International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 17 (2017) pp. 6870-6880
© Research India Publications. http://www.ripublication.com

Performance Evaluation of e-Advertising Framework using Intelligent Mobile Agents

Roopa G M* and Dr. Nirmala C R

**Assistant Professor , Prof.& Head*
Department of Computer Science & Engineering,
Bapuji Institute of Engineering & Technology, Davangere, Karnataka- India.
E-mail: roopa.rgm@bietdvg.edu , crn@bietdvg.edu
**Orcid ID: 0000-0001-5451-5972*

Abstract

In today's competitive world, e-Business/e-Commerce has become one of the major segments of the online business and deals with the process of buying and selling of products and services online via Internet. e-Advertising being part of e-Commerce has major impact on e-transaction and helps in revenue generation. Currently, e-shops/e-publisher sites publish the product catalogue and advertisements on the Internet using one-to-one mapping of advertisers to the ad-publisher sites. In such cases, huge bandwidth is required as publishing data size increases, network traffic increases with the increase in advertiser requests and continuous connectivity is required for advertisers to post the advertisements on ad-publisher sites. Here, we propose a Hybrid e- Advertising framework and is designed using Concurrent Mobile Agents and provides an environment for parallel processing the advertiser requests to deal with issues of one-to-many and many-to-many mapping between the advertisers and the ad-publishers sites. A new concept of aggregating the set of advertisers request and simultaneously publishing at various ad-publisher sites is proposed. Next, we have given a comparative analysis of the framework with the Client/Server approach which is in practice. The framework is designed and developed using Java based Aglet Software Development Kit and MySQL for storage requirement.

Keywords: e-Advertising, Mobile Agent, Parallel Processing, Aggregation, push and pull model, Performance Evaluation

INTRODUCTION

The advancement in the web technologies like World Wide Web, Internet ,Xhtml, Ajax has great impact on the development of applications related to e-Commerce / e-Business and e-Advertising [12]. They also provide an important channel for most of the business firms, companies and organizations to carry out their business transactions. Thus the daily life of the customers has become convenient with online shopping. But, much of the research work has reported that the number of customers buying and selling of products, performing electronic transactions on the web is increasing at a phenomenal pace [6]. Moreover, the increase in advertisers, ad-publishers and products lead to the problem of information overload, which of course consumes too much of the advertiser time on visiting the various ad-publisher sites for promoting their products and services. And also consumes too much of customer time in searching the relevant product information of their interest. But in the current scenario most of the applications related to e-commerce and e-advertising are still non-automated [6].

Hence, it is necessary to provide the advertisers and the customers with user friendly and more convenient environment for publishing the advertisements and to provide personalized product services for customers.

One way to solve the above problem is to exploit the use of Intelligent Mobile Agents for e-advertising. The traditional client/server model for e-advertising adopts one-to-one mapping between the advertisers and the ad-publishers, which requires a reliable connection [5] for publishing the advertisements, the network traffic increases as the number of advertiser requests increases with the increase in the number of ad-publisher sites. A good bandwidth, is required [22] as large amount of information needs to be published. It is observed that most of the advertiser needs to request the distributed ad-publisher sites for the purpose of publishing the advertisements, and then the parallel execution of Mobile agents at various ad-publisher sites over the network seems to be an efficient solution. And at the same time, the dynamic and asynchronous execution of Mobile agents provides an environment for parallel processing of publishing the advertisements to achieve greater efficiency [12]. Mobile agents can also help the web users to be more productive to manage information and move towards a high level of automation [6].

In this paper, we propose a framework which makes use of Intelligent Concurrent Mobile agents for e-advertising which supports the real activities of advertisers, ad-publishers and customers and also facilitates the parallel processing of the advertiser requests for publishing the advertisements and we have introduced the new concept of aggregating the advertiser requests which deals with the issues related to one-to-many and many-to-many mapping between the advertisers and ad-publishers sites. Next, we have identified various e-Advertising application parameters which influence the performance , such as the total time taken to process the advertiser request for publishing and network bandwidth utilization in terms of data transfer rate.

Experiments have been carried out in three cases. Firstly, we have considered a number of advertisers request to publish the advertisements, secondly the number of ad-publisher sites and finally about the data transfer rate to study their effect on the performance of both the paradigms. Here we have conducted two sets of experiments, in the first experiment we have conducted a study on the traditional Client/Server model for e-Advertising which employs one-to-one mapping between the advertisers and the ad-Publisher sites, where in every advertiser request is synchronously processed. Client/Server test-bed is set up using HTML, PHP, MySQL which is the widely adopted package for publishing the advertisements. In the second experiment, we have considered the Mobile Agent paradigm for e-Advertising which employs one-to-one, one-to-many and many-to-many mapping between the advertisers and the ad-Publisher sites where in every advertiser request is processed asynchronously and autonomously.

Based on the proposed framework, we have analyzed, designed and implemented a prototype to carry out the task of e-advertising using Aglet technology developed by IBM Research labs, Tokyo, Japan [19], which is Java based framework for creating and running Mobile agent objects. The framework enhances the by adopting the strong features of Aglets for parallel processing the advertiser request for the purpose of posting the advertisements.

The results of both the experiments show that, the Mobile Agent approach over perform by a wide margin when compared to that of the Client/Server model to publish the advertisements.

RELATED WORK

Authors in [7,9,24] have identified several parameters that influence performance and have carried out a quantitative evaluation of client-server versus Mobile Agent implementations. A comparative study is carried between client server approach and Mobile Agent approach applied for e-Commerce applications. Finally they have concluded that the Mobile Agent is a promising paradigm for the design of information retrieval systems in a distributed environment for given running application.

