Comparing the Use of Mobile Intelligent Agents vs Client Server Approach in a Distributed Mobile Health Application

Kevin Miller*, Gunjan Mansingh
Department of Computing, University of the West Indies, Mona, Kingston, Jamaica.

* Corresponding author. Tel.: 1-876-883-9487; email: kevin.miller02@uwimona.edu.jm
Manuscript submitted January 21, 2015; accepted August 5, 2015.
doi: 10.17706/jcp.10.6.365-373

Abstract: It is well known, that any reasonable size application that constantly accesses a database, will inevitably have to contend with performance issues as the application grows. These issues are compounded when the application is serving millions of requests, by heterogeneous clients. To tackle these issues, the use of mobile intelligent agents, instead of the traditional client server approach was explored. This solution was assessed by evaluating the use of mobile intelligent agents in a distributed medical application by the name OptiPres. In this application, instead of executing multiple queries remotely, mobile intelligent agents were used to migrate to database servers and execute queries locally while performing local processing. The aim of this technique is to reduce the network load and thus increase the responsiveness and speed of distributed applications. The preliminary results show that using mobile intelligent agents to collaborate in performing local queries outperformed the traditional approach to access database servers.

Key words: Client server, databases, decision support system, distributed, network, mobile agents.

1. Introduction

The traditional technique used in most distributed applications including the Internet, is the client-server approach. This approach involves, a client which is normally a system that relies on the services located on a remote system that is referred to as a server. Servers are usually more powerful than clients and provides mechanisms for clients to access their resources. Clients usually access the resources on servers by using message passing or remote procedure call. However, as the demand of these applications increases, it was realized that the traditional client-server approach was lacking in terms of bandwidth use and server flexibility, load balancing, availability to name a few [1], [2]. Additionally, with the reality of many applications' data growing (Big Data) and the pervasiveness of cloud computing, developers are constantly exploring different ways of making the design, implementation and maintenance of these distributed systems easier and the performance better. In order to solve the inadequacies of the client-server approach, one emerging technology that is being explored, is the use of mobile intelligent agents [3]. This is as a result of the potential benefits of using mobile agents (reduction in network load, parallel processing etc.,). Mobile agents are currently being explored in many distributed systems which includes network management systems, wireless networks and health systems [4]-[7]. An interesting area of research for mobile agent technology is the analysis of big data [8]. This is as a result of the potential benefits that can be gained, as it relates to parallel processing. Many companies have resorted to architectures such as Apache Hadoop [9] that supports the MapReduce programming model [10]. This strategy has been quite successful. However, many of the implementations still suffer from performance and reliability issues [8]. A proposed

implementation is to use mobile intelligent agents to manage such environments which was demonstrated in this paper [8]. Considering all the proposed benefits of mobile agent technology, and the nature of our application, we have decided to use it as the middleware in our distributed medical system. We predict that using mobile agents will improve the performance of our system as less time will be spent sending data over the network.

The remainder of this paper is as follows: The background literature will be discussed in Section 2. Section 3 will give a brief introduction to OptiPres. We then present the performance evaluation using mobile agents vs the client server approach. The final section, which is Section 4, will be on the conclusion and the future of this research.

2. Background

Mobile code [11] is basically a piece of software having the ability to traverse through a heterogeneous network executing itself automatically upon arrival on a host. The development of Java by Sun Microsystems and the explosive growth of the internet are among the main driving forces behind the exploration of mobile code technology for distributed applications. Although there are other languages that supports mobile code programming (Limbo, Object Caml, Obliq, Telescript and Safe-Tcl) , Java is by far the most popular and this is evident in the different mobile code systems that are built that uses Java as their implementation language [12]. Mobile agents are a form of mobile code and this is the technology that we have embraced for our system. We will now explore in some details the basic concepts of mobile agent technology. We will also briefly look at two design paradigms that influenced the genesis of mobile agent technology: Remote Evaluation and Remote Procedure Call. In addition, we will look at the Java Agent Development Environment (Jade) which is our framework of choice for our agent development.

2.1. Mobile Agents

Mobile intelligent agents are autonomous movable code that are capable of migrating from one host to another carrying data and also its state to continue its execution [13]. Mobile agents usually have all or most of these properties: autonomy, mobility, adaptive/learning, goal oriented, active/proactive, collaborative/communicative [14]. As a result of these unique properties, mobile agent systems usually benefit from the following: reduction in network load, improvement in network latency, parallel processing, asynchronous execution, protocol encapsulation, dynamic adaptation, fault-tolerance and robustness.

