartoum, Sudan

Abstract—Mobile agent technology is one of the mobile computing areas. This technology could be used in several types of applications, such as cloud computing, e-commerce, databases, distributed systems management, network management, etc. The purpose of this paper is to propose a new model for increasing the mobile agent systems performance. The performance is considered as one of the important factors that makes the system reliable. This paper suggests a knowledge-based content to be used to improve the mobile agent systems performance. In the beginning, this work started by conducting intensive survey about related models and mechanisms to investigate the gaps in the performance. A comparative discussion has been conducted between some researches issued and the proposed model. The proposed model has been described in full details based on the components. A scenario-based approach has been used to implement the proposed model by using .Net framework and C# language. The model has been tested and evaluated based on different scenarios. As findings, the overall performance has been improved by 83% when the knowledge-based content is used. In addition, the system performance will improve automatically by the time because the content of the knowledge is increased. The proposed model is suitable to be used in any type of mobile agent applications. The originality of the model is based on conducted survey and own knowledge.

Keywords—Mobile agent; mobility; performance; intelligent system

I. INTRODUCTION

Mobile agent technology is one of the mobile computing areas. This technology could be used in several types of application such as cloud computing, e-commerce, databases, distributed systems management, network management.etc. The mobile agent systems have many benefits if we compared with the client/server model such as saving network bandwidth, reducing network latency and reducing network consuming cost. The mobile agent system works based on concept of remote programming. The mobile agent travels to several nodes to accomplish tasks on behalf of users. The agent is one of the system components. It is represented as object which consists of two parts: Data Stat and Code. The data state represents the information domain of the agent. The code represents the statements that will be executed in the hosts (Service Providers). By using an itinerary table and mobility mechanism, the mobile agent travels among network hosts. The mobile agent home creates the agents according to users' requests. In addition, it dispatches and receives the agents with results. The host or service provider represents the node that is visited by mobile agents. The hosts can receive

multiple mobile agents simultaneously. The mobile agent system uses a communication mechanism that allows the agents to communicate together and with other system components. The mobility feature is a key feature of the mobile agent which allows the agents move from host to another. Vigna and Fuggetta et al. [12] defined the agent component as two parts: execution and resources units. The execution unit represents a computation algorithm. The resource unit represents as information domain that will be used by the execution units. There are two types of mobility: strong and weak mobility. Strong mobility allows the mobile agent that can carry a code and an execution state during the journey. Weak mobility allows the mobile agent to carry only a code with some initial values.

The mobile agent performance is one of the key issues [5] that make the system reliable and successful. The performance depends on many factors, one of them is the agent journey duration time. The duration time depends on two items: number visited hosts and the services execution time. In this paper a new model is proposed to improve the mobile agent performance. The model is based on reducing the number of visited hosts by the mobile agent. The main idea of the model is to use a knowledge-based content. The knowledge-based content represents the mobile agent experiences from pervious journeys. Before the mobile agent starts a new journey, it should consult the knowledge-based content in order to check if there is a previous knowledge could be used to reduce the number of visited hosts. This idea works only if the mobile agent wants to make a selection or searching for benefits among hosts for example, buying books or tickets or any goods.

The reset of the paper was organized in five sections: Section 2 explores some models and mechanisms the mobile agent performance with some discussions. Section 3 presents the proposed model with full details of all components. The model implementation has been mentioned with full discussion of the results in Section 4. This work was concluded in Section 5 with some recommendations as future work to enhance the model.

II. RELATED WORK

In this section, some researches issued have presented and discussed. These researches were agreed on how to find a mobile agent model with high performance by using different mechanisms such as: network protocols, using parallel

processing, ranking services provider, etc. the following section presents some of them.