Ryszard Kowalczyk et al. [13] they presented Intelligent Mobile Agent-based e-marketplace system called InterMarket. Their work on InterMarket aims at enabling Mobile access and automated trading in e-marketplaces based on integration of Mobile Agents and intelligent decision-making Agents offered as an add-on component to a commercial e-marketplace platform. They have attempted to demonstrate the benefits with the commercial e-marketplace, focusing initially on the human trader emulation in a catalog-based e-procurement business model scenario, predominantly for Agent-to-human trading with the use of stationary trading Agents and Mobile communication Agents.

Wang Y. and Ren J. [12] discussed and presented various hierarchical dispatch models where the dispatch of multiple Mobile Agents can be processed in parallel. Their study on these models have proved that, the comparison with several serial Mobile Agent models the parallel Mobile Agent models can improve the performance significantly.

Roch H. Glitho [16] describes an Agent-based approach to searching for information on the WWW and also pointed out the major drawbacks of the current searching techniques. They have presented many various Agent interaction strategies. The authors have suggest that one possible means of achieving the task for information retrieval by using Mobile Agents to wander the web seeking the information on behalf of the user.

David Siew, Xun Yi [17] intends to apply software Agents to automate Internet advertising and presented four models of Agent-mediated Internet advertising: negotiation and payment, push, post and pull models. Software Agent technology offers a new paradigm for advertising on the Internet. Unlike "traditional" software, software Agents are personalized (incorporating cooperation, negotiation and conflict resolution), continuously running and semiautonomous. From their work they have concluded that Software Agents are most suitable for automating several of the most time consuming stages of the Internet advertising.

SUMMARY

From the literature review and research issues in the current scenario we can say that the existing approaches used for e-Advertising have been implemented on the traditional client/server model. As the client/server model suffers from serious drawbacks such as there should exists a reliable connection to have continuous communication between advertisers and various ad-publisher sites, a synchronous communication between advertiser and ad-publisher for every request and response, needs a good bandwidth as large volume of data is transmitted over the network, frequent requests from the advertisers will increase the load on the server and also network traffic. The issues are also related to one-to-many and many-to-many mapping of advertisements between the advertisers and the ad-publisher sites, followed by monitoring and tracking of the posting status to give confirmation to the advertiser about the delivery of advertisements.

It is observed from the literature survey that Mobile Agent paradigm has emerged as a very promising approach for conducting e-Commerce/e-Business and e-Advertising. The study on software Agents have shown that they can help the web users to be more productive to manage information and to move towards a higher level of automation. By identifying the technical advantages of Mobile Agents like, low bandwidth remote interaction, support for disconnected operation, support for weak clients, ease of distributing individual services, scalability , semantic routing ,lower overhead for secure transactions and robust remote interaction. Thus Mobile Agents can provide a pervasive, open, generalized framework for the development and personalization network services.

PROBLEM DEFINITION

In the current scenario, identifying the various mapping relationships among advertisers, advertisements / products, ad-publishers, advertising spaces, and customers in a static or dynamic manner is the challenging task. As it is observed that most of the e-Advertising applications that are currently running are developed on the foundation of client/ server model that adopts one-to-one mapping technique where the set of advertisers needs to contact set of ad-publisher sites to promote their products or services that usually consumes lots of bandwidth and also there should exists a reliable synchronous communication between the advertisers and the ad-publishers for every request and response.

Thus to overcome the major drawbacks of the existing model for e-Advertising , we have proposed a common framework for various aspects of e-Advertising by applying Intelligent Concurrent Mobile Agents, along with that an attempt is made to make the framework more robust, efficient and effective.

COMPARISON OF MOBILE AGENT PARADIGM WITH THE CLIENT/SERVER MODEL

The most commonly supported traditional paradigm for distributed network computing, e-Commerce, e-Business, e-Advertising is the client/server model. Here, the client initiates the request by establishing a continuous connection with the server across the network until the client request is processed. Fig.1 shows the traditional approach supported for e-Commerce/e-Advertising applications for buying/selling/promoting the products/services.

The existing approach becomes unreliable and expensive when the number of client requests to be processed increases. Such, applications consume huge network bandwidth, high network delay, and heavy load on the server, high cost of network management, continuous network connectivity and one-to-one mapping between the client/server. This leads to the basis for mobile agents which has the capability of performing the tasks on behalf of the user, thinks intelligently using some of the AI algorithms and learn/remain-persistent.

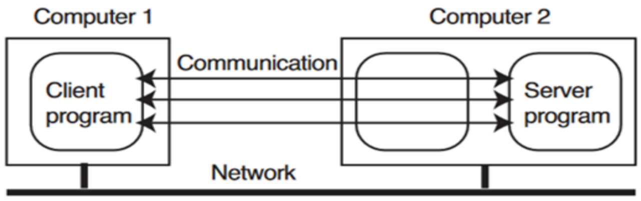

Figure 1: Client/Sever Communication Paradigm

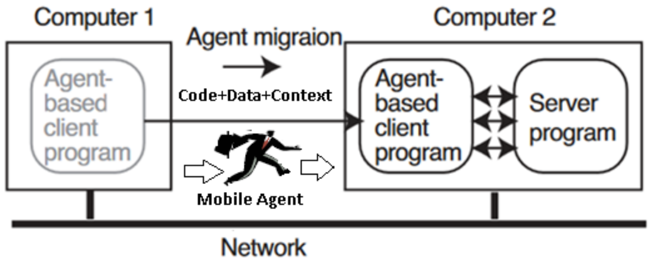

Figure 2: Mobile Agent Based Communication Paradigm

Mobile Agent paradigm shown in Fig.2, proves to be the best option for conducting effective multi-media communications, and information retrieval/gathering over the Internet. They migrate from one host to another in the network with its own execution control to process the client request and return the results/messages in an asynchronous pattern. The mobile agent paradigm overcomes the limitations of the client/server model mentioned above.

