International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:3, 2007

A Comparative Performance Evaluation Model of Mobile Agent Versus Remote Method Invocation for Information Retrieval

Yousry El-Gamal, Khalid El-Gazzar, and Magdy Saeb

***Abstract*—**The development of distributed systems has been affected by the need to accommodate an increasing degree of flexibility, adaptability, and autonomy. The Mobile Agent technology is emerging as an alternative to build a smart generation of highly distributed systems. In this work, we investigate the performance aspect of agent-based technologies for information retrieval. We present a comparative performance evaluation model of Mobile Agents versus Remote Method Invocation by means of an analytical approach. We demonstrate the effectiveness of mobile agents for dynamic code deployment and remote data processing by reducing total latency and at the same time producing minimum network traffic. We argue that exploiting agent-based technologies significantly enhances the performance of distributed systems in the domain of information retrieval.

***Keywords*—**Mobile Agent, performance evaluation, RMI, information retrieval, distributed systems, database.

I. Introduction

MOBILE agents are considered one of the most power all-embracing forms of code mobility. Mobile agents have not yet been well received by the internet community [1] since issues such as reliability and security are yet to receive developers' confidence. However, along with the wide spread of Java-based applications, mobile agents have become extensively popular not only in the research community but also in industrial projects [2]. One of the most attractive applications for mobile agents is the notion of "distributed information processing". This is particularly clear in the mobile computing scenarios where users have portable computing devices with only intermittent, low bandwidth connections to the main network. A mobile agent can abandon the portable device, move onto the network locations of the needed information resource and perform a locally custom-retrieval task. Only the results are transmitted back to a portable device [3]. Moreover, the mobile agent can carry on a task while the connection to the portable device is temporally lost and then continue once the link returns to send the found result.

Furthermore, mobile agents can exploit the high processing power available in the server machines by shifting the computations into the server side. In this work, we present a comparative performance evaluation model of Mobile Agents versus Remote Method Invocation by means of an analytical approach. In the following few lines, we discuss various approaches for information retrieval.

Static decision approach of choosing between mobile agents and client-server paradigms was discussed several times for a variety of applications. References [4]-[5]-[6] have comprehensively discussed this approach. The authors built their model based on, we believe, some impractical assumptions. Actually, this was reported by the authors themselves, to keep their model simple. Other authors have discussed the same approach by using applications from the network management domain [7]-[8]-[9]. Conversely, other researchers have proposed a mixed approach such as in reference [10]. In their work, they concluded that only a mixture between mobile agent migrations and remote procedure calls would produce minimal network traffic. The impact of various migration strategies of mobile agents on the overall performance was discussed in [11]. Minor work was found that discusses the influence of the wireless network transmission quality on the overall response time [12]. In this work, the authors have proposed a mathematical model to compare the response time between client-server and mobile agent approaches. The scalability of mobile agents was investigated, and the authors pointed out that the server scalability of mobile agent server software is a severe penalty for overall performance of mobile agent-based approaches [13] Using the experimental results approach, some authors have presented results of experiments with the "Aglets mobile agent system" and a client server implementation based on Java RMI [14]. Recently, comparison between mobile agent and RMI-based applications has attracted a growing attention. This is probably because mobile agents have a better fault tolerance when compared to the RMI [15]. Two important properties for fault-tolerant mobile agent execution: nonblocking and executing exactly-once were discussed in [16]. The mobile agent paradigm is considered a promising model for load balancing in general, and diffusion load-balancing techniques in particular [17]-[27]. The authors of reference [18] have presented a decentralized algorithm for dynamic load balancing based on the mobile agent paradigm.

Yousry El-Gamal is Minister of Education (e-mail: yelgamal@aast.edu).
K. El-Gazzar, and M. Saeb are with Computer Engineering Department, Arab Academy for Science, Technology & Maritime Transport, Alexandria, Egypt (e-mail: k_elgazzar@aast.edu, mail@magdysaeb.net).

An immense effort is put into mobile agent security issues [19]-[20]-[21]-[28]. More efforts were put in studying the performance of mobile agent platforms themselves [22]-[23]-[24]. The remainder of this paper is organized as follows: Section two presents the concept of agent migration. Section three describes the proposed comparative analytical model for performance comparison between mobile agents and RMI. Section four discusses results, and finally section five provides a summary and our conclusions.