Salamat et al. proposed extended hierarchical query retrieval (EHQR) approach to enhance the mobile agent performance. The main idea behind this approach was to send many agents simultaneously in order to reduce the time taken for tasks. To evaluate EHQR, two experiments had been conducted by using query online and offline [1]. By using SNMP (simple network management protocol), Rantes et al. developed a model for evaluating mobile agent performance factors. After conducting many experiments, the results mentioned that the mobile agent performance depends on the network management and some parameters related to a network topology, network latency [2]. Holt et al. also used SNMP to analysis mobile agent performance factors. The model evaluated the mobile agent in two environments: Local Area Network (LAN) and Wide Area Network (WAN). The study mentioned that the bounded size of the mobile agent is optimal in a large network domain and the adopting of clustering strategy controls the mobile agent size [3]. Devadas et al. proposed a knowledge based component in a mobile agent system that can help the agents to communicate together. By this way the mobile agent performance will be increased [4]. Tarig proposed a new mechanism for increasing the mobile agent performance by reducing the mobile agent size during the agent journey. The mechanism called Free Area Mechanism (FAM). The mechanism was implemented using .Net framework and many experiments had been conducted to test the performance [5]. Sasirekha et al. proposed a new mechanism to improve the mobile agent routing algorithm. The algorithm called cluster-chain mobile agent routing (CCMAR). It used a wireless sensor network (WSN) into a few clusters and runs. Two phases were used to implement the algorithm. First, the nodes in the chained cluster aggregate the data in the cluster. Second, the mobile agents collect the data that aggregated in the cluster [6]. Aloui et al. proposed a solution for Multiple agents Itinerary Planning (MIP). The solution was based on agent's location and their size to make balance in consuming network energy. After conducting many experiments, the results mentioned that the performance was increased [7].

The accurate prediction for the resources is very important for the mobile agent to improve the performance. Chaudhar et al. proposed to use Cognitive Agent in Mobile Ad hoc Network. The cognitive agent makes the mobile agent thinking like human to take the right decision regarding to the resources. By this way, the mobile agent can determines the best traffic plan to achieve its tasks [8]. Channappagoudar et al. had conducted a study related to the resource allocation protocol. The study used static and mobile agents to evaluate the performance. The main idea of the proposed protocol was to allow the static agents to collect resource information about nodes in the network and providing the mobile agent by this information. By this way, the mobile agents will increase their performance [9]. Prapulla et al. proposed a model for multi mobile agents to reduce the energy consummation and latency. The model based on two types of mobile agents: Link Agent and Data Agent. The link agent aimed to monitoring the network resources and status. The data agent aimed to transfer

data among nodes. The idea of this model helps mobile agents to prepare their itinerary tables. By clustering the network nodes, the model was implemented and the efficiency was discussed [10]. Based on opinion-based, Zuo et al. proposed a model for increasing the mobile agent system performance. The model ranks the reputation of network nodes by aggregating information. The node reputation ranking was based on set of categories such as services quality. By this way, the mobile agents were owned valuable information before starting their journeys and the overall performance will be increased. The model was implemented and evaluated by using Algets technology [11].

Baek et al. [13] suggested planning algorithms tried to search a minimum number of agents, and the total consuming time of route by setting lime of time execution. There are two important planning factors affecting the performance of the agent system in the network environment that are the mobile agent's itinerary and the number of the mobile agent. The experiment of this research proves that if the size of a mobile agent is begin increased while retrieval operations are performed, the bandwidth varies from link to link. In this case, the agent will consume more time. Cook [14] has mentioned that building software system composed of mobile agents introduces interesting new concerns for software engineering research. He described some assumptions behind mobile agent systems and software engineering. One of them: Code is cheaper to move than data and this assumption implies that the size of the mobile agent is important and it should be reduced.

As mentioned above, all these models or mechanism were aimed to improve the mobile agent systems performance. If we compare them with our proposed mechanism, we find the dynamic and increment improvement of performance by using the knowledge-based. This fact is more suitable with nature of the information systems because the information related to the service providers are not fixed and rapidly changed. Also, the proposed mechanism allows the mobile agent systems automatically adapting with information available in the knowledge based-content to improve the performance. In addition, the performance will depend on the mobile agents' experiences.

III. PROPOSED MODEL

A. Model Concepts and Components

This research aims to propose a new model to increase the mobile agent systems performance. The main idea behind this model is to use a knowledge-based content. The knowledge-based content helps the mobile agent system to reduce number of visited hosts. It can provide the mobile agents by valuable information related the services located in hosts. The knowledge-based content is incremental database that consists of information that collected by the mobile agents. When the mobile agent completes its journey, the mobile agent home extracts the information about hosts available in the mobile agent and stores it in the knowledge-based content. This process is repeated every time when the mobile agents back to their home. In addition, before the mobile agents start their journey, they should optimize their itinerary tables based on information available in the knowledge-based content. By this way the number of visited hosts will be decreased and the

overall performance automatically is going to be improved. In addition, the knowledge-based content is dynamically updated and rapidly increased by knowledge the mobile agents' experiences. Fig. 1 presents the main components of the mechanism.

