Journal of Theoretical and Applied Information Technology
31st January 2020. Vol.98. No 02

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MATHEMATICAL MODEL FOR COMPARING PERFORMANCE EVALUATION OF MOBILE AGENT PLATFORMS

[1]FAIZ ALSHROUF, [2]SHADI R MASADEH, [3]FAISAL ALZYOUD

[1] Associate Professor, Isra University, Department of Computer Science, Jordan

[2] Associate Professor, Isra University, Department of Cyber Security, Jordan

[3]Assistant Professor, Isra University, Department of Computer Science, Jordan
E-mail: [1]Fayez.shrouf@iu.edu.jo, [2]Shadi.almasadeh@iu.edu.jo, [3]Faisal.alzyoud@iu.edu.jo

ABSTRACT

A The paradigm of Mobile Agents is emerging in the field of distributed computing. Mobile agents present features, such as autonomy and capability to roam to hosts, process data, and save remote communications. Many mobile agent platforms have been developed for research purposes while other platforms have been deployed as commercial products. Several Java-based platforms have been already implemented and many researchers is now starting to implement some based applications. In most of the cases, the performance of the applications is an issue of paramount importance. A common problem when one wants to benefit from mobile agent platform is the decision about which platform to use. In fact performance evaluation is exploited to address different issues according to which, different measurements are required. In practice, there is a need for tools and techniques for evaluation of the performances of the adopted mobile agent platforms. Related work proposes a set of performance metrics, slightly different one from the other, and measure different approaches. In this research, a mathematical model is presented that evaluates performance evaluation of different agent platforms and agents interested behavior. The study suggests set of agent platforms to evaluate and finally comparing performance amongst them.

Keywords *Mobile Agents; Performance Evaluation; Distributed Applications; Java; Mathematical Model.*

1. INTRODUCTION

Many existing applications in management of telecommunication are using the centralized approach, where every element in the network sends data to a central location. This approach has some relevant drawbacks: it is not flexible, it is not scalable, and produces too much traffic in the network. The use of mobile agents potentially solves most of the problems and provides applications that can be more scalable, more robust, and can be upgraded easily with other applications.

The use of mobile agents has received attention from several research institutions, and various effort is given to standardize agent's technology, as FIPA and MASIF standards. Commercial companies started to launch and developed several commercial mobile agent systems to implement in different areas, like e-commerce, Internet domain, network management, and telecommunications. Several mobile agent systems have been developed for different purposes. These include: [JAMES], [Swarm] from Siemens [1] [2], [Voyager] from Objectspace[3], [Aglets] from IBM [4], [Concordia] from Mitsubishi [5], [Odyssey] from General Magic [6], [JumpingBeans] from AdAstra [7], [SPRINGS] from Zaragosa [7]. [Grassshopper] from IKV [8], [JADE] from Telecom [9], [Tracy] from Jena [10].

Performance evaluation of mobile agent systems focuses on two main issues: first, comparing mobile agent platforms from quantitative and qualitative domain [11], and second, comparing mobile agents' platforms performance. In this paper, we focus on conducting analysis of performance evaluation of mobile agent's platforms in terms of platform execution time for a predefined agent code size.

However, some questions have been made by this research: what is the performance of the existing mobile agents platforms? What type of mathematical approach could be used to conduct mobile agent's performance? What are the benefits of the approach that have been introduced in this

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

evaluation? How platforms are compared with the other ones?

To answer these questions, we have studied an experimental approach of benchmarking given by [1]. The study presents some results for eight Java-based mobile agent systems and reached some conclusions about its run-time behavior. All the platforms in the mentioned study have used a methodology of the benchmarking: under certain conditions. The platforms have been tested in an environment of 2 MHz processor, 2Mb RAM, and Windows NT 4.0, JDK 1.6.1 with JIT options. Furthermore, all the platforms have used small agent i.e. 100Kb, and 1Mb. We conducted this research for three additional platforms including JADE, SPRINGS, and Tracy, under the same environmental conditions. The research scope also included a mathematical model that generalized the results of platform execution time for large agent code size including: medium agent code size 2 MB, large agent code size 10MB, and very large agent code size 1 GB. The approach focuses mainly on predicting mobile agent's execution time and comparing performance evaluation of mobile agent's platforms.