II. The Agent Migration

Through the agent lifetime, an agent could migrate from an execution environment to another. The agent migration process consists of deactivating the agent, capturing its state, transporting the agent to a new location, restoring the agent state, and then resuming the agent execution. In this environment, different migration strategies can be employed. In the next two sub-sections, we illustrate these different migration strategies and mobility models.

A. The Migration Strategy

There are two main discerned categories to transfer the code and data to the destination platform in a mobile agent technology. These are "push" and "pull". The strategy used in migration has a great impact on the mobile agent performance. One of the pitfalls of the "push strategy" is that it drives classes that could not have been used in the next locations or could never have been used at all. On the other hand, a "pull strategy" requires a fast reliable retained connection or at least a fast way to reconnect to the agent source through the agent lifetime. All of these strategies can be classified [11] as follows:

Push-all-to-next

The code and all referenced objects are totally transferred to the next location

Push-all-to-all

The complete code of the agent are transmitted to all destination platforms the agent intending to visit so it needs all itinerary to be known in advance.

Pull-all-units

The agent only transmits the data and after the destination receives it starts to download all class files immediately when the first class file must be downloaded.

Pull-per-unit

After the destination receives the data, it tries to download the needed class file only.

B. The Mobility Models

Mobile agents consist of three main components: code segment, data state, and execution state. When they are all captured and transferred, we classify the mobility here as a strong migration. On the other hand, weak mobility is the ability to transfer code and data state only. There is no migration to the execution state in this case. Some believe that Java has been put forward as the preferred language for the development of mobile agent applications due to its nature in handling heterogeneous platforms. Unfortunately, java does not support strong migration. For example, it does not provide sufficient mechanisms for capturing the execution state of the agent [25]. However, during the last few years, many techniques were introduced to overcome this drawback and they are classified into two major categories. Those that use modified or custom Virtual Machines (VMs) and those that change the compilation model [26]. Nevertheless, each of these approaches has its pitfalls. In load-balance applications, it is required that the application be restored to the exact state before agent migration to be transparent to the application itself. Therefore, strong migration seems to be the rational choice. In other applications, where agents are considered to be efficient such as in information retrieval applications, mobile agents that support only weak mobility is considered to be sufficiently acceptable. This is primarily to avoid the high cost of strong mobility. The next section provides a model for this type of applications.

III. The Performance Model

In this section, we propose an analytical model that describes the network load and the response time in order to compare the performance of both the mobile agents and the Remote Method Invocation (RMI). RMI is an object equivalent of the classical client-server approach. In this research, we are only concerned with the parameters that can be useful in the comparative evaluation. For example, the number of requests that arrives to the server can affect the total processing time of the request. However, such effect when applied on both approaches will have no comparative impact. We are interested in identifying the parameters required for choosing a certain paradigm. Therefore, the influence of the server conditions and the network conditions will not be of value in the comparative evaluation. We consider a common application scenario as the foundation for the development of the proposed model. The scenario consists of a client that searches for a single data item located in one of the n servers. The task is terminated once the data item is found. We assume that the client begins to send a request B_{req} in bytes and the servers reply by B_{res} if the requested data are found otherwise, a reply B_{NF} is returned. The same process is repeated on the next server until the required data are fetched. On the other hand, the mobile agent approach visits sequentially the set of servers until it obtains the desired information. Fig. 1 clearly demonstrates this scenario.

The mobile agent compresses the data which are found at the server before transmitting it back to the client by a compression ratio σ, where $0 \leq \sigma \leq 1$. We assume that the mobile agent consists of code B_C, data state B_D, where B_D is the sum of the bytes of the result, and B_S is the execution state. The probability of finding data at server i is given by p_i, where $0 \leq p_i \leq 1$. The migration process consists of marshalling data and state, transmitting the code, data and state to the destination, unmarshalling data and state then,

resuming the agent execution. We assume that the time to marshalling and unmarshalling one byte is t_m. The time to process the request at the server is t_p. The time to transfer one byte from location L_1 to location L_2 over the direct link L_1-L_2 is $t_{B_{L1-L2}}$. That is, indirect routes are ignored. We emphasize that, each server will be visited only once. Therefore, there is no reusing of the same server again in one searching task. The load due to TCP header is ignored. There is no network queuing time. For sake of simplicity, we assume that the returned result has a constant size. We additionally assume a constant overhead scheduling time (t_s) in the case of mobile agents. That is, the agent action is considered as a heavy task. Finally, we neglect the authentication overhead. We consider that all of these assumptions have negligible effect on the comparative study suggested in this work.