As presented above, Fig. 1 depicts the mobile agent model components: Mobile Agent, Mobile Agent Home, Knowledge-based content and Hosts. Each one has specific role in the model as following:

Mobile Agent is an object, which contains tasks to be performed on behalf of users. The mobile agent visits hosts according to its itinerary table. The journey duration depends of many factors and one of them is number of visited hosts. The number of hosts is specified based on the mobile agent's tasks.

Mobile Agent Home is a place where the mobile agents start their journey. After finishing their journey, the mobile agents return home with results. Mobile Agent Home is responsible to create the mobile agent and extract their results after completing their journey. The results will be saved in the knowledge-based content.

Knowledge-Based content is a knowledge container which can store services information provided by hosts. Before mobile agents start their journey, they visit the knowledge-based content to obtain information related to their duties. In addition, after mobile agents finishing their journey, they

come again to the knowledge-based content and update the information. By this way, the knowledge-based content is incremented and that make the model more efficient and intelligent. The knowledge format in the knowledge base depends on kind of services that are provided by the hosts.

Hosts represent the service providers. It can receive and serve multiple mobile agents simultaneously. The hosts could introduce the services for example, selling books, selling electronic devices, air tickets etc.

As mentioned above in Fig. 1, the model works in specific logic steps such as:

- 1) The mobile agent is created in the mobile agent home based on user's request. The mobile agent visits the knowledge-based content to search for related knowledge to its tasks. If any, the mobile agent can benefit from it by reducing the number of hosts in the itinerary table (Key issue to increase the performance).
- 2) The mobile agent returns to the home to start its journey by using updated itinerary table.
- 3) The mobile agent starts its journey by visiting hosts (Service Providers) to perform the tasks.
- 4) After completing the journey, the mobile agent returns home with results.
- 5) The knowledge that is collected by the mobile agent will be stored in the knowledge-based content.

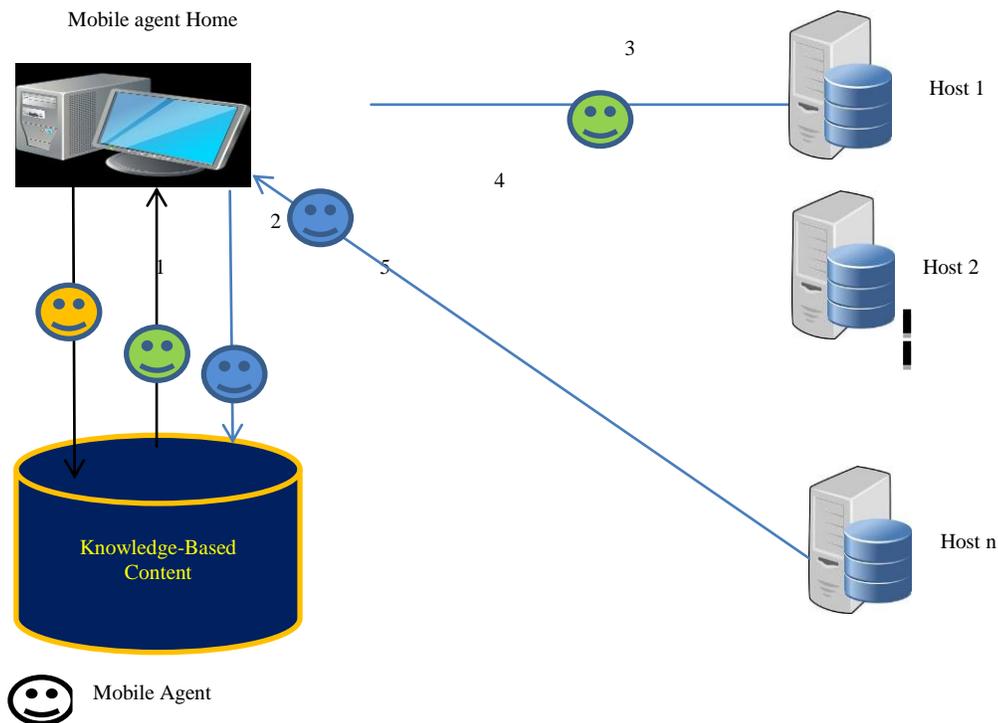


Fig. 1. Model components.