2. RELATED WORK

Many authors presented a comparative study of mobile agent's platforms. [7] conducted tests to study the performance executing a parallel algorithm using 100 agents (with no calls among them). They used SPRINGS, Voyager, and Aglets to perform the test. They concluded that SPRINGS gives better performance over other platforms. Another test is conducted which uses 100 agents (with calls among them). They use JADE, Voyager, Springs, Aglets, and Grasshopper). They concluded that JADE gives better performance over other platforms. [2], they presented experimental study of eight Java-based mobile agent platforms namely (JAMES, JAMES (pref.), Aglets, Voyager, Odyssey, Jumping Beans, Grasshopper, and Swarm). They conducted two main tests, first test, they suggest number of agencies:1, No cache, and agent size is 100 KB, second test, they suggest number of agencies:1, No cache, and agent size is 1 Mb). They concluded that JAMES and JAMES (pref.) give the best performance over other mobile agent's platforms. [12], they conducted a study of comparing the performance of two mobile agent's platforms Aglets and TACOMA in distributed search BFS and DFS. Results of comparisons indicated that the behavior is quite similar but the performance when using TACOMA is better than Aglets in case of multiple agents running in (MANET). [13] they presented a linear prediction model to analyze the execution of benchmarks for performance evaluation of mobile agents based systems. The analysis is conducted by using different platforms. The results show better performance of JAMES comparing to other agent systems in run-time execution of migration and communication facilities. [14], they presented a methodology for evaluating the performance of seven mobile agent's platforms: JADE, Aglets, Jumping Beans, TACOMA, Swarm, Concordia, and Tryllian. The study uses four metrics: availability, environment, development and characteristics. The evaluation indicated that the performance of JADE has the best performance over other platforms. [15], they presented performance evaluation for mobile agent systems including: SPRINGS, JADE, Voyager, Grasshopper, Aglets, and JADE. Results indicated that the performance is better when implementing SPRINGS, Voyager, and Grasshopper for small number of agents, and performance becomes better when implementing JADE, Aglets, and Voyager for large number of agents. [16], they presented performance evaluation of three multi agent systems in terms of platform design. They concluded that the internal design of multi agent platform affect its performance. The experiments performed are focused on features involved in agent communication. [11], they presented comparisons and evaluation performance criteria of software agents platforms for e-commerce. The study evaluates qualitative and quantitative criteria parameters for Aglets, Concordia, Voyager, and JADE.

3. OVERVIEW OF MOBILE AGENT PLATFORMS

In this section, we present an up-to-date analysis of mobile agent's platforms: JAMES, Odyssey, Swarm, Grasshopper, Aglets, Voyager, JADE, Concordia, Tracy, Jumping Beans, and SPRINGS. We present set of features and characteristics of each platform.

3.1 JAMES

JAMES is a project developed by University of Coimbra (Portugal) in cooperation with Siemens. The JAMES platform is mainly oriented for applications in the area of Telecommunication and Network Management. JAMES is implemented in Java and integrated with CORBA standards. It is high performance, secured, robustness, flexible distribution of the agent's code, a code prefetching

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

scheme, a pool of threads and migration channels and protocol enhancements. JAMES has been enhanced with comprehensive support for fault tolerance, resource control. Furthermore, the architecture of JAMES allows any modeling and simulation technique to be integrated into the framework via plug-ins. Moreover, it provides a solid foundation of abstractions, algorithms, workflows and tools, focusing on efficiency. In this context, JAMES has high performance, scalability, and robustness. Another feature added to JAMES is that it provides hundreds of plug-ins, allowing automatic selection from the available list of alternative plug-ins.

3.2 Odyssey

Odyssey is a Java-based mobile agent system from General Magic. The Platform has a transport independent API that work with RMI, and DCOM objects. It provides Java agent's classes which support good functionality for roaming and migration of mobile code over a network, communication in Odyssey includes synchronous messaging scheme, while asynchronous messaging is not supported by the platform. Odyssey has good performance, scalability, and robustness. More details about Odyssey can be founded at; http://www.genmagic/com/technlogy/Odyssey.html

3.3 Swarm

Swarm is a platform is being developed by Siemens, from the University of Stuttgart, Germany. Swarm is being developed mainly and used by ACTS, AMASE project is used to provide middleware components in wireless networks. Swarm provides extensive features for inter agent communication scheme, but the platform has limited and complicated GUI feature. Furthermore, Swarm was originally developed for multi agent simulation of complex adaptive systems. Swarm implements mobile software agents to develop multiple robots using formation control algorithms. Swarm has an average performance and scalability and low robustness.

3.4 Grasshopper

Grasshopper was developed by IKV++ (version 2.2.4, 2003), it is a mobile agent platform that has been designed in conformance with MASIF and FIPA standards. It is distributed commercially by Enago Mobile. The platform is implemented in Java, it supports several protocols by the use of internal ORB, and provides GUI for managing mobile agents, agencies, regions. Furthermore, the platform supports security, agent's communication scheme and agent's persistency. Grasshopper has an average performance, scalability, and robustness.