MOBILE AGENT BASED E-ADVERTISING FRAMEWORK

The e-Advertising framework, which is built on the basis of Intelligent Mobile Agents and the new technique that is adopted for publishing the advertisements, is discussed here. Following are the main components of the framework, it includes the four main modules, namely, Advertiser module, Aggregator Module, ad-Publisher module and the customer module. The architecture diagram of the framework is given in Fig 3.

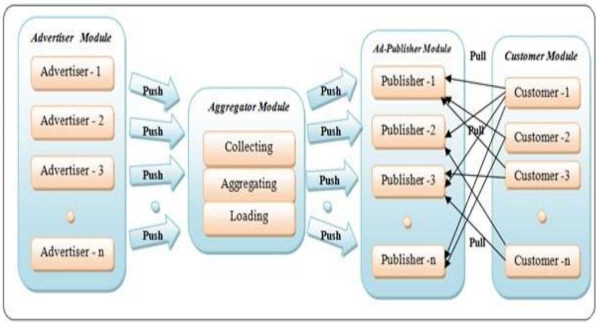

Figure 3: Mobile Agent Based e-Advertising Framework

A. *Description. The framework consists of the following modules*

- Advertiser module

The advertiser plays the role of advertisements supplier. The advertiser creates the advertisements that need to be published via Aggregator module on the ad-publisher site. Each advertiser maintains a database that holds the product/Item information like, images, cost, model, make, category and description . Here Mobile Agents are created at every advertiser site that carry the product information on behalf of the advertiser and migrates from the advertiser site to the aggregator module to push the advertisements. In order to perform the above task, module takes the help of the following Agents that are shown in Fig. 4.

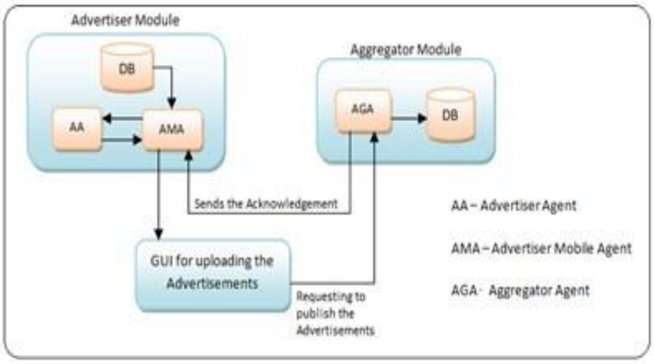
Advertiser Module
DB
AA
AMA
Aggregator Module
AGA
DB
Sends the Acknowledgement
GUI for uploading the Advertisements
Requesting to publish the Advertisements
AA – Advertiser Agent
AMA – Advertiser Mobile Agent
AGA- Aggregator Agent

Figure 4: Advertiser Module

- Advertiser Agent (AA): The advertiser Agent acts as stationary Agent at advertiser site, which is responsible for creating and managing the Mobile Agents. This stationary Agent creates the Mobile Agent, loads the advertisements created and obtains its proxy then dispatches the loaded Mobile Agents to the predefined Aggregator module and waits for notification from the Aggregator module.
- Advertiser Mobile Agent (AMA): This Mobile Agent carries the created advertisement from every advertiser site and hands it to the stationary Agent at Aggregator module. The stationary Agent is responsible for collecting and aggregating and processing the advertisements pushed by the set of advertisers. After the advertisements have been pushed, the advertiser Mobile Agent can either return back to its dispatching server or can be disposed at the aggregator module.

The pseudo codes of the Advertiser-Agent and Advertiser Mobile Agents are given here:

Advertiser-Agent:

```
//Class for Static Advertiser-Agent
 import*. aglet;
class Advertiser extends Aglet {
public void run ( ) {
 Advertiser-Aglet( object initialize) {
Creates Advertiser Mobile Agent
for each advertiser in the advertiser list
loads and dispatch the advertiser Mobile Agents with the advertisements
get_proxy(Aglet-Id){
Dispatch ( ) {
      waits till it receives the reply }
} }
 } }
```

Advertiser Mobile-Agent :

```
//Class for Advertiser Mobile-Agent
import *.aglet;
class AdvertiserAgent extends Aglet {
public void run ( ) {
on Dispatch( ){
for each advertiser in the advertiser list
 loaded Mobile Agents migrates to the aggregator module
 requests to processRequest( )
}  }  }
public void processRequest(reqAdvertisements Type advertisements){
if advertisement is accepted
   pushes the advertisements to the aggregator Agent for
     aggregation
     returns the acknowledge to the advertiser Agent
else
     notify with error message to the resend }
```

- Aggregator module

This module provides an interface between the set of advertisers and the ad-publisher sites, which is based on a ”one-to-many” and many-to-many model wherein the advertisements are published across multiple ad-publishers over the Internet. Thus acts as a single stop-shop for the advertisers to push their advertisements which contains the product information, rather than browsing various ad-publishers sites to upload the advertisements for the promotion. This reduces the work of managing relationships with multiple advertisers multiple ad-publishers. The Aggregator Agent creates a single package of the advertisements from various advertisers and maintains a database that stores the information related to the product, advertiser and ad-publisher. In order to carry out the above tasks, this module takes the help of the following Agents that are shown in Fig. 5:

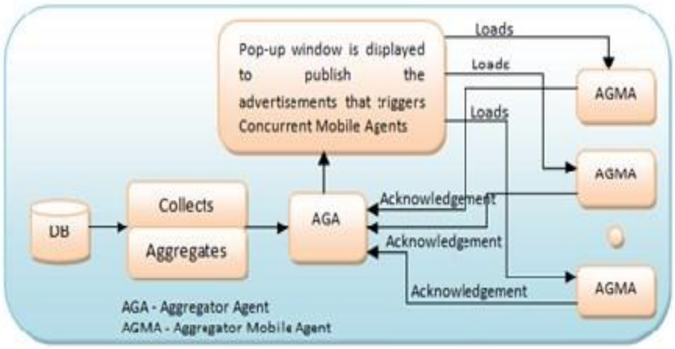
Loads
Loadc
Loads
Pop-up window is displayed to publish the advertisements that triggers Concurrent Mobile Agents
AGMA
AGMA
AGMA
DB
Collects
Aggregates
AGA
Acknowledgement
Acknowledgement
Acknowledgement
AGA - Aggregator Agent
AGMA - Aggregator Mobile Agent

Figure 5: Aggregator module

- Aggregator Agent (AGA):

  This stationary Agent creates, manages the Concurrent Mobile Agents. The Agent is responsible for aggregating the advertisements and creating a package of the advertisements which are to be pushed from set of advertisers. It also collects necessary information from both the advertisers and pre-defined ad-publishers sites.

- Aggregator Mobile Agent (AGMA):

  These Agents are created by Aggregator Agent. Mobile Agents migrates to predefined ad-publisher sites for publishing the advertisements simultaneously. Mobile Agents after publishing will return to dispatching server and acknowledges the Aggregator Agent.

The pseudo codes of the Aggregator Agent and Aggregator Mobile Agent are given here:

<u>Aggregator-Agent:</u>

```
        //Class for Static Aggregator-Agent
  import *.aglet;
 class Aggregator extend Aglet {
  public void run( ) {
 Aggregator Aglet( initialize the object) {
  wait for the arrival of Mobile-Agents from advertiser sites
 and                    on arrival
  for each advertiser in the advertiser list
 AgentResponse(advertisement)
aggregates only the successfully received ads
             creates multiple Mobile-Agents
 loads and dispatch aggregator Mobile-Agent with
aggregated-
    advertisements
get_proxy(Aglet_Id){
Dispatch ( ) {
    waits until it receives the reply
    }     }
             }
}     }
     public void AgentResponse( req_Advertisement_Type
     advertisement) {
     for each advertiser in the list
     if aggregator Agent receives the ads and process the
     request successfully
         collects/aggregate the advertisements returns
     acknowledgement to the advertiser-site
      else
         cannot receive or process the request
     }
```

<u>Aggregator Mobile-Agent:</u>

```
//Class for Aggregator Mobile-Agent
import *.aglet;
class AggregatorAgent extends Aglet {
public void run( ) {
on Dispatch( ){
loaded Mobile-Agents migrates to the publisher sites
and publishes the ads
successfully based on the advertiser preference
 returns the acknowledge to the aggregator Agent
     }   }      }
```

- Ad-Publisher module

  This module consists of pre-defined set of ad-Publisher sites which reserves the screen space and time slots for displaying the advertisements. The Aggregator Mobile Agent pushes the advertisements via Aggregator Mobile Agents and the same is responsible for displaying the advertisements in the reserved screen space for a specified time slot.

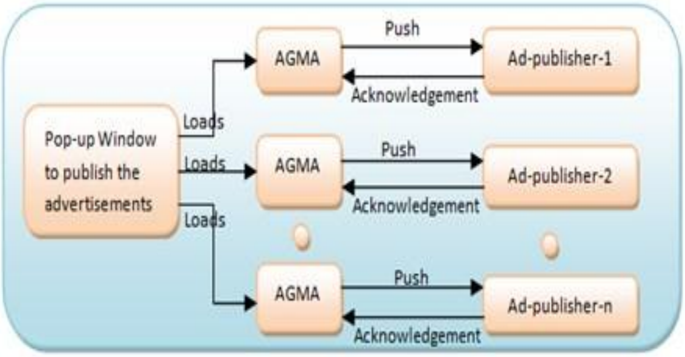

Figure 6: Ad-Publisher module

- Customer Module

Most of the advertisers select Internet and WWW as a medium to widen their sales in reaching their potential customers among the web users.

This module plays the role of a potential customer who wishes to purchase products and services. Depending on their interest the customer can pull the product information for further processing.

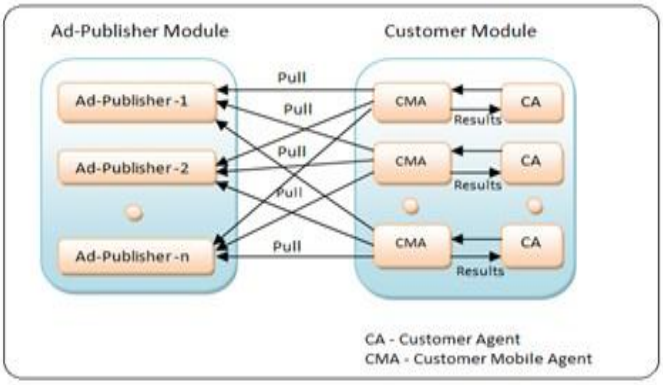

Figure 7: Customer Module

To carry out the above tasks, this module takes the help of the following agents that is shown in Figure. 7:

– Customer Agent (CA):

This is a stationary Agent that creates and manages the Mobile Agents at every customer site and is responsible for recording the product information for further processing.