B. Model Implementation

By using .Net framework and C# language, the model has been implemented and tested. Each component has a separate program as following:

Mobile agent is represented as object form, a class called "Agent". The class consists of several variables represent the object state such as Agent Name, Start journey time, finish journey time, itinerary table, Tasks, etc. The class also contains some methods to access those variables. Fig. 2 presents the pseudo code for the Mobile Agent Class. This class also was assigned as a Serializable to enable the object to be converted as a network stream.

The mobile agent home was implemented as two separate programs: sender and receiver mobile agents. The sender

program was represented as class called "Agent_Sender". It consists of some variables and methods. It aims to create and dispatch the mobile agent object. For example, assigning the tasks, itinerary table and sending the mobile agent after converting it as a network stream. Fig. 3 presents the pseudo code.

The mobile agent home (Receiver program) was implemented as class. It aims to receive the mobile agents after completing their journey. It consists of two main methods: Receiving the mobile agent and extracting results from the mobile agents. The results will be stored in the knowledge-based content to be used in the future journeys. Fig. 4 presents the class of receiver agents. In addition, the program works as server listener to receive multiple mobile agents simultaneously.

```
[Serializable]
public class Agent
{
    public string agent_name;
    public DateTime start;
    public DateTime end;
    public String[] Tasks = new string[100];
    public int[]itinerary = new int[100];
    public int Current_location = -1;
    // Access methods
}
```

Fig. 2. Mobile agent class.

```
class Agent_Sender
{
    public Socket agent_socket;

    public Agent_Sender(Socket s)
    {
        this.agent_socket = s;
    }

    public void send_agent(Agent agent, int p)
    {
        int port = p;
        IPAddress ip = IPAddress.Parse("127.0.0.1");
        TcpClient client = new TcpClient();
        client.Connect(ip, port);
        NetworkStream stream_data = client.GetStream();
        BinaryFormatter bf = new BinaryFormatter();
        MemoryStream ms = new MemoryStream();
        //...
        client.Close();
    }

    // setting Tasks
    // Setting Nodes Information
    send_agent(Agent agent, int p)
}
}
```

Fig. 3. Mobile agent home (sender).

```
class Receiver_Agent
{
    public Socket agent_socket;
    public Receiver_Agent(Socket s)
    {
        this.agent_socket = s;
    }
    public void Extract_result(Agent a,int duration,string[] v)
    {
        OleDbConnection con = new
        OleDbConnection("provider=Microsoft.ace.Oledb.12.0; data
        source=d:\\AgentDB.accdb;Persist Security Info=False");
        con.Open();
        // Add result to database (Knowledge base
        con.Close();
    }
    public void create_agent()
    {
        byte[] agent_data = new byte[50000];
        int z = agent_socket.Receive(agent_data, agent_data.Length, 0);
        BinaryFormatter bf = new BinaryFormatter();
        MemoryStream ms = new MemoryStream(agent_data);
        Agent a = new Agent();
        ms.Seek(0, 0);
        a = (Agent)bf.Deserialize(ms);
        // create the mobile agent
    }
    static void Main(string[] args)
    {
        System.Console.WriteLine("***** MOBILE AGENT HOME *****");
        string ipaddress = "127.0.0.1";
        int port = System.Convert.ToInt32("8081");
        IPAddress ip = IPAddress.Parse(ipaddress)
        System.Net.Sockets.TcpListener listener = new
        System.Net.Sockets.TcpListener(ip, port);
        listener.Start();
    }
}
```