3.5 Jumping Beans

Jumping Beans is a mobile agent platform from Ad Astra Engineering. It based on JavaBeans that jump from computer to computer during execution. The beans are actually components or objects. The platform is built on API and developers can use objects to add mobility to their projects. Jumping Beans has features including: enforcing security, agent management, easy integration with other existing environment, and crash recovery when needed. The platform has high security level, but it is low level in performance (i.e. if an agent wants to migrate between two agencies, it has to go first to the Agent Manager. More details can be found in http://www.JumpingBeans.com.

3.6 Aglets

Aglets is Java-based platform in which agents roam from one host to another. Aglets is developed by IBM, Tokyo in 1996, and maintained by open source community since 2001. The migration of Aglets is based on proprietary protocol called Agent Transfer Protocol (ATP) and controlled by Tahiti server, which controlling creation, cloning, disposing, and dispatching Aglets. Aglets features including: user friendly GUI, supporting synchronous and asynchronous messaging scheme, and widely used in the development of distributed computing systems. An important disadvantage of the platform is that using of the proxies (i.e. a proxy cannot be used after the agent moves to another place), therefore, the programmer must obtain and updated proxy. Furthermore, the developer must avoid the execution of long-running jobs; otherwise this would prevent agent incoming messages from being considered. This scheme could lead to deadlock problem if two agents send synchronous messages at the same time. Aglet has average performance, scalable, robustness, and security.

3.7 Voyager

Voyager platform was developed by ObjectSpace and currently by Recursion Software (last version Voyager Edge 6.0.1) is a distributed computing middleware focused on simplifying the management of remote communication CORBA and RMI protocols. It offers dynamic generation of CORBA proxies and mobile agents. While Voyager provides an extensive set of object

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

messaging capabilities it also allows object to move as agents in the network. Voyager combines the properties of a Java-based object request broker with those of a mobile agent system. In this way Voyager allows Java programmers to create network applications using both traditional and agent-enhanced distributed programming techniques. Voyager has set of features including: facilitates agent communication, supports agent security, and provides flexible life spans for agents by supporting variety of span methods. Another advantage of this platform, it gives location transparency through forwarding chains of proxies. Voyager has disadvantage involves in low performance, scalability, and robustness.

3.8 JADE

JADE is pure Java-based mobile agent platform developed by Telecom Italia, (last version: JADE 4.5, 2018). This platform provides variety of tools for controlling, managing agents. One of the features provided by this platform is that; it supports ontologies which represent agent behavior and focuses on the development of multi-agent systems. Mobility in JADE is built-in Agent Mobility Service and agents, therefore, searches the target location by Agent Management System according to FIPA standards. JADE provides user friendly interface GUI, high scalability, performance, and stability. Besides the agent abstraction, JADE provides a simple and powerful task execution and composition model, peer-to-peer agent communication based on asynchronous message passing paradigm, and advanced features that facilitates the development of distributed system.

JADE has an extensive feature on mobile devices and designed on new development environment such as Android devices. For further information, visit JADE web site: http://www.jade.tilab.com

3.9 CONCORDIA

Concordia is a framework for mobile agent system developed and supported by Mitsubishi Electric Information Technology Center, USA. Concordia is a complete Java based framework for network-efficient mobile agent applications which extend to any device supporting Java. The Concordia system is made up of numerous components, each of which integrates together to create full mobile agent framework. Concordia Server is the major building block, inside which has various Concordia Managers reside. Each Concordia component is responsible for a portion of the overall Concordia design, in a modular and extensible fashion. Concordia components are: Agent Manager, Administrator Manager, Security Manager, Persistence Manager, Event Manager, Queue Manager, and Directory Manager. A key advantage of using Concordia is using the security manager and it is disadvantage includes: low performance and scalability. Available web-site: http://www.merl.com/HSL/Projects/Concordia

3.10 Tracy

Tracy is mobile agent platform developed at the University of Jena in Germany. The architecture of Tracy execute plug-in software components that can be added to the running agency. The platform provides a high level services between agent communication and security. Communication between agents which implement local message passing. Tracy has weak performance and stability. For more information Tracy web-site: http://tracy.informatik.uni-jena.de.

3.11 SPRINGS

SPRINGS for short refers to (Scalable PlatfoRm for movINg Software). SPRINGS developed at the University of Zaragosa in Spain. a novel multi agent platform featuring location transparency, automatic update of proxies, and scalability. Besides, it minimizes live lock problems that arise when agents move quickly to remote hosts. This platform has set of features including scalability, stability, and built with high number of agents. The architecture has been inspired by JADE and Grasshopper features. The main disadvantage of SPRINGS is that; it does not support mobile agent communication using FIPA standards. SPRINGS has high performance, scalability, and it does not offer sophisticated security mechanism.

The API of this platform is available (updated on January 13, 2006). http://sid.cps.unizar.es/SPRINGS.

We summarize mobile agent's platforms according to three main features: performance, scalability, and robustness as shown in Table 1 and Figure 1.