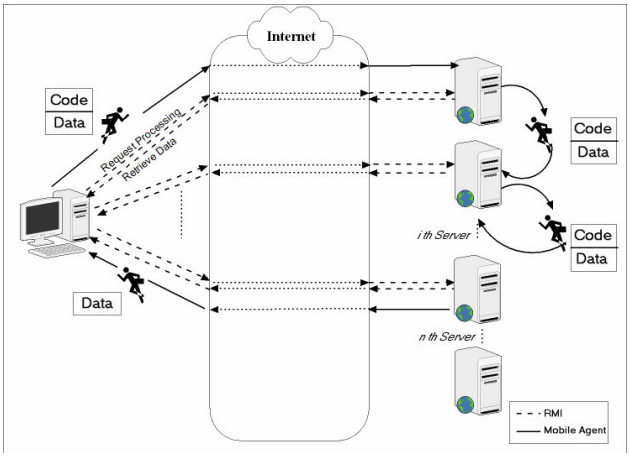

Fig. 1 Illustration of a graphical scenario of a client trying to search for data in a set of web servers

A. The Network Load

We start by analyzing the RMI approach. The client sends a request to invoke a method placed on the server. This method searches locally for the data and replies to the invoker by the found data result or returns a NOT-FOUND reply. The client repeats the search process in the next server until the data item is found. Hence, the network load affected by this approach (B_{CS}) is calculated by:

$$\begin{aligned} B_{cs} &= p_1(B_{req}+B_{res})+ \\ &\quad p_2(1-p_1)(2B_{req}+B_{NF}+B_{res})+... \\ &\quad p_n(1-p_1)(1-p_2)...(1-p_{n-1})(nB_{req}+(n-1)B_{NF}+B_{res}) \end{aligned}$$

That is,

$$B_{cs}=\sum_{i=1}^{n} p_i \prod_{j=1}^{i-1}(1-p_j)(iB_{req}+(i-1)B_{NF}+B_{res}) \quad (1)$$

For the ease of modeling, we assume that all servers have the same probability (p) of finding data item. Then, equation 1 can be rewritten as:

$$B_{cs}=np\sum_{i=1}^{n}(1-p)^{i-1}(iB_{req}+(i-1)B_{NF}+B_{res}) \quad (2)$$

Now, we investigate the mobile agent approach. For sake of simplicity, we additionally assume that when the agent migrates to a new location it carries all its code, data, and all state information by using the "push all-to-next" migration strategy. However, while the agent migrates back to home, it carries only the data state. B_{MA} is the total network traffic, in bytes, caused by this paradigm. This network load is calculated by:

$$\begin{aligned} B_{MA} &= p_1(B_C+2B_S+B_{req}+(1-\sigma)B_{res})+ \\ &\quad p_2(1-p_1)(2B_C+3B_S+2B_{req}+(1-\sigma)B_{res})+... \\ &\quad p_n(1-p_1)(1-p_2)...(1-p_{n-1}) \\ &\quad (nB_C+(n+1)B_S+nB_{req}+(1-\sigma)B_{res}) \end{aligned}$$

That is,

$$B_{MA}=\sum_{i=1}^{n} p_i \prod_{j=1}^{i-1}(1-p_j)(iB_C+(i+1)B_S+iB_{req}+(1-\sigma)B_{res}) \quad (3)$$

Again, if we assume that all the servers have the same probability (p) of finding data item. Then equation 3 can be written as:

$$B_{MA}=np\sum_{i=1}^{n}(1-p)^{i-1}(iB_C+(i+1)B_S+iB_{req}+(1-\sigma)B_{res}) \quad (4)$$

Considering the case of handheld devices or as an example a PDA, we are concerned only with the traffic cost from the client side such as GPRS connection to the internet. The traffic by the mobile agent paradigm is calculated by:

$$B_{MA_{GPRS}}=B_C+2B_S+B_{req}+(1-\sigma)B_{res} \quad (5)$$

This load remains constant whatever the number of the visited locations. In addition, the impact of client wireless connection reliability is considered negligible in the case of using mobile agent approach. Therefore, the use of the mobile agent approach reduces the network traffic by the client side to a minimum. Whereas, the network traffic that is caused by the RMI approach remains unchanged from that one calculated above by equation 2.