Fig. 4. Mobile agent home (receiver).

```
namespace host1
{
    class Program
    {
        public Socket agent_socket;
        public Program(Socket s)
        {
            this.agent_socket = s;
        }
        public static void send_agent(Agent ag, int p)
        {
            int port = p;
            IPAddress ip = IPAddress.Parse("127.0.0.1");
            TcpClient client = new TcpClient();
            client.Connect(ip, port);
            NetworkStream stream_data = client.GetStream();
            BinaryFormatter bf = new BinaryFormatter();
            MemoryStream ms = new MemoryStream();
            bf.Serialize(ms, ag);
            //...
        }
        public void create_agent()
        {
            System.Console.WriteLine("new agent under processing");
            byte[] agent_data = new byte[50000];
            int z = agent_socket.Receive(agent_data, agent_data.Length, 0);
            //...
        }
        static void Main(string[] args)
        {
            System.Console.WriteLine("**** The Service Provider Name: HOST No 1 ****");
            string ipaddress = "127.0.0.1";
            int port = System.Convert.ToInt32("8082");
            IPAddress ip = IPAddress.Parse(ipaddress);
            System.Net.Sockets.TcpListener listener = new System.Net.Sockets.TcpListener(ip,
port);
            listener.Start();
            System.Console.WriteLine();
            System.Console.WriteLine("##### Now Host 1 Running ##### ");
            try
            {
                while (true)
                {
                    Socket line = listener.AcceptSocket();
                    //...
                }
            }
        }
    }
}
```

Fig. 5. Host (service provider).

The host (Service Provider) was represented as a program. Each node has its one running program with unique IP address. The node can receive multiple mobile agents and it serves them according to their tasks. After completing their jobs, the node dispatches the mobile agent according to the itinerary table. Fig. 5 presents the host program. This program is run each time when a new host is started.

By using Microsoft Access 2007, the knowledge base was implemented as a database. The database consists of two tables. The first table was used to store results from mobile agent journeys. The second table represents the visited hosts.

C. Model Test and Evaluation

A scenario-based approach has been used to test and evaluate the proposed model. In the scenario, the mobile agent buys books on behalf of users based on best price. The user specifies the required book information. By using this information, the mobile agent searches among the nodes (Hosts), the required book and its price. The host represents a bookstore. After completing its journey, the mobile agent saves all collected information in the knowledge-based content. As sample, Fig. 6 presents multiple mobile agents visit Host 5.

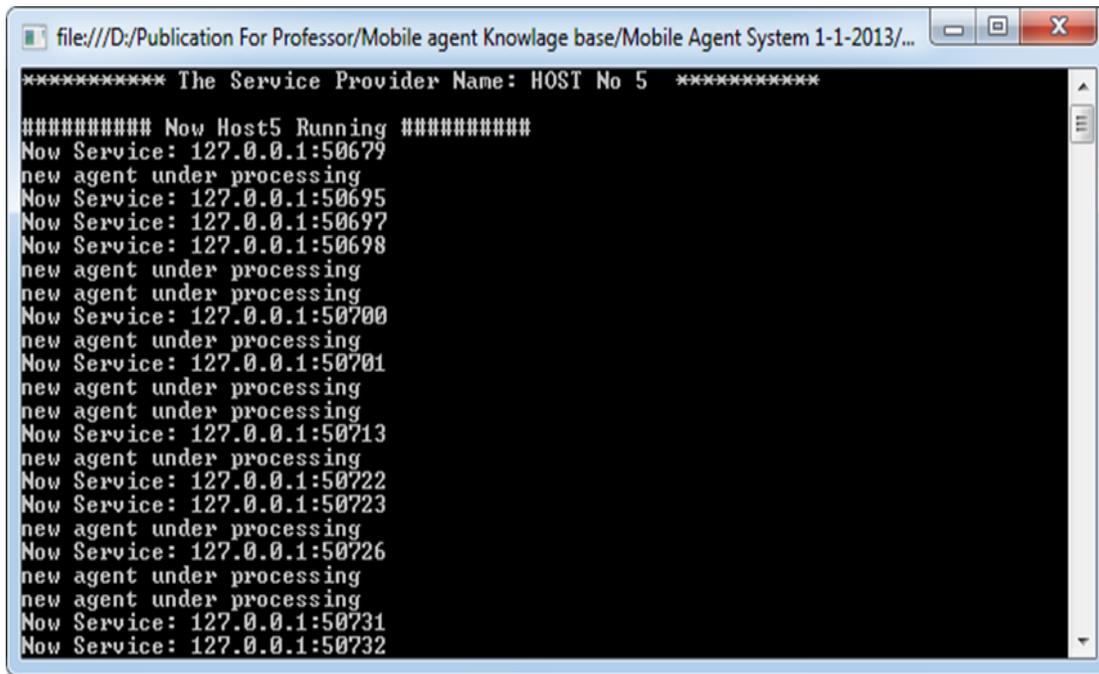


Fig. 6. Mobile agents under processing.