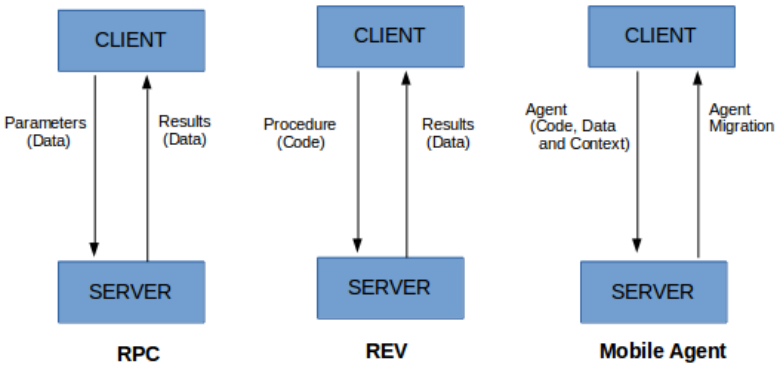

Fig. 1. Comparison of RPC, REV and mobile agents.

The development of mobile agent technology was influenced by concepts such as Remote Evaluation (REV) and Remote Procedure Calls (RPC) [1], [7]. However, it must be made clear that these concepts operate quite differently from mobile agent technology. Fig. 1 demonstrates the difference between mobile

agent technology, RPC and REV. In the RPC, the client sends data to the server, the server then processes this data and sends the result back to the client. In REV, the process is a little different in that the client this time sends code instead of data to the server, the server then uses this code to process data and then sends the result back to the client. Notice that in both cases, results in the form of data are sent back to the client. However mobile agent technology uses a totally different strategy. An agent which incorporates the code, data and context is sent from the client to the server. The agent will then carry out its function locally on that server. After the agent completes its tasks it then has options; it could terminate itself, go back to the client or migrate to another host where it can continue its job.

2.2. The Java Agent Development Kit: JADE

JADE (Java Agent DEvelopment Framework) is a free software framework that simplifies the development of multi-agent systems [15]. JADE is fully implemented in Java and complies with FIPA (FIPA is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies) specifications. It also provides graphical tools that can be used for deployment and debugging purposes. A Jade platform is made up of one or more agent containers which provides an environment where the agents can operate. There are two agents that are automatically created once you start a Jade platform, the Agent Management System (AMS) and the Directory Facilitator (DF). The AMS as the name suggests, is responsible for managing the entire agent platform while the DF is a Yellow Pages Service where agents can locate other agents. Fig. 2 shows an example of a scenario of a live agent platform consisting of two created containers along with the main container that is automatically created. The scenario also consists of four agents that were manually created along with the default AMS and DF.

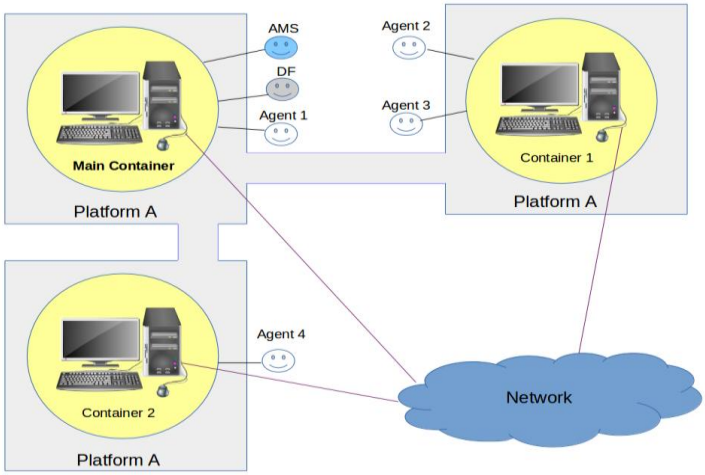

Fig. 2. Jade architecture.