B. The Response Time

We measure response time in seconds. Regarding the RMI approach this response time is calculated by:

$$\begin{aligned} t_{cs} &= p_1((B_{req}+B_{res})t_{B_{01}}+t_p)+ \\ &\quad p_2(1-p_1)((B_{req}+B_{NF})t_{B_{01}}+(B_{req}+B_{res})t_{B_{02}}+2t_p)+... \\ &\quad p_n(1-p_1)(1-p_2)...(1-p_{n-1}) \\ &\quad ((B_{req}+B_{NF})(t_{B_{01}}+t_{B_{02}}+...+t_{B_{0(n-1)}})+(B_{req}+B_{res})t_{B_{0n}}+nt_p) \end{aligned}$$

That is,

$$t_{cs}=\sum_{i=1}^{n} p_i \prod_{j=1}^{i-1}(1-p_j)((B_{req}+B_{NF})\sum_{j=1}^{i-1}(t_{B_{0j}})+t_{B_{0i}}(B_{req}+B_{res})+it_p) \quad (6)$$

Applying the same approach in simplifying equations 3, 4, and suppose that the network have the same delay per byte over all the links. Then equation 6 can be simplified to be as shown next.

$$t_{cs} = np\sum_{i=1}^{n}(1-p)^{i-1}(t_B(B_{res}+iB_{req}+(i-1)B_{NF})+it_p) \quad (7)$$

The corresponding response time in the mobile agent approach is calculated by:

$$\begin{aligned}
&\text{hint}: B_{ma} = B_C + B_S + B_{req}\\
&t_{MA} = p_1(B_{ma}(t_{B_{01}}+t_m)+(B_S+B_{res})(t_{B_{01}}+t_m)+t_S+t_p)+\\
&\quad p_2(1-p_1)\\
&\quad (B_{ma}(t_{B_{01}}+t_m)+B_{ma}(t_{B_{12}}+t_m)+(B_S+(1-\sigma)B_{res})(t_{B_{02}}+t_m)+2t_S+2t_p)+\ldots\\
&\quad p_n(1-p_1)(1-p_2)\ldots(1-p_{n-1})\\
&\quad (B_{ma}(nt_m+t_{B_{01}}+t_{B_{12}}+\ldots+t_{B_{(n-1)n}})+(B_S+(1-\sigma)B_{res})(t_{B_{0n}}+t_m)+nt_S+nt_p)\\
&t_{MA} = \sum_{i=1}^{n}p_i\prod_{j=1}^{i-1}(1-p_j)(B_{ma}(nt_m+\sum_{j=1}^{i}(t_{B_{(j-1)j}})+(B_S+(1-\sigma)B_{res})(t_{B_{0i}}+t_m)+it_S+it_p)
\end{aligned} \quad (8)$$

Simplifying this equation, using the above simplification assumptions leads us to:

$$t_{MA} = np\sum_{i=1}^{n}(1-p)^{i-1}((iB_C+iB_{req}+(i+1)B_S+(1-\sigma)B_{res})(t_B+t_m)+it_S+it_p) \quad (9)$$

Using the calculated network load from equation 3 simplifies the response time equation 9 as:

$$t_{MA} = np(B_{MA}(t_B+t_m)+\frac{n(n+1)}{2}(t_S+t_p)) \quad (10)$$

IV. Performance Evaluation

In this section, we provide a discussion of the proposed model results. Furthermore, we determine the break-even points between the two paradigms. These break-even points determine which paradigm performs better with a given set of parameters.

A. Evaluation Criteria

In the proposed comparative model, we choose to use two quantifiable measurable quantities as evaluating metrics; the response time and the network load. These two metrics are employed to compare the performance of the RMI-based paradigm versus the mobile agent-based paradigm. The objective is to decide mathematically which paradigm produces less network traffic in order to provide most rapid response.