– Customer Mobile Agent (CMA):

On arrival of the advertisements at the ad-Publisher sites, the Mobile Agent is created by the customer Agent that migrates to pull the product information based on the attributes like name, brand and make of the product from various publisher sites .At every publisher site, the Mobile Agent searches for the required advertisements. On finding the advertisement, Mobile Agent returns to the dispatching server and finally displays the results to the customers.

WORKFLOW PROCEDURE

Based on the proposed framework, the necessary procedures required for completing the task of publishing advertisements are as follows:

- Advertiser request for publishing

Front-end (GUI) will be provided for advertisers to upload the advertisements by submitting the relevant product information such as name, model, category, cost and description for publishing. Here GUI supports both one-to-many and many-to-many mapping between the advertiser and the ad-publisher. Finally receives the acknowledgement once the request is processed.

- Processing Advertiser request

Publishing request by the advertiser, the aggregator agent receives the request and processes it based on the product information provided. If the advertisements are accepted, then they are aggregated by the aggregator agent and sends the acknowledgement. Otherwise acknowledges with an error message to resend the request.

- Collecting and Aggregating the advertisements

All successfully processed advertisements are collected and aggregated by the Aggregator Agent and also creates a single package of the set of advertisements that are to be published.

- Loading the advertisements

The collected and aggregated advertisements are loaded with concurrent Mobile agents for publishing.

- Publishing the advertisements

Mobile agents which are created by the aggregator agent are responsible to publish the loaded advertisements and return the acknowledgement after publishing is successful.

- Viewing and Purchasing the product

As the advertisements are published successfully the Mobile agents at every customer site pulls the product information of their interest based the model, cost, brand and etc. This information is used by the customer stationary agent for further processing.

PERFORMANCE EVALUATION

The proposed work describes the total time taken and the network bandwidth utilization in order to compare the performance of the Client/Server and Mobile Agent Paradigm applied to e-Advertising.

Following are the parameters as influencing the choice for concurrent mobile agents as an implementation strategy for e-Advertising applications.

- Advertisement Data size (varies from 50 KB to 2 MB)
- Number of advertiser requests (varies from 1 to 64)
- Number of ad-publisher sites (varies from 1 to 128)
- Size of an MA (fixed at 5 KB)
- Network Load (as per typical academic load on 100 Mbps LAN).
- Aggregation time (varies from 05 to 10ms)
- Processing time for serving the advertiser request (varies from 20 ms to 6000 ms).
- Data Transfer rate (throughput)
- Module Dependencies.

- Experimental Setup for the proposed Framework

  Experimental setup to deploy and check the functionality of the proposed framework has been carried out in LAB-1, LAB-2 and LAB-3 of Research Centre CSE Department where all the Computing System have the similar configuration of Intel Core i5/i3, 8GB RAM and are connected through a 10/100/1000Mbps LAN. Here each host is configured with JDK 1.6 above, Aglet Software Development Kit 2.0.2 and MySQL (5.5.24) for storage purpose.

- Experimental Setup for Client /Server Model

  The Client/Server test bed is created for the purpose of comparing and evaluating the performance of the proposed framework. This is a web based application that hosts advertisements, where the advertiser makes requests to publish at various ad-publisher sites. The application is deployed on Wamp-Server (2.2.22) and it makes use of PHP and MySQL for the purpose of posting the advertisements.

- The set of parameters considered in comparing the two approaches

The comparative parameters considered and their meaning is given below:

– Total time taken: Refers to the total time taken for processing each request from the advertiser.

$$T_{CS} = \sum_{i=1}^{n} \left(t\left(req\right)_i + t\left(res\right)_i \right) \qquad (1)$$

$$T_{MA} = \left(\left(\sum_{i=1}^{m} \left(t\left(ad\right)_i \right) / m \right) + t_p + t\left(pub\right) \right) \qquad (2)$$

where,

T_{CS} = Refers to the time taken by the advertiser to send, request and waits for a server publishing response in case of a Client/Server model.

$t(req)_i$ = Request time by each advertiser to publish.

$t(res)_i$ = Response time to acknowledge each advertiser. n=Refers to the number of advertisers and ad-Publishers in the Client/Server model.

T_{MA} = Refers to the total time taken by advertisers to send, request and waits for publishing response in case of Mobile Agent approach.

$t(ad)_i = (t(a)_i - t(d)_i)$ for ith advertiser

m=Refers to the number of advertisers in the Mobile Agent approach.

$t(d)_i$ = Refers to the time that corresponds to dispatch the ith Mobile Agent from the ith advertiser.

$t(a)_i$ = Refers to the time that corresponds to the arrival of the ith Advertiser Mobile Agent at the Aggregator Module.

t_p = Refers to the processing time for collecting and aggregating the advertisements.

$t(pub) = (t(a) - t(d))$time to publish the advertisements

$t(d)$ = Refers to the time that corresponds to dispatch concurrent Aggregator Mobile-Agents to push advertisements at ad-Publisher sites.

$t(a)$ = Refers to the time that corresponds to the arrival of the Aggregator Mobile Agents at ad-publisher sites to publish the advertisements.