3. Introduction to OptiPres

In order to place our experiment in context, we will now give a brief introduction to our mobile agent application. OptiPres is a decision support system that uses mobile intelligent agents to assist doctors in prescribing optimal drug therapy in a distributed system [16]. This idea was first introduced in this paper [16] and has since been through much refinement. We developed our system using JADE and the Android SDK. This system will be used by doctors on their android smart phones while prescribing medicines. It will assist them in making more informed decisions by either choosing the optimal drug(s) from processing a repository of past decisions (cases) or by presenting a set of possible drugs and using criteria specified by them to identify the optimal drug(s). To achieve this, we uniquely combined the Analytic Hierarchy Process

(AHP) and Cased-Based Reasoning (CBR) decision strategies and embedded this technique as behaviours in the personal mobile intelligent agent on the doctors' smart phones. This personal agent also has the ability to create additional helper agents. These agents will migrate to a basic Electronic Medical Record (EMR) system and a Drug Database to extract the appropriate information by executing local queries and performing local processing. Fig. 3 shows the architecture of OptiPres in a hospital setting while Fig. 4 and Fig. 5 show sample screenshots of the final results of a live simulation of OptiPres. The Case-based reasoner algorithm is always fired first. If there are no matching cases, then the AHP algorithm will be executed. This means, that only one of the results will be displayed to the user, depending on the availability of relevant data. In the simulation, we ran the scenario below with cases (Fig. 4) and then without any cases (Fig. 5), to demonstrate the flexibility of OptiPres. These results are in response to the following scenario:

Scenario-Woman, 22 years, 2 months pregnant. Large abscess on her right forearm. You conclude that she will need surgery fast, but in the meantime you want to relieve the pain. Your usual drug for common pain is acetylsalicylic acid (aspirin) tablets.

The results of OptiPres were compared with the recommendations from doctors to the above scenario. Our initial results show that OptiPres is as good as a doctor in prescribing appropriate drugs for patients.

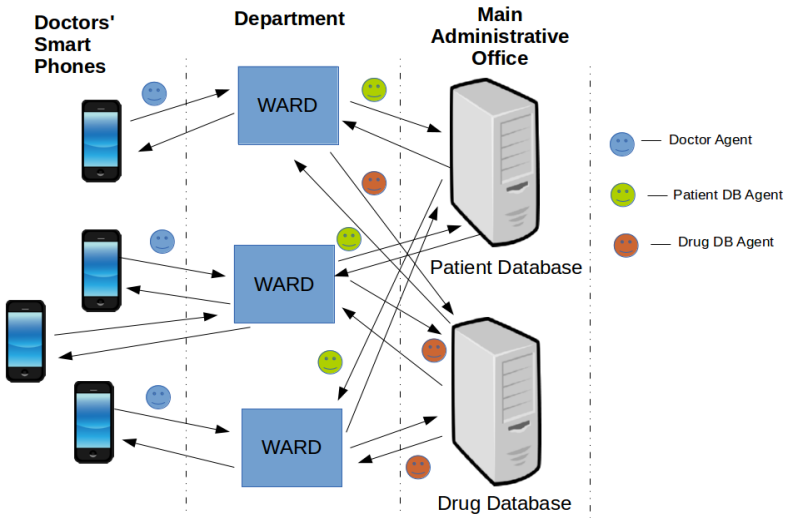

Fig. 3. Intelligent agent architecture.

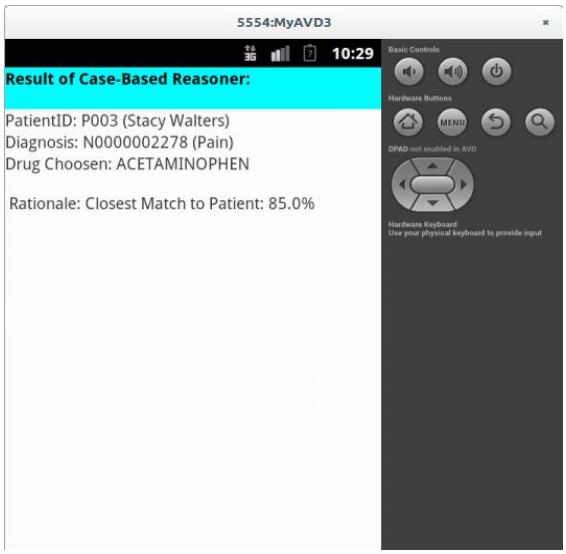

Fig. 4. Case-based result.