Journal of Theoretical and Applied Information Technology
31st January 2020. Vol.98. No 02

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Results of Mobile Agent Platforms Features

Feature / Mobile Agent's Platform	Performance	Scalability	Robustness
JAMES	High	High	High
Odyssey	Average	Average	Average
Swarm	Average	Average	Low
Grasshopper	Average	Average	Average
Jumping Beans	Low	Low	Low
Aglets	Average	Average	Average
Voyager	Low	Low	Low
JADE	High	High	High
CONCORDIA	Low	Low	Low
Tracy	Weak	Weak	Weak
Springs	Weak	Weak	Weak

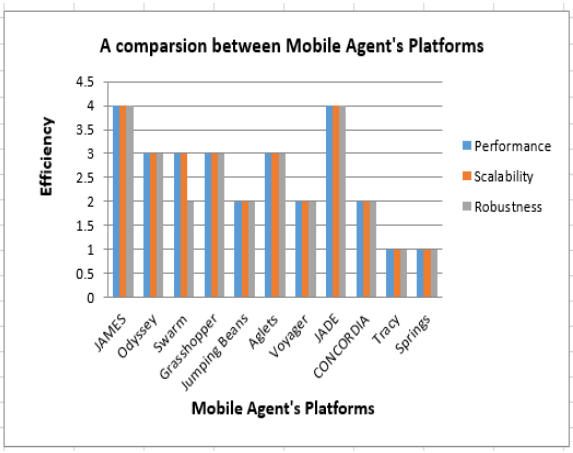

Figure 1: A comparison between Mobile Agent's Platforms Performance, Scalability, and Robustness

4. MOBILE AGENT PARADIGM

An agent is a software object that can move from one machine (Laptop or computer) or mobile device to another machine under control to achieve tasks upon user requests. An agent composed of three parts: the agent code represents developer algorithm, agent execution thread, and agent data. The mobile agent, the remote evaluation and the code on demand paradigm are part of the code mobility. However, an agent based platform must provide a set of capabilities to the agent. Details with basic features that an agent-based platform provides. The agent migrates to set of servers where information is collected and all parts of the agent including: agent code and execution thread, as shown in Figure 2.

Figure 2: Mobile Agent Paradigm

When an agent migrates to another machine, it should continue its execution on the remote host. However, most mobile agent platforms provide weak migration, i.e. an agent migrates to the destination without its execution state. This causes the agent to restarts execution from its beginning state each time of its trip.

The mobile agent works on a clear scenario. For example, a connected laptop to a network of computers. A mobile operator can work and move without staying on a mobile device, which means it can go to the Web sites to get information from the same vendor. And then a local assignment. The results obtained are then sent to the mobile device. The device can continue its work on a task and at the same time lose contact with the other temporarily. The result obtained will then be sent to the unit to follow up on the task [18].

Figure 3 shows the difference between client-server-based application and mobile agent-based application, where it is obvious that mobile agents can be useful in reducing the network's raw data flow [3].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

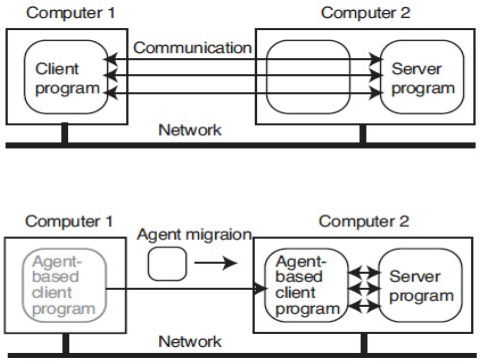

Figure 3: Mobile agent on Network Paradigm

The mobile agent is moved within and processed by a mobile agent system, which should have an agent's server (or agency), on each host. An agency consists of set of modules that constitutes the architecture of agent taxonomy:

- A **communication module** for sending and receiving the agents, and to exchange messages with non-local agents.
- **A repository module** to achieve authentication, queue up agents and set agents' priorities for later execution.
- **An executing module,** which is an interpreter to run agents
- **The state engine module** that keeps the agency's current execution state handles communication between local and inter agents, and finally, decide what to do with the agent by using an inference engine
- **A database or directory module** for storing or retrieving data by the agents
- **A security module** to monitor the agency as a whole and the agents' authorized activities.

Mobile agents support flexible and adaptive load transfer from host to host, depending on bandwidth and other available resources. So, mobile agent technology is good for wireless and dial-up environments. In a mobile agent, a group of nodes is moved in order to obtain a service that is to be presented to the user. Because of the increased use of services, there is an additional load that results in poor performance. It emphasizes that the measurement of service now relies on performance in terms of speed of implementation and responsiveness.