For example, if an advertiser needs to publish his advertisements at two different ad- publisher sites, the advertiser needs to make two separate requests that finally lead to variations in time depending upon speed and network bandwidth. Whereas, Mobile Agents are capable of publishing the advertisements to various ad-publisher sites using a single request by reaching the predefined ad-publisher sites.

– Bandwidth (Data Transfer Rate): Refers to the size of the advertisements (product information) that can be carried from the advertiser site to the ad-publisher site over a given period of time, which is expressed in bits per second (bps).

$$BW_{CS} = \text{File Size} / \left(etime - stime \right) \qquad (3)$$

$$BW_{MA} = \text{File Size} / \left(\left(atime - dtime \right)_{agg} + \left(atime - dtime \right)_{pub} \right) \qquad (4)$$

where,

BW_{CS} = Refers to the size of the advertisement that is carried from the advertiser site to the ad-publisher site within a given period of time for the Client/Server Model.

File Size=Refers to various advertisements size.

e-time =Refers to the end time at the ad-publisher site.

s-time=Refers to the start time at the advertiser requesting to publish the advertisements.

BW_{MA} = Refers to the size of the advertisement that is carried from the advertiser site to the ad-publisher site within a given period of time for the Mobile Agent approach.

File Size=Refers to various advertisements size.

atime=Refers to the arrival time of the Mobile Agents at the Aggregator module and ad-publisher site.

dtime=Refers to the dispatch time of the Mobile Agents from Advertiser site and Aggregator module.

- Theoretical Proof

From the theoretical analysis it can be proved that, the proposed framework benefits from the use of Mobile Agents that are dispatched in parallel to publish in all ad-publisher sites simultaneously when compared to that of the existing approach.

– Time Complexity for the Mobile Agent Approach

Here initially the framework starts with the serial dispatch of Mobile Agents from the set of advertiser to the aggregator module where each Mobile Agent carries information within a range of 10Kbytes to 1Mbytes respectively. The aggregator Agent adopts the concept of one-to-many or many-to-many mapping of advertisements and acts as the master Agent which aggregates the multiple request into single request and dispatches Mobile Agents in parallel which are supposed to perform publishing the ads at various ad-publisher sites. Hence the time complexity can be given as:

$$T(n) = O(m + n) \tag{5}$$

where,

m=Represents the number of advertisers.
n=Represents the number of ad-Publishers.

– Time Complexity for the Client / Server Approach

In this approach , the advertiser dispatches serial requests that employs a one-to-one mapping of advertisements for posting the advertisements at the set of ad-publishers sites. This model is simple and is slowing as the time taken to process each request is linear, as it accepts n inputs and produces n outputs. Where n represents the number of requests made by each advertiser.

Hence the time complexity can be given as:

$$T(n) = O(m * n) \tag{6}$$

where,

m=Represents the number of advertisers.

n=Represents the number of ad-Publishers.

In the above discussion we have made an assumption that, the number of advertisers and ad-publisher must be greater than two to show that there is huge variations in processing the request and time taken by two approaches.

Consider the case where the number of advertisers are same as the number of ad-Publishers then , the time complexity of the Mobile Agent approach is O(2n) whereas of the Client / Server model is O(n*n). Thus, from the theoretical analysis we can say that the proposed framework with concurrent Mobile Agents is more efficient than the existing approach that is used for posting the advertisements. The same has been proved practically and is shown below graphically.

RESULTS AND DISCUSSION

1. CASE I:

Deals with the total time taken for processing the number of requests made by the advertisers to publish the advertisements.

From the Fig. 8, we can say that, the time taken for processing the different requests from the advertiser is more in case of the Client/Server approach when compared to that of the proposed framework that uses Concurrent Mobile Agents for publishing the advertisements.

2. CASE II:

Deals with the publishing time for pushing the ads to different ad-Publisher sites.

From the Fig. 9, we can say that, as the number of ad-Publisher sites increases the performance differences between both the approaches is significant. The performance is improved by the use of concurrent Mobile Agents to push the advertisements on various ad-Publisher sites.

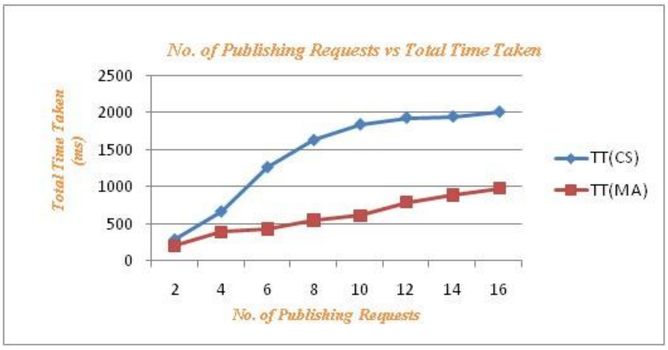

Figure 8: Graph showing the Total Time Taken for processing the Requests for both the approaches.

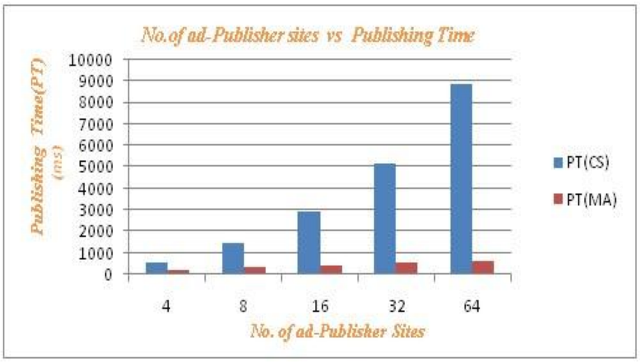

Figure 9: Graph showing the Publishing time taken by both the approaches.