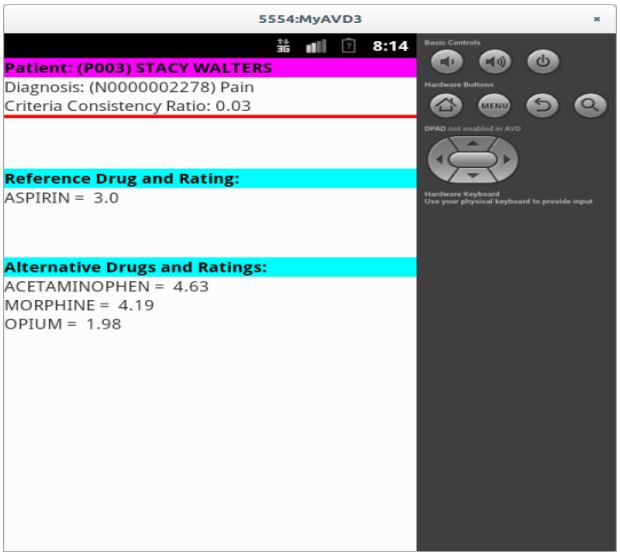

Fig. 5. AHP result.

4. Comparison of Mobile Intelligent Agents vs Client Server

The experiment was divided into two (2) sections:

1) We simulated the creation of multiple drug models using only 1 mobile agent (database agent) to access the database server and compare it against the traditional remote query client server approach. We also used another agent to manage the database agent. This agent is responsible for creating and destroying the database agent after it has done its task. This agent will also receive the drug models created by the database agent.
2) We simulated the creation of multiple drug models by dividing the task of creating drugs among multiple agents. This was compared against using only one agent to complete the given task.

The task in creating a list of drug models involves the agent(s) accessing multiple tables and filtering unwanted information while keeping only the pertinent information to assist in the decision making process. This data is then used to create a list of Java serialized objects that represents the selected drugs. The processing of drug models is ideal for these experiments as the agents will be performing this function frequently on thousands of drugs. The selected drug models will be needed on the ward workstations and therefore have to be fetched from the drug database server that is stored in a remote location. We also developed a python script that uses the library python-psutil to monitor the network utilization while the system is running. Gnuplot was then used to plot the graphs from the data generated by our python script.

4.1. Simulation Using One Dedicated Mobile Agent

In this section, we simulated the creation of one (1) agent to fetch and create models of 1,000, 5,000 and 10,000 drugs respectively and compare this with the traditional approach using the same number of drugs. Our results supported our predictions. The results are presented below:

Fig. 6(a) and Fig. 6(b) show the result of comparing the use of an agent against the traditional approach to fetch data and create 1000 drug models. Fig. 6(a) is using the traditional database access approach, while Fig. 6(b) is using a mobile intelligent agent. As predicted, in the image using the mobile agent, there is a short spike in data received which lasts for just about 3 seconds. This is where the models are being sent to the agent that created the drug mobile agent. Fig. 6(a) shows a huge hit in performance (Data sent and received) which lasted for the duration of the database access.

Fig. 7(a) and Fig. 7(b) show the result of creating 5,000 drug models. The pattern is basically the same as the previous experiment when 1,000 drug models were used.

Fig. 8(a) and Fig. 8(b) show the result of comparison to create 10,000 drug models and this is also showing the same pattern as our previous experiments.

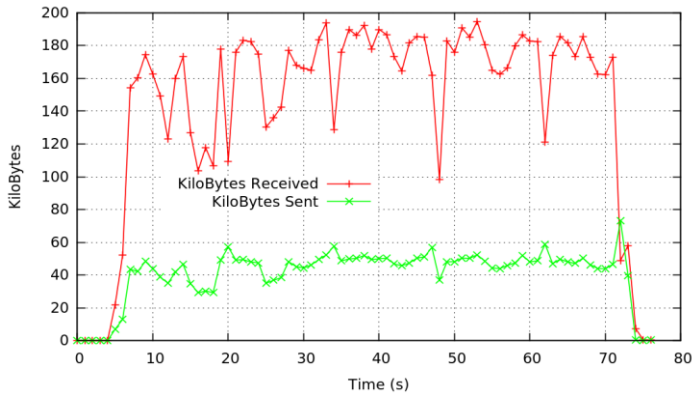

Fig. 6(a). Network performance using traditional database mobile access to create 1,000 drug models.