5. MOBILE AGENTS PLATFROMS PERFORMANCE VALUATION

In this work, the authors performed some evaluations for mobile agent's platforms. They focus on two main issues: comparison of different agent's solution or optimization of the target platform. In both cases, the performance evaluation relies on some performance metrics including: communication of agents, agent's mobility, usability and documentation, and agent execution run time. The last metric is considered in this research that explore platform performance evaluation.

This research shows performance evaluation of mobile platforms based on experimental tests for small agent data size, while the rest of the research proposed a mathematical model that simulates results when agent data size is large and very large.

6. ENVIONMENT CONDITIONS AND TEST EXPERIMENTS

In order to carry out tests, we define three parameters that have been used to implement these tests including: first, environment conditions such as configuration of hardware definitions (6 computers, 2 MHz processor, 2 GB RAM) and software installation Windows 7, Java 1.6.1 with JIT, second, number of agencies (number of itinerary performed by the agent, in this case, we proposed one agent) and agent lap, which is defined as the roam of agent across the network through a closed itinerary, in this case, we proposed agent lap is 1, and third: agent data size, in this case, we proposed only two agent data sizes 100 KB, and 1 MB. The mentioned conditions are applicable among all 11 platforms that have been used to report test experiments. We implement three additional tests for JADE, Qdyssey, and Tracy. The remaining tests were given and reported by [1] [2]. Average weighted test for each mobile agent platform is measured in milliseconds and is given in Table 2.

Journal of Theoretical and Applied Information Technology
31st January 2020. Vol.98. No 02

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Results of Mobile Agent Platforms Execution Time

Agent code size / platform	100KB Execution time (ms)	1 MB Execution time (ms)
JAMES	.70	1.17
JADE	.76	1.20
SPRINGS	.81	1.20
Odyseey	.89	1.22
Swarm	1.01	1.88
Grasshopper	1.47	2.19
Voyager	1.67	2.37
Aglets	1.73	2.35
Concordia	2.56	3.12
Tracy	4.12	4.76
Jumping Beans	5.9	6.42

According to experimental results, JAMES has been considered the large high performance execution time for small agent data size (100 KB, 1 MB), while Jumping Beans is the lowest performance execution time. Figure 4 shows mobile agent platforms with corresponding execution time for small agent size

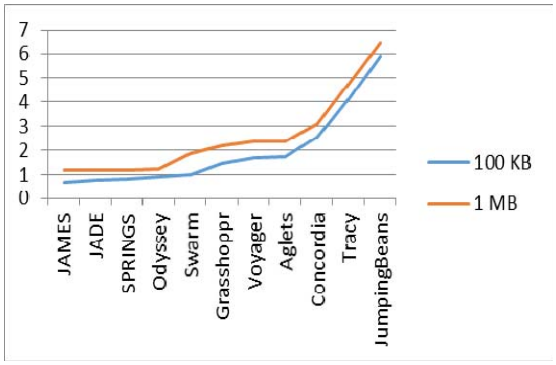

Figure 4: Platforms Performance Comparisons for Small Agent Code Data Size

All the platforms have been tested in the same conditions, the same testing parameters, the same application and the same configuration. When agent data size becomes quite large, i.e. more than 1 MB, the majority of platforms could not be completed the test, i.e., the agent could not continue its migration to another machines. We use the previous mentioned execution results of small agent data size to simulate results by defining mathematical model for different agent data sizes including (medium data size 2 MB, large data size 10 MB, and very large data size 1 GB).

7. METHODOLOGY: PROPOSED MATHEMATICAL MODEL

The proposed model has been used to derive medium, large, very large agent data sizes. Resulted are computed according to the mathematical prediction model based on Newton's first ordered divided difference formula [17].

For given $x_0, x_1, x_2, x_3 \ldots\ldots\ldots\ldots x_n, \exists\ p_n(x_i)$

is a polynomial of degree *n* for values

$$f(x_i)\ at\ x = x_i \qquad and\ i = 0,1,\ldots\ldots,\ n\ .$$

This polynomial approximates a function such that

$$p_n(x_i) = f(x_i); \qquad \forall i = 0,\ldots\ldots\ldots\ldots n$$

and suppose

$$f(x_i) = \alpha_i \forall i = 0,1,2,\ldots\ldots,n \qquad (1)$$

$$e_k(x) = \prod_{i=0}^{k-1}(x - x_i) \qquad (2)$$

Then

$$P_n(x_k) = \sum_{k=0}^{n} \alpha_k e_k(x) \qquad (3)$$

Assume n=1, we derive the mathematical formula, which is called the first order difference formula. i.e.

$$P_n(x_1) = \sum_{k=0}^{n} \alpha_k e_k(x_1)$$

$$= \alpha_0 + \alpha_1(x - x_0) \qquad (4)$$

Using (1) and (2), we get

$$\alpha_1 = \quad f(x_0, x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \qquad (5)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$f(x_0, x_1)$, is called the first order divided difference formula of the linear mathematical model.