3. CASE III:

Here we consider the data transfer rate to push the advertisements (product information) from the advertiser site to the ad-publisher site for different file size over a given period of time. From the Fig.10, we can say that, the high value of bandwidth allows faster transmission of data. The data transfer rate for various requests from the advertiser with different file size in case of the Client/Server approach is low when compared to that of the proposed framework with Concurrent Mobile Agents for publishing the advertisements.

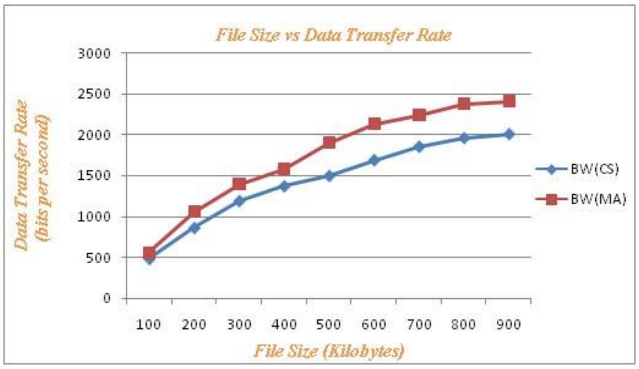

Figure 10: Graph showing the Data Transfer Rate to push the advertisements at the ad-publisher sites for both the approaches.

Finally, it is practically proved that the proposed framework is more efficient than the existing approach for publishing the advertisements to different ad-Publisher sites.

THE ADVANTAGE OF USING CONCURRENT MOBILE AGENTS FOR E-ADVERTISING

- The framework is more efficient when compared to that of the traditional client/server model used for publishing the advertisements. By inheriting the benefits of Mobile Agents , the advertiser request is processed in parallel that consumes less time for publishing the advertisements.
- The framework can change dynamically , where the aggregator stationary agent can create and dispatch as many numbers of Mobile agents at run time based on the number of ad-publisher site .
- The set of Mobile agents dispatched from the customer side can provide a personalized product service which reduces the customer time in searching the advertisements of their interest. It also deals with the issues related to one-to-many and many-to-many mapping between the advertisers and the ad-publisher sites.
- The Client/Server implementations are more suitable for small number of advertiser requests for publishing the advertisements.
- When the advertisement data size and number of ad-publisher sites for publishing increases, the performance of Client/Server implementation degrades and under performs.
- For situations in which large number of advertisements needs to be pushed and pulled from large number of ad-publisher sites, concurrent Mobile Agent implementations have been found to be more effective.
- The Concurrent Mobile Agent approach scales well as the number of advertiser requests to be processed and the number of ad-publisher sites to be visited increases.
- Mobile Agents are more reliable as the adveriser request processing does not depend on the continuous network conncectivity.
- The extra burden and time is reduced as the advertiser is relieved from repeated publishing requests when the network connection is disrupted.

OBSERVATIONS

1. Achievements of Case I:

   In this scenario we have conducted the experiment that deals with both the cases of one-to-many and many-to-many mapping of advertiser request (publishing request) to the ad-publisher sites. Here we conclude that the processing time required by the Client/Server model is twice that of the concurrent Mobile Agent approach .

2. Achievements of Case II:

   Parallel Implementation of Mobile Agents performs better than the sequential implementations of Client /Server to publish the advertisements. As the number of ad-publisher site increases , the client/server model under performs, whereas the mobile agent approach perform very efficiently and its publishing time reaches saturation point with slight variations . And is shown in figure 7 and from this we can say that the mobile agent framework works very efficiently.

3. Achievements of Case III:

   From the Experiment results, as the publishing time required by the Client /Server model (Case I) is twice that of the Mobile Agent approach.The data transfer rate is higher in case of Mobile agent approach than compared to that of the Existing approach for various file sizes.

4. Final Achievement:

   - We can conclude that, the proposed framework that makes use of Concurrent Mobile Agents perform better than the Client/ Server implementation for a given set of e-Advertising Application parameters. The performance evaluation of both the approaches is shown in figure (8,9,10).
   - Mobile Agents scale effectively as the number of advertiser request and ad-publisher sites increases. Scalability being one of the needs in network computing, we find that mobile Agents are appropriate technology for implementing e-Advertising applications.
   - Mobile Agent paradigm promises an efficient and effective technology.

CONCLUSION

The framework is built on the basis of Mobile agent technology, which supports the parallel processing of advertiser requests for the purpose of publishing the advertisements over a distributed ad-publisher sites.

The use of Mobile agents simulates the commercial activities of the advertiser, ad-publisher and the customer for publishing and purchasing the products and services. The benefit of the proposed framework is that it exploits the parallel processing of the advertiser request with higher efficiency and provides better support for both advertiser and customer.

This work proves to be a very efficient way in carrying out the task of publishing the advertisements without wasting much of the advertiser time in visiting various ad-publisher sites for publishing the advertisements and also provides a better solution to deal with one-to-many and many-to-many mapping of advertisers and ad-publishers. The use of Mobile agent overcomes the drawback of the traditional approach used for publishing the advertisements and also helps the majority of web users and advertisers to move towards a high level of automation.

REFERENCES

[1] Christoforos Panayiotou, George Samaras, Evaggelia Pitoura, araskevas Evripidou ”Parallel Computing Using Java Mobile Agents”. Department of Computer Science , University ofCyprus and Ioannina ,2013.