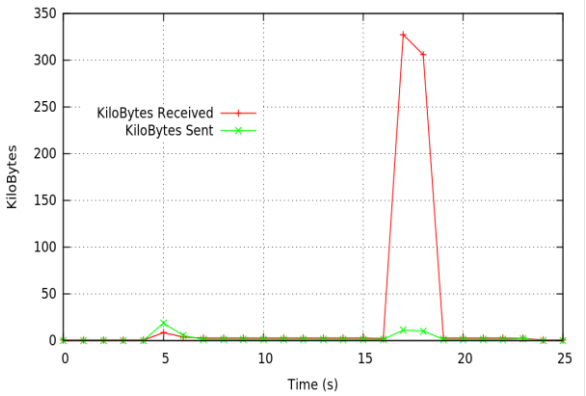

Fig. 6(b). Network performance using agent to create 1,000 drug models.

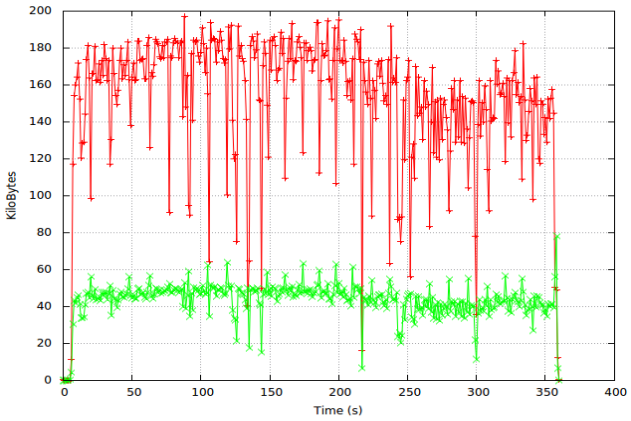

Fig. 7(a). Network performance using traditional database access to create 5,000 drug models.

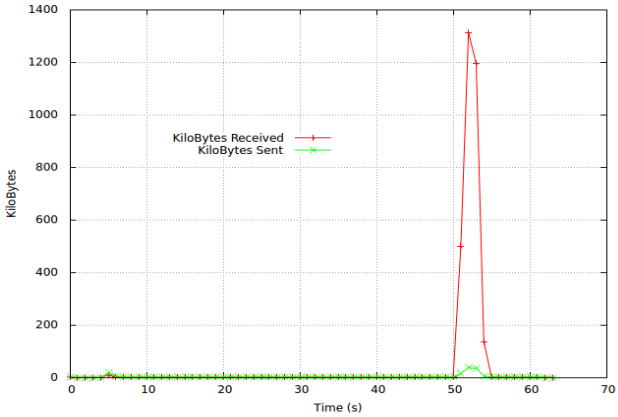

Fig. 7(b). Network performance using mobile agent to create 5,000 drug models.

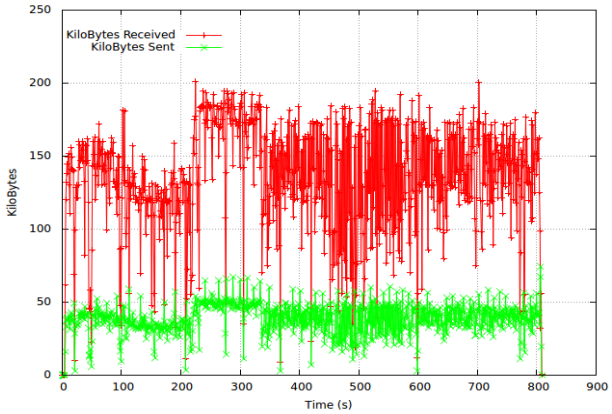

Fig. 8(a). Network performance using traditional agents database access to create 10,000 drug models.

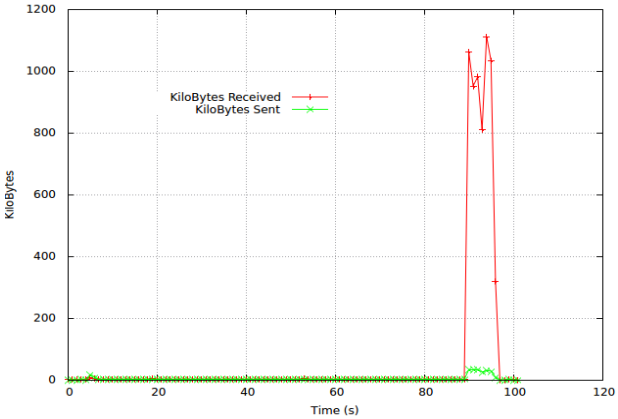

Fig. 8(b). Network performance using mobile to create 10,000 drug models.