In computational processing algorithm, we have supposed $(x_0 = 100KB, x_1 = 1MB)$ and $(f(x_0), f(x_1))$ the corresponding execution time for each platform, furthermore, α_0 is proposed the value of platform execution time corresponding to the smallest agent data size (i.e. 100KB. Based on these assumptions, we have computed α_1 using equation 5, and finally, we have used α_0,and α_1 to compute the linear formula given in (4) for each platform. Table (3) shows mobile agent platforms and the corresponding α_0, α_1,and platform linear prediction formula.

Table 3: Mobile Agents Platforms Linear Formula

Agent size / platform	α_0	α_1	Linear formula
JAMES	.70	.00052	.000522x+.6476
JADE	.76	.00049	.00049x+.711
SPRINGS	.81	.00043	.00043x+.767
Odyseey	.89	.00036	.00036x+.9267
Swarm	1.01	.00096	.00096x+.914
Grasshopper	1.47	.0008	..0008x+1.39
Voyager	1.67	.00078	.00078x+1.592
Aglets	1.73	.00069	.00069x+1.799
Concordia	2.56	.00062	.00062x+2.48
Tracy	4.12	.0007	.0007x+4.05
Jumping Beans	5.9	.00058	.00058x+5.842

8. PERFORMANCE RESULTS AND ANALYSIS

Using table 3, we have computed the execution run time in milliseconds of each platform for different cases: (medium agent data size 2 MB, large agent data size 10 MB, and very large agent data size 1 GB). Results are analyzed in each separate case. Table 4, presents mobile agent platforms execution time of the three case

Table 4: mobile agent platforms Execution Time

Agent size / Platform	2 MB Execution Time	10 MB Execution Time	1 GB Execution Time
JAMES	1.6878	5.8478	520.6478
JADE	1.691	5.611	490.711
SPRINGS	1.627	5.067	430.767
Odyssey	1.646	4.526	360.964
Swarm	2.834	10.5	960.914
Grasshopper	2.99	9.39	801.34
Voyager	3.152	9.392	781.592
Aglets	3.179	8.699	691.799
Concordia	3.72	8.68	622.48
Tracy	5.45	11.05	704.05
Jumping Beans	7.002	11.642	585.842

8. 1 Case 1: Analysis of Medium Agent Data Size 2 MB

In this analysis, we have computed platform execution run time using its linear formula for all the platforms by assuming platform environment parameters: (one Agent, 1 lap, Medium agent code size 2 MB) The results presented in Figure 5. Results present SPRINGS, Odyssey, JAMES, and JADE gives the best results among all other platforms. Tracy and Jumping Beans are the slower run time platforms: for instance, in this case Tracy and Jumping Beans executed from 5 to 7 times slower than SPRINGS, Odyssey, JAMES, and JADE.

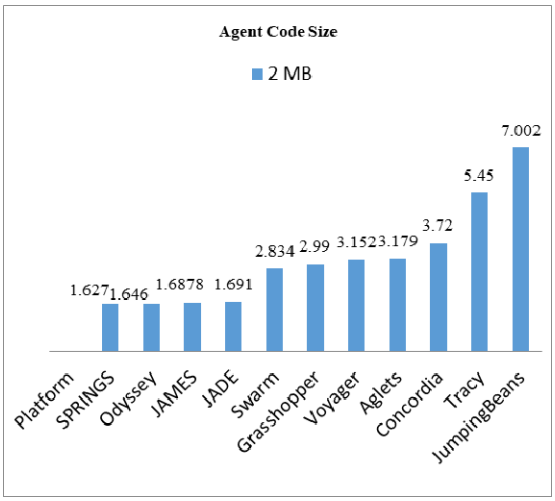

Figure 5: Platform Execution Run Time: Medium Agent Code Size (2MB)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

8. 2 Case 2: Analysis of Large Agent Data Size 10 MB

In this case, when agent code size increases to 10 MB and by assuming platform environment parameters are the same as in case 1, results shows in Figure 6, present that Odyssey, SPRINGS, JADE, and JAMES still have the best results among all other platforms. However, it presents also that ordering of platforms is quite different comparing to previous case. Odyssey becomes the first platform and JAMES becomes the fourth one. It is reported that the average execution time in platforms (Concordia (8.68 ms), Aglets (8.69 ms), Grasshopper and Voyager is (9.39 ms). The slowest platforms are still Tracy (11.05ms) and Jumping Beans (11.64 ms).

Platforms are given below from (faster to slower):
Odyssey, SPRINGS, JADE, JAMES, Concordia, Aglets, Grasshopper, Voyager, Swarm, Tracy, and Jumping Beans.