B. Results and Discussion

To evaluate the performance of the RMI and mobile agent paradigms, we have developed an implementation of the proposed method for both paradigms using a customized version of IBM Aglets to verify and validate the model results. Different scenarios are considered using both paradigms. The Agent consists of B_C = 2 Kbytes. Weak Migration is considered that is suitable for information retrieval applications, and so B_S = zero. The probability of finding data at servers is considered to be unknown, and can be guessed as p = 0.5. B_{NF} = 20 bytes, σ is varied from zero to one. The time to process the request is simulated by a latency t_p equal to 0.002 second. Finally, the network throughput = 400 byte/sec. Fig. 2 shows the result of comparison between the RMI and the mobile agent approaches using the "Expected Network Load" versus "Result Size". It clearly shows that mobile agents performs better when the result size is more than 6 k byte, and using a compression ratio σ equal to 0.7, when ten locations are visited. Whereas, the slightly improvement taken from the RMI in smaller server result sizes is negligible. Fig. 3 illustrates the network load produced by each paradigm against the number of servers. The break-even point moves forward and backward according to the changes in the mobile agent code size, and the positive effect of compression ratio. Fig. 3 shows the impact of the number of servers on the network load. It can be seen that only for small number of servers, mobile agent produce less network traffic. However, for large server results, mobile agents always produce less network traffic due to the positive effect of the compression of the server result. Fig. 4 illustrates the network load considering the client side only, it is obvious that mobile agents produce minimal network load. Moreover, it is independent from the location of found information.

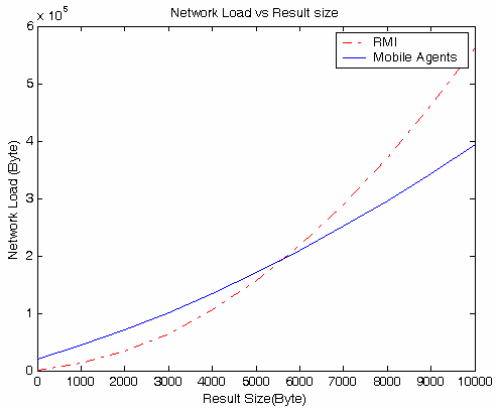

Fig. 2 Network load versus server result size for fixed compression ration σ. Mobile agent produce smaller network load while σ increases

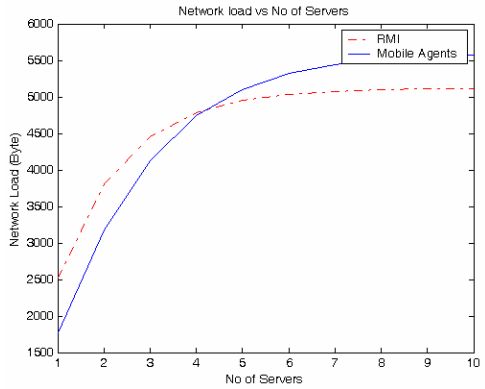

Fig. 3 Network load versus no of servers. Mobile agent always responds fast

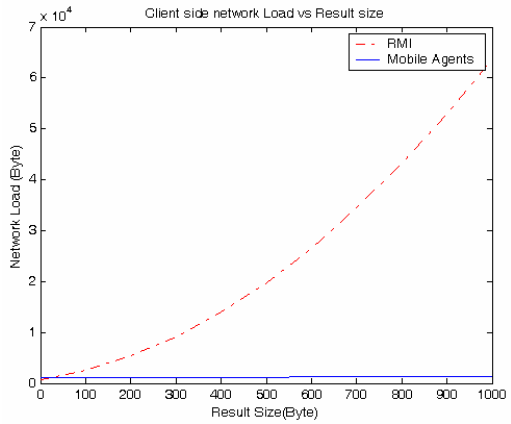

Fig. 4 Network load at the client side in the case of handheld devices

Figs. 5 and 6 show the corresponding diagrams for the response time. These diagrams illustrate the influence of the result size and number of visited locations respectively on the overall response time. The most notable feature from these graphs is the fast response time of the mobile agents for large results. However, for example if the data rate is high enough, mobile agents of large code size may not think to be the best paradigm. Mobile agents suffer from performance degradation when the number of search locations is increased.

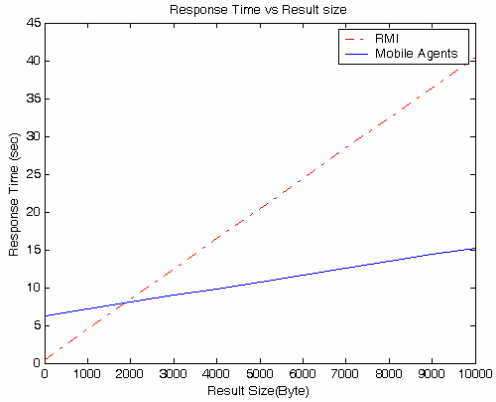

Fig. 5 Response time versus result size. Clearly Mobile agent has fast response and smaller latency comparing to the RMI