[2] Jyoti Tewari, Swati Arya and Prem Narayan Singh “Approach of Intelligent Software Agents in Future Development” International Journal of advanced Research in Computer Science and Software Engineeringvol.3, Issue-5, May 2013.

[3] Lianne Lee and Nandan Parameswaran “Intelligent Shopping Using Migrating Agents” Proceedings of the Twenty Sixth International Florida Artificial Intelligence Research Society Conference 2013.

[4] Yashpal Singh, Kapil Gulati and S Niranjan “Dimensions and Issues of Mobile Agent Technology” International Journal of Artifical Intelligence (IJAIA), Vol.3, No. 5, September 2012.

[5] Ajab Maheshwari, Dr. Pratosh Bansal ”Hybrid Approach of Client-Server Model and Mobile Agent technology to Drive an E-Commerce Application”. International Journal of Computer Science and Information Technology and Security ,Aug-2012,Volume-2.

[6] Ziming Zeng ”An Agent- Based Online Shopping System in E-Commerce”. Computer and Information Science www.ccsenet.org ,2009,Volume-2

[7] Yousry El-Gamal, Khalid El-Gazzar and Magdy Saed ”A Comparative Performance Evaluation Model of Mobile Agent Versus Remote Method Invocation for Information Retrieval”. World Academy of Science , Engineering and Technology, International Journal of Electrical , Computer, Electronics and Communication Engineering, Vol: 1, No.:3, 2007.

[8] Wenyu Qu, Hong Shen and John Sum Stochastic “Analysis of Mobile Agent Based e-Shopping” International Journal of Electronics Business Copyright 2005 Inderscience Enterprises Ltd. Vol. 3, Nos. ¾.

[9] R B Patel and K Garg “A Comparative Study of Mobile Agent and Client-Server Technologies in a

Real Application” Advances in Data Management , CSI 2005.

[10] Xubin He and Qing Yang “Performance Evaluation of Distributed Web Server Architectures under E-Commerce Workloads” Journal of Parallel and Distributed Computing, 2003.

[11] Yan Wang “Dispatching Multiple Mobile Agents in Parallel For Visiting E-Shops”. In the Proceedings of 3rd Conference on Mobile Data Management, IEEE Computer Society Press Jan. 8-11 2002 , Singapore , PP 53-60.

[12] Yan Wang, Kian-Lee Tan and Jian Ren ”A Study of Building Internet Marketplaces on the Basis of Mobile agents for parallel Processing”. World Wide Web: Internet and Web Information Systems ,2002, PP- 44-66, Volume-5.

[13] Ryszard Kowalczyk, Bogdan Franczyk, Andreas Speck ”InterMarket Towards Intelligent Mobile Agent e-Marketplaces” Proceedings of the Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS02) 2002.

[14] Roch H. Glitho, “Mobile Agents and Their use for Information Retreival: A Brief Overview and an Elaborate Case study”, IEEE Network January/February 2002.

[15] F. Dignum and C. Sierra, editors, “Agent Mediated Electronic Commerce” A European Agent Link Perspective, volume 1991 of Lecture Notes in Artifical Intelligence, pages 7897. Springer-Verlag, 2001.

[16] R. Kowalczyk and V. Bui. “ Constraint-based reasoning in e-negotiation Agents”. In F. Dignum and U. Cortes, editors, Agent-Mediated Electronic Commerce III, volume 2003 of Lecture Notes in Artifical Intelligence, pages 3146. Springer-Verlag, 2001.

[17] David Siew, Xun Yi, “Agent-mediated Internet Advertising” Second International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000) , 2000.

[18] Luis Moura Silva, Victor Batista, Paulo Martins, Guilherme Soares ”Using Mobile Agents for parallel Processing”. Department of Engenharia Informatica , University of Coimbra

[19] Danny B Lange and Mitsuru Oshima “Programming and Deploying Mobile Agents with Java”. 2nd Addison Wesley, 1998.

[20] Y.Aridov and D.Lang ”Agent design patterns: element of agent application design” In proceedings of Autonomous Agents 1998, pp108-115. ACM, 1998.

[21] Danny B Lange and Mitsuru Oshima ”Seven good Reasons for Mobile Agents”. Communication of ACM 1999, Volume-42.

[22] Johnny Wong, Guy Helmer, Les Miller, Vasant Honavar “SMART Mobile Agent Facility”. Department of Computer Science , University of IOWA.

[23] Hyacinth S. Nwana ”Software Agents: An Overview”' Knowledge Engineering Review, Vol. 11, No 3, pp. 205-244, October/November 1996.

[24] Rahul Jha and Sridhar Iyer ”Performance Evaluation of Mobile Agent for E-Commerce Applications”. Kanwal Rekhi School of Information Technology , Indian Institute of Technology Bombay, Powai, Mumbai - 400 076..rahul, sri@it.iitb.ac.in , http://www.it.iitb.ac.in

[25] M. Bakhouya, J. Gaber and A. Koukam ”Observations on Client-Server and Mobile Agent Paradigms for Resource Allocation” Laboratoire SeT, Universite de Technologie de Belfort-Montbeliard ,90000 Belfort, France ,bakhouya, gaber, abder.koukam@utbm.fr.

[26] Concordia, 1997, “Concordia: An Infrastructure for Collaborating Mobile Agents” Proc. of Workshop On mobile agents MA’97, Berlin, April 7-8th. LNCS 1219, Springer Verlag.

[27] Mitsuru Oshima, Guenter karjoth, Kouichi Ono “ *Aglets Specification 1.1*” Draft. IBM Corp. September 1998.