4.2. Simulation Using Multiple Mobile Agents

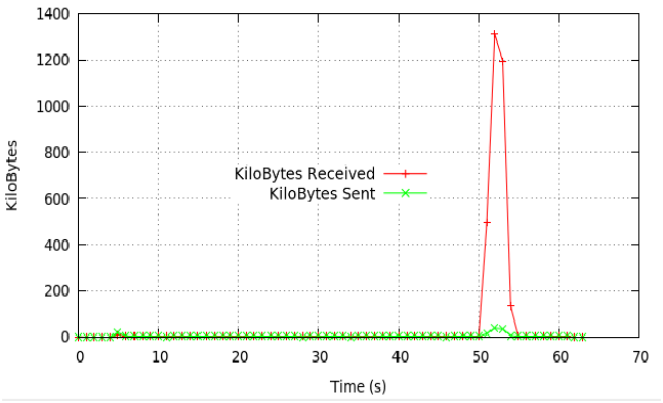

Fig. 9(a). Network performance using 1 mobile agent to Create 5,000 drug models.

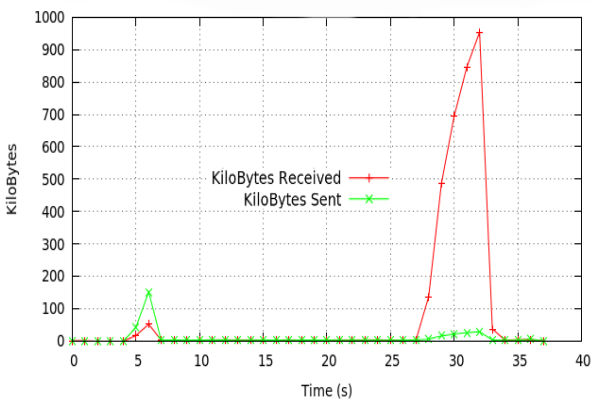

Fig. 9(b). Network performance using 10 mobile agents cooperating to create 5,000 rug models.

In this section, we improved the performance of our system by making the agents collaborate to perform the task of creating drug models. Therefore, we simulated the creation of 5,000, 10,000 and 100, 000 drug models using 10, 20 and 50 agents respectively. These agents will share the workload in an effort to complete the task at hand faster. We compare this with using 1 agent with the same number of drugs. Our results again supported our predictions. The results are presented below:

Fig. 9(a) and Fig. 9(b) show the result of comparing the use of 10 agents to cooperate and create 5,000 drug models to 1 agent doing the same. Fig. 9(a) is using 10 mobile intelligent agents, while figure is using only 1 agent. You should observe that using 10 agents shows basically the same trend as using one agent. However, the time to perform the task is shorter as the agents are now working in parallel. The task now takes 34 seconds instead of 55 seconds to complete.

Fig. 10(a) and Fig. 10(b) show the result of using 20 agents to collaborate and create 10,000 drug models vs using 1 mobile agent performing the same task. When using 1 agent, the task took 97 seconds to complete while using 20 agents took 57 seconds.

In Fig. 11(a) and Fig. 11(b) we went a bit further and created 100,000 drug models. When we used 1 agent, the time taken was approximately 800 seconds while using 50 agents to create 100,000 drug models took approximately 375 seconds. So in all cases, although the network utilization was approximately the same, we accomplished the task in a way shorter time.

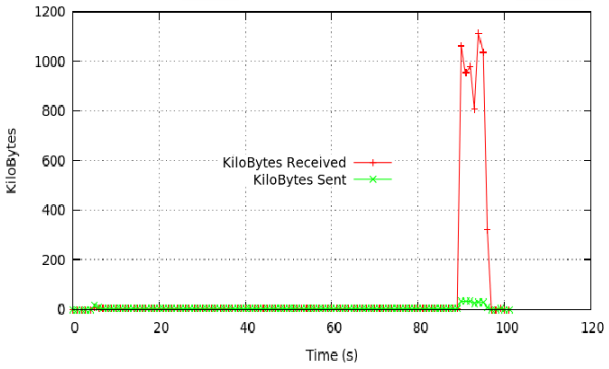

Fig. 10(a). Network performance using 1 mobile agent create 10,000 drug models.