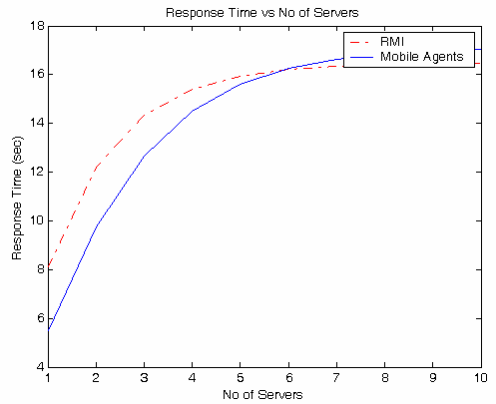

Fig. 6 Response time versus No of servers

V. Summary and Conclusion

In this exertion, we have demonstrated the following:

- A performance model that is based on rather more realistic assumptions.
- Employing this model, the RMI and the mobile agent paradigms were compared based on network load and the response time.
- A breakeven point was determined that can benefit the system developer to recognize and choose the correct paradigm to be utilized for a given application and an environment.
- The model was verified employing a customized version of Aglets for both paradigms.

Based on our results, one can conclude that the mobile agent is a promising paradigm for the design of information retrieval systems in a distributed environment. However, one should always consider that there is a paradigm suitable for a given application running in a specific environment.

In this article, we have proposed a mathematical model to compare the performance of the two code mobility extremes: mobile agents and client-server paradigms. The mobile agent paradigm performs better if the code size is small enough and the compression ratio for the server results is acceptable. Nevertheless, given the expected wide spread use of handheld devices, mobile agents produce less client connection cost and supports unreliable wireless connection by continuing execution. This takes place when the client connection is temporarily lost.

We argue that the proposed model in this work can be instrumental in the creation of more mature applications. It enables formal reasoning and verification of the selected design decision. We are continuing to extend this model to support different migration strategies to produce less network traffic, and achieve lesser response time.

References

[1] Giovanni Vigna, " Mobile Agents: Ten Reasons for Failure," 2004 IEEE International Conference on Mobile Data Management (MDM'04), pp. 298, January, 2004.

[2] Christian Erfurth, Peter Braun, Wilhelm Rossak, "Some Thoughts on Migration Intelligence for Mobile Agents," Technical Report 01/09, Friedrich-Schiller-University¨, Jena, April, 2001.

[3] Robert S. Gray, David Kotz, and Ronald A. Peterson, Jr., "Mobile-Agent versus Client/Server Performance: Scalability in an Information-Retrieval Task," Proceedings of the Fifth IEEE International Conference on Mobile Agents, pp. 229-243, Atlanta, Georgia, December, 2001.

[4] Antonio Carzaniga, Gian Pietro Picco, and Giovanni Vigna, "Designing Distributed Applications with A Mobile Code Paradigm," Proceedings of the 19th International Conference on Software Engineering ICSE97, Seattle USA, pp. 22–32, April, 1997.

[5] Giovanni Vigna, "Mobile Code Technologies, Paradigms, and Applications," PhD thesis, Politecnico di Milano, February, 1998.

[6] Antonio Puliafito, Salvatore Riccobene, and Marco Scarpa, "Which paradigm should I use? An analytical comparison of the client-server, remote evaluation and mobile agent paradigms," Concurrency and Computation: Practice and Experience, Vol. 13, No. 1, pp. 71–94, 2001.

[7] Mario Baldi, Silvano Gai, and Gian Pietro Picco, "Exploiting code mobility in decentralized and flexible network management," Proceedings of the First International Workshop on Mobile Agents MA97, Berlin (Germany), April, 1997.