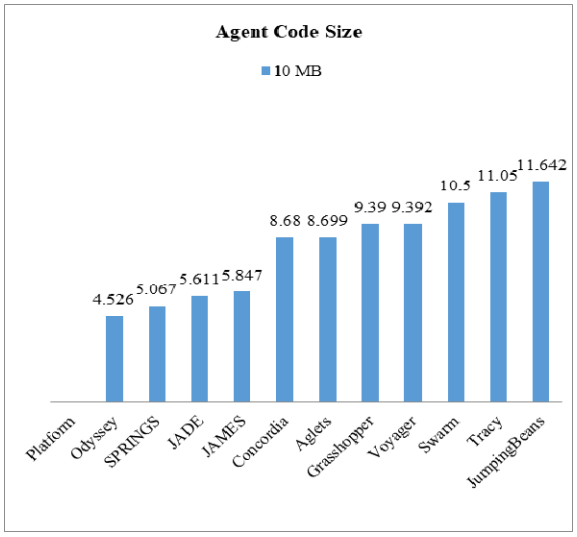

Figure 6: Platform Execution Run Time: Large Agent Code Size (10 MB)

8. 3 Case 3: Analysis of Very Large Agent Data Size 1 GB

In the last case, when agent code size becomes very large (1 GB), and by assuming that all platforms are running under the same environment parameters, numerical results show as in Figure 7, that the same platforms: Odyssey, SPRINGS, JADE, and JAMES are still offered the best results among all other platforms. Ordering of platforms in case where agent code data size is very large gives the highest evaluation to Odyssey (360.964 ms). The average execution time in this case: Jumping Beans (585.842 ms), Concordia (622.48 ms), Aglets (691.799 ms), and Tracy (704. 05 ms). Finally, the weak performance execution time is reported as: Voyager (781.592 ms), (Grasshopper, 801.34 ms), and (Swarm, 960.914 ms).

Platforms are given below from (faster to slower):
Odyssey, SPRINGS, JADE, JAMES, Jumping Beans, Concordia, Aglets, Tracy, Voyager, Grasshopper, Swarm.

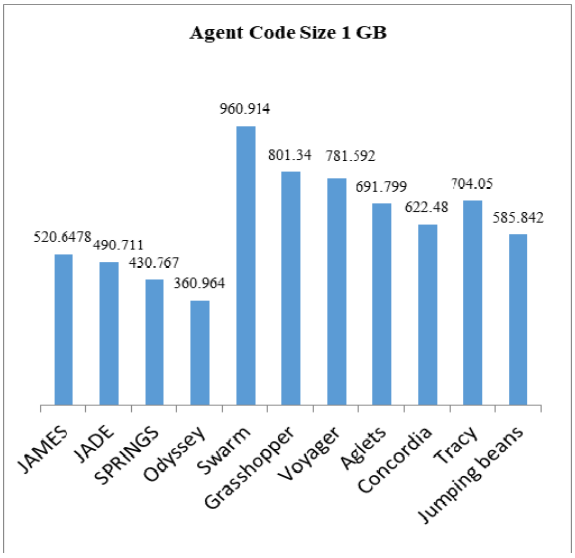

Figure 7. Platform Execution Run Time: Very Large Agent Code Size

9. CONCLUDING REMARKS AND FUTURE DIRECTIONS

In this research, we will present some conclusions about the behavior of mobile agent's platforms. We will also relate some facts that have been observed during the analysis of the tests. These facts give some feedback for developers to improve some weak points when they implement a platform. The developer should be also considered about good features, the performance, the robustness, and the weak points of these platforms.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We implement a mathematical model that help in the analysis process. We believe that in some extent, mathematical models can solve problems and can be used to forecast platform performance evaluation and predict their behavior when existing some shortcomings related to experimental tests.

Results of experimental tests for small agent code data size including (100 KB and 1MB) have been used for mobile agent's platforms to establish linear formula based on a mathematical model. The model has been used to predict results about agent platforms in three different situations: Medium agent code data size (2 MB), Large agent code data size (10 MB), and very large agent code data size (1 GB). The performance execution of the platform has been categorized in three different characteristics: high, average, and weak performance. Table 5, summarizers research concluding results of performance behavior of each mobile agent platform.

Table 5: Summarization of Behavior of Mobile Agent Platforms

Agent code size / Platform	Small	Medium	Large	Very Large
JAMES	High	High	High	High
JADE	High	High	High	High
SPRINGS	High	High	High	High
Odyssey	High	High	High	High
Swarm	Average	Average	Weak	Weak
Grasshopper	Average	Average	Average	Weak
Voyager	Average	Average	Average	Weak
Aglets	Average	Average	Average	Average
Concordia	Weak	Weak	Average	Average
Tracy	Weak	Weak	Weak	Average
Jumping Beans	Weak	Weak	Weak	Average

We think that performance is one of the metrics that should be taken into developer thinking. Other features are not considered in this research and also play a key role such that: robustness and functionality. Another important issue that the developer should take into his/her considerations is the field of platform implementation and the range of his application.