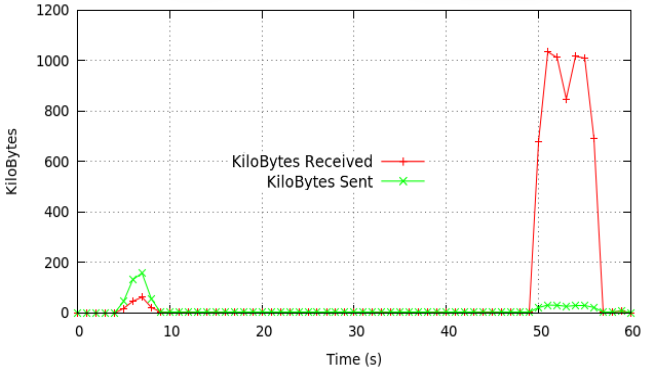

Fig. 10(b). Network performance using 20 mobile to agents cooperating to create 10,000 drug models.

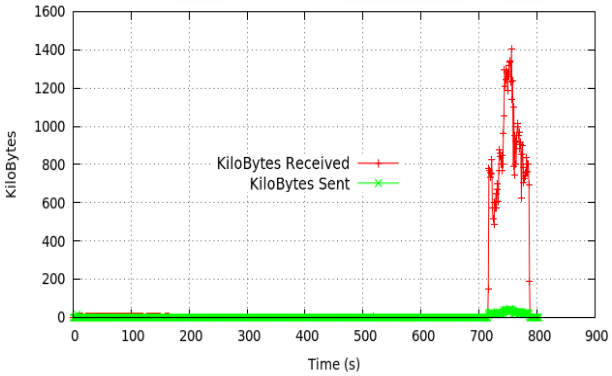

Fig. 11(a). Network performance using 1 mobile agent to create 100,000 drug models.

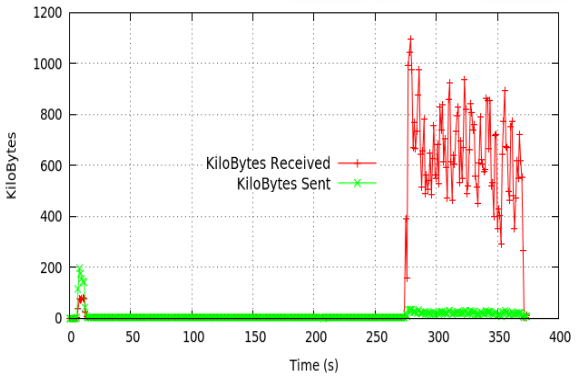

Fig .11(b). Network performance using 50 mobile agents cooperating to create 100,000 drug models.

5. Conclusion

In this paper, we started by theorizing that using mobile agents instead of client server approach should improve the performance of our application since it relies heavily on a database backend. Hence, we introduced OptiPres, which is a decision support system that we developed that depends heavily on the manipulation of drug data in databases. This application uses mobile intelligent agents as the middleware. We also discussed the concept on mobile agents and the proposed benefits that applications can gain. We then provided the results of our simulation in the form of graphs. As was predicted, using mobile intelligent

agents can greatly improve the performance of distributed applications. However, it does depend on the type of application that is being built as the use of mobile agents is not a "silver bullet" that will solve all the problems as it relates to distributed systems. The developers will need to assess their application needs and act accordingly. Additionally, the issue of security in mobile agent platforms is still a major concern as there is not yet any formal process to secure agents and the environment in which they execute. Nevertheless, our work demonstrated, using a real application, that the use of mobile agents to tackle performance issues with database application is very promising.

References

[1] Baldi, M., Gai, A., & Picco, G. P. (1997). Exploiting code mobility in decentralized and flexible network management. *Proceedings of the First International Workshop on Mobile Agents* (pp. 13-26). London UK: Springer-Verlag.

[2] Mane, K., & Pujari, V. (2013). Mobile agent for fast and secure services in client-server environment. *Proceedings of the 2013 Third International Conference on Advanced Computing and Communication Technologies* (pp. 171–173). Washington DC, USA: IEEE Computing Society.

[3] Lange, D., & Oshima, M. (1998). Introduction to mobile agents. *Personal and Ubiquitous Computing*, *2(2)*, 49–56.

[4] Miller, K., & Suresh, S. (2009). Monitoring patient health using policy based agents in wireless body sensor mesh networks. *Proceedings of World Congress on Nature & Biologically Inspired Computing* (pp. 503–508). Coimbatore: IEEE.