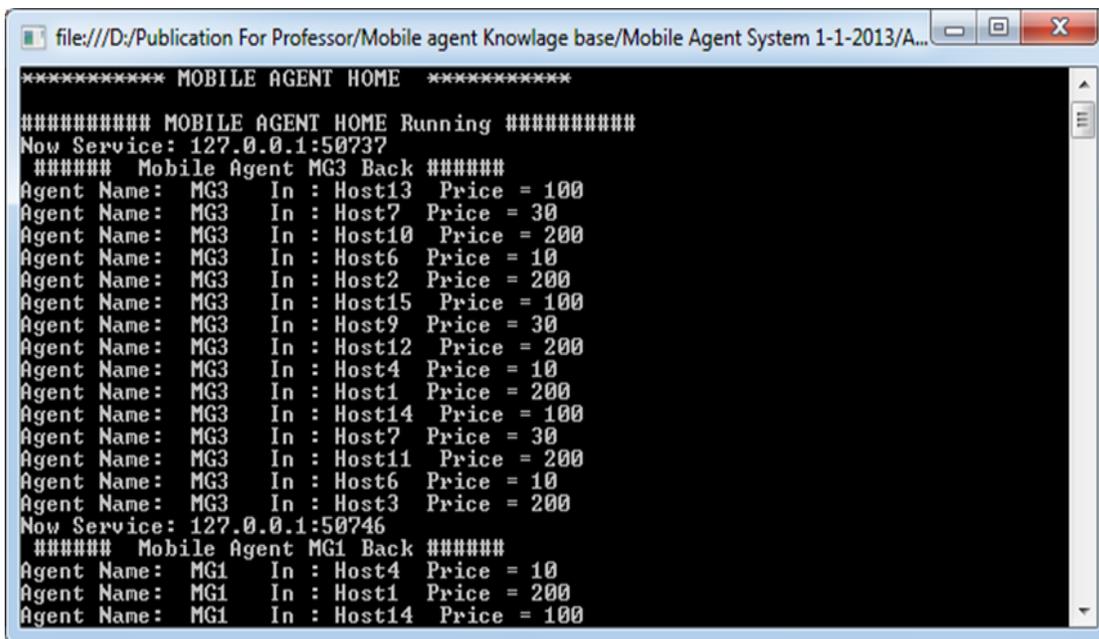


Fig. 7. Mobile agents return home.