[8] Gian Pietro Picco, "Understanding, Evaluating, Formalizing, and Exploiting Code Mobility," PhD thesis, Politecnico di Torino, Italy, February, 1998.
[9] Mario Baldi and Gian Pietro Picco, "Evaluating the tradeoffs of mobile code design paradigms in network management applications," Proceedings of the 20th International Conference on Software EngineeringICSE98, Kyoto, Japan, pp.146-155, April, 1998.
[10] Markus Straßer and Markus Schwehm, "A performance model for mobile agent systems," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA97, Las Vegas, USA, Vol. 2, pp. 1132–1140, Athens, GA, 1997.
[11] Peter Braun, Christian Erfurth, Wilhelm Rossak, "Performance Evaluation of various Migration strategies for mobile agents," Kommunikation in Verteilten Systemen, pp. 315-324, 2001.
[12] Ravi Jain, Farooq Anjum, and Amjad Umar, "A comparison of mobile agent and client-server paradigms for information retrieval tasks in virtual enterprises," Proceedings of the Academia/Industry Working Conference on Research Challenges AIWORC00, Buffalo, NY, USA, Los Alamitos, April, 2000.
[13] Robert S. Gray, David Kotz, Ronald A. Peterson, Joyce Barton, Daria A. Chac´on, Peter Gerken, Martin O. Hofmann, Jeffrey Bradshaw, Maggie R. Breedy, Renia Jeffers, and Niranjan Suri, "Mobile-agent versus client/server performance: Scalability in an information-retrieval task," Mobile Agents, Proceedings of the 5th International Conference MA2001, Atlanta USA, December, 2001.
[14] Daniel Hagimont, L. Ismail, "A Performance Evaluation of the Mobile Agent Paradigm," ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99, Vol. 34, No. 10, October, 1999.
[15] G.A. Aderounumu, B.O. Oyatokun, M.O. Adigum, "Remote Method invocation and Mobile Agent: A Comparative Analysis," The Journal of Issues in Informing Science and Information Technology, Vol. 3, 2006.
[16] Pleisch. S., and Schiper, A., "fault-tolerant mobile agent execution," IEEE Transactions on Computers, Vol. 52, No. 2, pp. 209-222, 2003.
[17] H. Farooq Ahmad, Hiroki Suguri, "Dynamic Information Allocation through Mobile Agents to Achieve Load Balancing in Evolving Environment," The 6th International Symposium on Autonomous Decentralized Systems (ISADS'03), pp. 25-33, April, 2003.
[18] Magdy Saeb, Cherine Fathy**,** "Performance Evaluation of Mobile Agent-based Dynamic Load Balancing Algorithm**,"** 9th International Conference on Distributed Multimedia Systems, DMS_Conference, Miami, Florida, USA, 2003.
[19] Sheng Zhong, Yang Richard Yang, "Verifiable distributed oblivious transfer and mobile agent security," Mobile Networks and Applications, Vol 11, No. 2, April, 2006.
[20] Adam Pridgen, Christine Julien, "Self-organization and security: A secure modular mobile agent system," Proceedings of the 2006 international workshop on Software engineering for large-scale multi-agent systems SELMAS '06, May 2006.
[21] Sergio Ilarri, Raquel Trillo, Eduardo Mena," SPRINGS: A Scalable Platform for Highly Mobile Agents in Distributed Computing Environments," Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks WOWMOM '06, June, 2006.
[22] Josef Altmann, Franz Gruber, Ludwig Klug, Wolfgang Stockner, and Edgar Weippl, "Using mobile agents in real world: A survey and evaluation of agent platforms," Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS at the 5th International Conference on Autonomous Agents, pp. 33-39, Montreal, Canada, ACM, ACM Press, June, 2001.
[23] U. Pinsdorf, V. Roth, "Mobile Agent Interoperability Patterns and Practice," Proceedings of Nonth IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS), Lund, Sweden, April, 2002.
[24] Rahul Jha and Sridhar Iyer, "Performance evaluation of mobile agents for e-commerce applications," International Conference on High Performance Computing (HiPC), Hyderabad, India, December, 2001.
[25] Rafael Fernandes Lopes, Francisco Silva, "Migration Transparency in a Mobile Agent Based Computational Grid," Proceedings of the 5th WSEAS Int. Conf. on Simulation, Modeling and Optimization, pp.31-36, Corfu, Greece, August, 2005.
[26] Arjav J. Chakravarti, Xiaojin Wang Jason O. Hallstorm, Gerald Baumgartner, "Implementation of Strong Mobility for Multi-Threaded Agents in Java," International Conference on Parallel Processing (ICPP'03), p321, 2003.
[27] Magdy Saeb, Cherine Fathy, "A modified Diffusion Load Balance Algorithm Employing Mobile Agents," WSEAS International Conference on Circuit, Systems & Computers, Corfu, Greece, July, 2003.
[28] Magdy Saeb, Meer Hamza, Ashraf Soliman, "Protecting Mobile Agents against Malicious Host Attacks Using Threat Diagnostic AND/OR Tree," Smart Objects Conference, SOC2003, Grenoble, France, May, 2003.