We interested in including another mobile agent's platforms such as Tryllian and Tacoma to evaluate the performance and to consider another metrics that can affect the performance evaluation. With this piece of work, we could encourage researchers to get benefits from points mentioned in the performance evaluation.

ACKNOWLEDGEMENT

The authors owe thanks to Isra University at Jordan for supporting this research. The university support us with all financial requirements, technical labs, and administrative requirements. Special thanks also owe to the faculty of Information Technology at Isra university for providing us with Software and Hardware support that assist in completion of this research.

REFRENCES:

[1] L.M.Silva, P.Simoes, G. Soares, P.Martines, V. Patista, C. Renato, L.Almeida, N.Stohr, JAMES: a platform of mobile agentsfor management of telecommunication networks, *Proceedings of IATA'99. Intelligent Agents for Telecommunication Applications,* Stockholm, Sweden, August 1999, pp. 254-267.

[2] L.M.Silva, G.Soares, P.Martins, V.Batista, L.Santos, "Comparing the performance of mobile agent systems: a study of benchmarking" , *Computer Communication, Elsevier, Vol. 23, No.1,* 2000, pp.769-778.

[3] Voyager and agent platforms comparison, Technical Report available at: http://www.objectspace.com/products/voyager.

[4] D. Lange, M.Oshima, "*Seven good reasons for mobile agents", Communication of the ACM, Vol. 42, No. ,* 1999, pp. 88-89.

[5] K. Kravari, N. Bassiliades, *"A Survey of Agent Platforms", Journal of Artificial Societies and Social Simulation (JASSS), Vol. 18. No.1. 2015*,pp. 10-19. [DOI: 10.18564/jasss.2661.

[6] B.D. Noble, M. Satyanarayanan, "Experience with adaptive mobile applications in Odyssey*", International Journal of Mobile Networks and applications, Vol..4, No. 4, 1999, pp. 245-254.*

[7] R. Trillo, S.Iiarri, E. Mena, "Comparison and Performance Evaluation of Mobile Agent Platform", *3rd International Conference on Autonomic and Autonomous Systems,* IEEE (Greece), pp. 19-25 June 2007.

[8] M.Breugst, S. Choy, M. Hoft, T. Magedans, " Grasshopper-An Agent Platform for Mobile Agent-Based Services in Fixed and Mobile Telecommunications Environments", *Software Agents for Future Communication Systems,* Springer-Verlag Berlin Hidelberg, 1999, chapter 14, pp. 326-357.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[9] F. Bellifemine, G. Caire, D. Greenwood, "Developing multi agent systems with JADE" John Wiley & Sons, pp. 240-252, USA, 2007.

[10] A. Burkle, A. Hertel, W.Muller, M, Weiser, " Evaluating the security of mobile agent platform", *Autonomous Agents and Multi-Agent Systems"*, Springer 2007, *Vol. (18), Issue (2)*, pp. 295-311.

[11] F. Alshrouf, A. Alhroob, K.Alshqeerat, Y. Alkubaisi, "Comparisons and Evaluation Performance Criteria of Software Agents' Platforms for E-Commerce", *International Journal of Applied Engineering Research, Vol. 13, No. 7,* 2018, pp. 5423-5427.

[12] A. Deshpande, B. Prakash, M. Alkkal, A. Siddiqui, "Mobile Agent Search in Multi-Agent System', *Proceedings of 3rd national Conference on Nascent Technologies,* 2012, pp.6-18.

[13] F. Alshrouf, T. Aiman. "Analysis of Mobile Agent Systems Performance using Linear Prediction Model", *International Journal of Reviews in Computing, Vol.12. N0. 7.,* 2012,pp.38-43.

[14] E. Shakshuki,"A methodology for evaluating agent toolkits", *Proceeding of International Conference on Information Technology,* USA.IEEE Computer Society, 2014, pp. 391-396.

[15] L. Prakash, V. Rajguru, S. Deshmukh, "Analysis of Mobile Agents", *Journal of Global Research in Computer Science, Vol.2, No. 11,* 2011, pp. 2560-2566.

[16] J. M. Alberola, J. M. Such, A. Fornes, A. Espinosa, V. Botti. "Performance evaluation of three multi agent platforms', *Springer Science Business Media,* 2010,pp. 146-154.

[17] R. Burden, D. Faires. " Numerical Methods" *Brooks/COLE Learning Inc.* USA, 2003, pp. 67-82.

[18] R.S.Gray, D. Kotz, R.A. Petrson, J. Batron " Mobile Agent versus client/server performance: Scalability in an information-retrieval task". *International Conference on Mobile Agents, Springer, Berlin, Hiedlberg,* 2001, pp. 229-243.