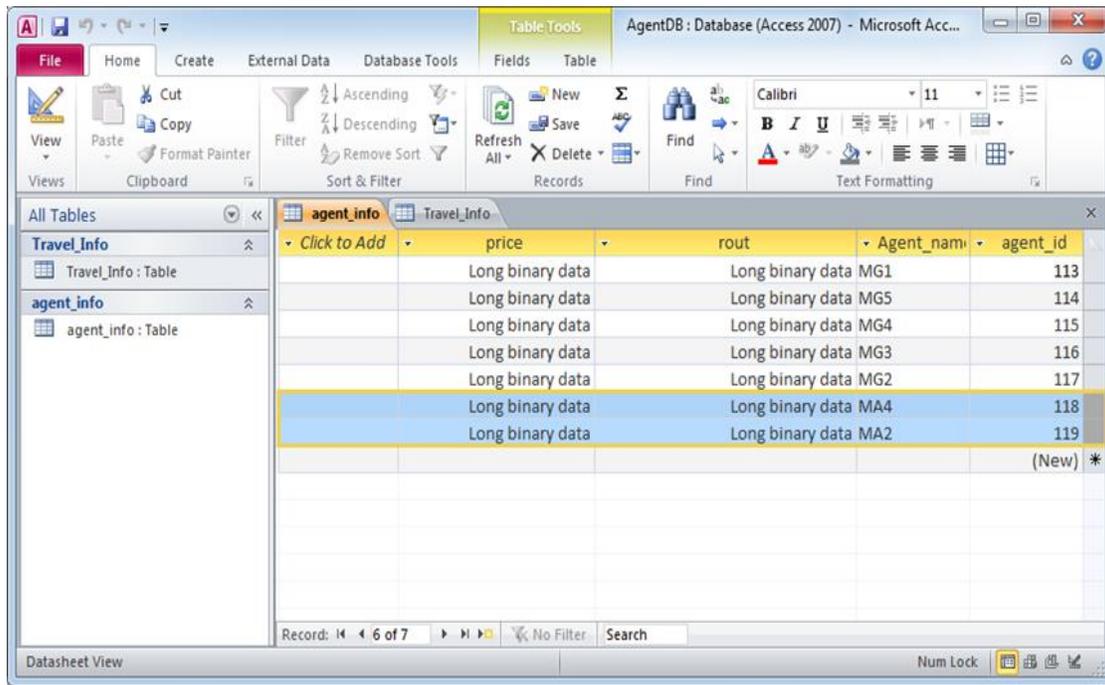


Fig. 8. Mobile agents in home.

In Fig. 7, the mobile agents return home with results from hosts.

The mobile agent home saves all information collected by the mobile agents in knowledge-based content. Fig. 8 presents the information as a table of object. Each object represents collected information of one journey as prices and visited hosts (route).

To evaluate the model, the durations of journeys have been computed with listing of visited hosts. In Fig. 9, all these

information are presented. As mentioned, the duration of Mobile Agents 4, 2(MG4, MG2). In the first journey the durations were 292 msec and 332 for MG4 and MG2 msec respectively. The two mobile agents had sent again for the same tasks. In the second journey, the durations were 56 msec and 51 for MG4 and MG2 msec, respectively. The time of journey was reduced by 83% average. According to this fact, the model has improved the performance of the mobile agent system by using the knowledge-based content.

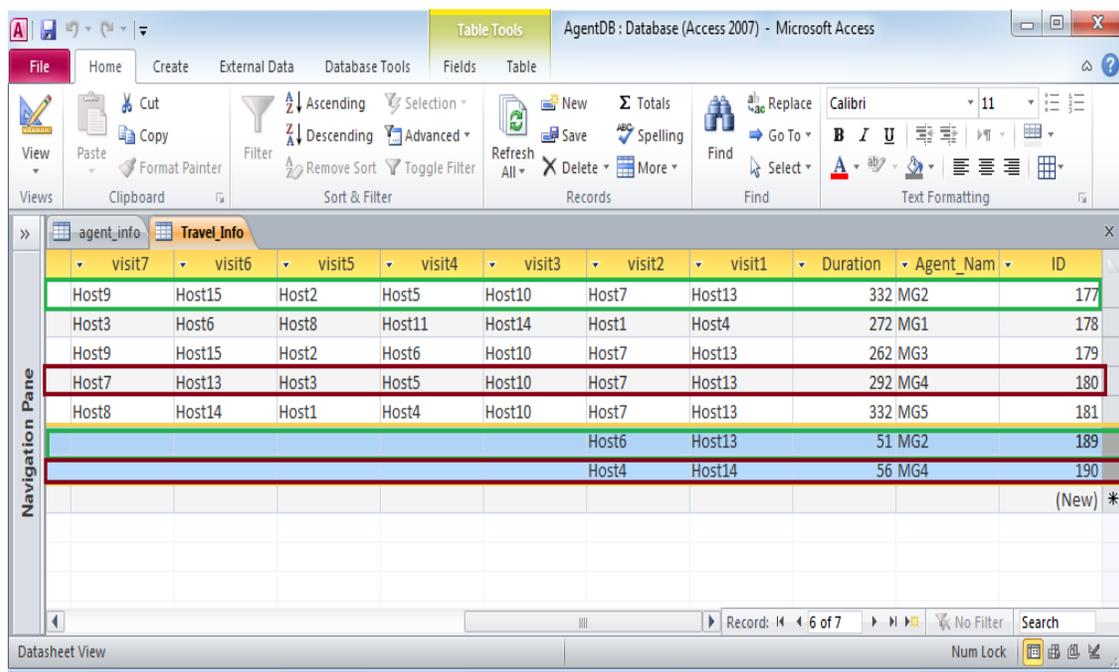


Fig. 9. Model performance measurements.