[5] Mateo, R. M., Cervantes, L. F., Yang, H. K., & Lee, J. (2007). Mobile agents using data mining for diagnosis support in ubiquitous healthcare. *Proceedings of the 1st KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications* (pp. 795-804). Berlin, Heidelberg: Springer-Verlag.

[6] Kirn, S. (2002). Ubiquitous healthcare: The OnkoNet mobile agents architecture. *Proceedings of Workshop on Mobile Computing in Medicine* (pp. 105-118). London UK: Springer Verlag.

[7] Bieszczad, A., Pagurek, B., & White, T. (1999). Network modeling for management applications using intelligent mobile agents. *Journal of Network and Systems Management, 7(3),* 295-321.

[8] Essa, Y., Attiya, G., & El-Sayed, A. (2013). Mobile agent based new framework for improving big data analysis*. Proceedings of International Conference on Cloud Computing and Big Data* (pp. 381-386). Fuzhou: IEEE.

[9] The Apache Software Foundation (2014). *Welcome to Apache Hadoop.* Retrieved September 22, 2014, from http://hadoop.apache.org/

[10] Nurain, N., Sarwar, H., Sajjad, M., & Mostakim, M. (2012). An in-depth study of map reduce in cloud environment. *International Conference on Advanced Computer Science Applications and Technologies* (pp. 263–268). Kuala Lumpur: IEEE.

[11] White, J. (1996). Mobile Agents White Paper. Retrieved July, 2014, from http://www.cis.upenn.edu/~bcpierce/courses/629/papers/White-Telescript.ps.gz

[12] Fuggetta, A., Picco, G. P., & Vigna, G. (1998). Understanding code mobility. *IEEE Transaction on Software Engineering*, *5(24),* 342–361.

[13] Lange, D. B., & Oshima, M. (1999). Seven good reasons for mobile agents. *Commun. ACM, 42(3),* 88–89.

[14] Franklin, S., & Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures and Languages* (pp. 21-35). London UK: Springer-Verlag.

[15] Telecom Italia. (2012). *Java Agent Development Framework*. Retrieved February 11, 2013, from http://jade.tilab.com/

[16] Miller, K., & Mansingh, G. (2013). Towards a distributed mobile agent decision support system for optimal patient drug prescription. *Proceedings of Third International Conference on Innovative Computing Technology* (pp. 233-238). London UK: IEEE.

Kevin Miller is a PhD candidate at the Department of Computing, University of the West Indies (U.W.I), Mona, Jamaica. He is also the data controller/system administrator for the department. Kevin obtained a master's degree in computer science with high commendation in his thesis from U.W.I in 2010. He also obtained a B.A. degree in computer science at U.W.I with first class honours which he received in 2006. While doing his undergraduate studies, he received the faculty award for academic performance for 2003-2004 and also the NCB Education Initiative scholarship for 2004-2006. Kevin currently has 6 publications to his credit which includes: three Conferences, two Journals and one Book chapter in the book entitled: "Digital Advances in Medicine, E-Health, and Communication Technologies". His research interests are mainly in networking, artificial intelligence, mobile intelligent agents and web technologies.

Gunjan Mansingh is a lecturer at the Department of Computing, the University of the West Indies (U.W.I), Mona, Jamaica. She obtained a PhD degree in information systems, an M.Phil. degree in computer science from U.W.I. and a B.Sc. degree from St Xavier's College, Bombay University, India. She teaches various courses at the undergraduate and the graduate level in computer science and information systems. She has conducted workshops in Jamaica on data mining and business intelligence. In 2013 she received a best research publication award and in 2014 she received a teaching award at her university. She is a co-editor of an edited book titled "Knowledge Management for Development: Domains, Strategies and Technologies for Developing Countries", Springer Integrated Series in Information Systems. Her research interests are in the following areas: business intelligence, data mining, decision support systems, knowledge management and knowledge management systems, expert systems and technology adoption. In her research she has worked in different domains in Jamaica such as healthcare, crime, agriculture and e-commerce. The research focus has been on harnessing data, information and knowledge in the various sources to assist in the decision making process. She has publications in refereed journals such as Decision Support Systems, Information Systems Frontier, Knowledge Management Research and Practice, Information Sciences, Expert Systems with Applications, International Journal of Metadata, Semantics and Ontologies and Health Systems. She also has papers published in the proceedings of IEEE Communications Conference Jamcon, e-business workshop, SIG GlobDev workshops, SIGDSS workshops, IRMA, Conf-IRM, AMCIS and IASTED.