IV. CONCLUSION

The purpose of this paper is to propose a new model for increasing the mobile agent systems performance. The performance is considered as one of important factors that makes the system reliable. This paper suggests a knowledge-based content to be used to improve the mobile agent systems performance. In the beginning, this work started by conducting intensive survey about related models and mechanisms to investigate the gaps in the performance. A comparative discussion has been done between some researches issued and the proposed model. The proposed model has been described in full details based on the components. A scenario-based approach has been used to implement the proposed model by using .Net framework and C# language. The model has been tested and evaluated based on different scenarios. As findings, the overall performance has been improved by 83% when the knowledge-based content is used. In addition, the system performance will improve automatically by the time because the content of the knowledge is increased. The proposed model is suitable to be used in any type of mobile agent applications. The originality of the model is based on conducted survey and own knowledge.

As future work of this work, it will be a good idea if some effort be done in the knowledge-based content by setting protocol for knowledge formats. By this way the knowledge-based content will be used by heterogeneous mobile agent systems. In addition, this idea will help to share the knowledge between different systems.

REFERENCES

- [1] Ali Selamat, Hafiz Selamat, Analysis on the performance of mobile agents for query retrieval, Information Sciences, Volume 172, Issues 3-4, 9 June 2005, Pages 281-307, ISSN 0020-0255.
- [2] rantes, J. A., Westphall, C. B., Custódio, R. F., & de Chaves, S. A. (2010). Analytical model to evaluate the performance of mobile agents in a generic network topology. *Journal of Network and Systems Management*, 18(4), 357-373. doi:http://dx.doi.org/10.1007/s10922-010-9173-x
- [3] Holt, A., Huang, C., & Monk, J. (2007). Performance analysis of mobile agents. *IET Communications*, 1(3), 532-538.
- [4] Devadas, T. Joshva, and R. Ganesan. "INTELLIGENT AGENT-BASED KNOWLEDGE MANAGEMENT." *International Journal of Advanced Research in Computer Science* 3.2 (2012).
- [5] Tarig Mohamed Ahmed: "Increasing Mobile Agent Performance by Using Free Areas Mechanism", in *Journal of Object Technology*, vol. 6, no. 4, May-June 2007, pp. 125-140
- [6] Sasirekha, Selvakumar, and Sankaranarayanan Swamynathan. "Cluster-chain mobile agent routing algorithm for efficient data aggregation in wireless sensor network." *Journal of Communications and Networks* 19.4 (2017): 392-401.
- [7] Aloui, I., Kazar, O., Kahloul, L., & Servigne, S. (2015). A new itinerary planning approach among multiple mobile agents in wireless sensor networks (WSN) to reduce energy consumption. *International Journal of Communication Networks and Information Security*, 7(2), 116-122.
- [8] Chaudhari, Shilpa Shashikant, and Rajashekhar C. Biradar. "Traffic and mobility aware resource prediction using cognitive agent in mobile ad hoc networks." *Journal of Network and Computer Applications* 72 (2016): 87-103.
- [9] Channappagoudar, Mallikarjun B., and Pallapa Venkataram. "Performance evaluation of mobile agent based resource management protocol for MANETs." *Ad Hoc Networks* 36 (2016): 308-320.
- [10] Prapulla, S. B., et al. "Multi Mobile Agent itinerary planning using Farthest Node First Nearest Node Next (FNFNNN) technique." *Computation System and Information Technology for Sustainable Solutions (CSITSS)*, International Conference on. IEEE, 2016.
- [11] Zuo, Yanjun, and Jigang Liu. "A reputation-based model for mobile agent migration for information search and retrieval." *International Journal of Information Management* 37.5 (2017): 357-366.
- [12] Fuggetta, Alfonso, Gian Pietro Picco, and Giovanni Vigna. "Understanding code mobility." *IEEE Transactions on software engineering* 24.5 (1998): 342-361.
- [13] J. Cook, Software Engineering Concerns for Mobile Agent Systems, paper, proceeding of Workshop on Software Engineering and Mobility, Ontario, Canada, May 2001.
- [14] L. Ismail and D. Hagimont. A performance evaluation of the mobile agent paradigm. In *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pages 306-313, 1999.