The Pattern as a Reusable Component to Adaptive Framework for Load Balancing Mobile Agent in Internet Banking

A.Meiappane [1], Dr. V. Prasanna Venkatesan [2], V. Jegatheeswari [3], B. Kalpana [4], U. Sarumathy [5]

[1] Department of Banking Technology, Pondicherry University
Pondicherry, 605 014, India

[2] Department of Banking Technology, Pondicherry University
Pondicherry, 605 014, India

[3 4 5] Student, Department of IT, Sri Manakula Vinayagar Engineering College
Pondicherry, 605 107, India

Abstract

In recent years, the Software Architecture of an Internet banking has become one of the most challengeable and risky fields of modern technology. The advent of the Internet has revolutionized the way banking is being done. In Internet banking, we analyzed some attributes such as usage, security, availability, load balancing and so on. Among those attributes Load Balancing is a serious problem in the Internet and it is necessary to build new systems incorporating load balancing as integral part of their design. The use of patterns, which is of a reusable component, is a good tool to help designers build load balancing systems. In this paper we are going to propose cluster load balancing technique by using mobile agent.

Keywords: *Software Architecture, Internet Banking, Load Balancing, Mobile Agent***.**

1. Introduction

The patterns play an important role in all fields such as e-commerce, eshopping [9] but in internet banking the patterns are not play major roles, so we are going do pattern in the internet banking. By using the patterns in the internet banking it provide various advantages such as reusability, availability, and also the performance will be improved[1].A framework is an integrated set of software artifacts (such as classes, objects, and components) that collaborate to provide a reusable architecture for a family of related applications. Framework*s* can be a very powerful means to reduce software cost and improve its quality, they can be hard to understand, select, learn, use, debug, and optimize. The Framework has many advantages:

- Extensible to support successions of quick updates and additions to address new requirements and take advantage of emerging markets.
- Flexible to support a growing range of multimedia data types, traffic flows, and end-to-end quality of service (QoS) requirements.
- Portable to reduce the effort required to support applications on heterogeneous OS platforms and compilers.
- Reliable to ensure that applications are robust and tolerant to faults.
- Scalable to enable applications to handle larger numbers of clients simultaneously.
- Affordable to ensure that the total ownership costs of software acquisition and evolution are not prohibitively high. [10]

1.1 Adaptive Framework

Adaptive Framework is a dynamic framework which monitors the running system and updates the architectural properties based on the changes in the system.

An adaptive framework which has three key elements namely, monitoring, decision making and re-configuring. This framework works with the help of an external monitoring and control mechanism.

An advantage of any adaptation based framework is that it provides a lot of reusable code which enables the developers to build applications more quickly [1].

1.2 Adaptive framework benefits

- Speed of implementation with repeatable architectural patterns and accelerators

- Flexibility to progressively transform to a simplified architecture one project at a time
- Choice of how to get started and who to partner with for business capabilities
- Cost Reduction through re-use of services and assets and through faster implementation
- Alignment of business and IT priorities for more effective results from solution implementation[11]

Here we propose an adaptive framework for Internet Banking to reduce the load balancing of server by using clustering.

2.Internet banking

Internet Banking refers to the provision of Banking Services through the medium of internet by the banks either by using its own Domain or through lease, by using this provision the banking services are utilized by the customers for their use & convenience. Online banking (or Internet banking) allows customers to conduct financial transactions on a secure website operated by their retail or virtual bank, credit union or building society.

2.1 Internet banking vital features

Net Banking has three basic features. They are as follows:

- The banks offer only relevant information about their products and services to the mass.
- Few banks provide interaction facility between the banks and its customers.
- Banks are coming up with arrangements of utility payments, like telephone bills, electricity bills, etc.

2.2 Advantages of Internet Banking

An internet banking account is simple to open and use. Internet banking costs less. You can access the information anywhere that you have access to the Internet. It makes your financial life much easier to manage.

- Pay Your Bills Online

You can use online banking to pay your bills. This will eliminate the need for stamps and protect yourself from the check being lost in the mail. Most banks will have a section in which you set up payees. You will need to fill out the information once, and then you can simply choose that profile every time you pay a bill online. If your bank will not pay bills online you may consider paying online through the company. Be careful since some of these companies may charge a convenience fee.

- View Your Transactions

Online banking allows you to access your account history and transactions from anywhere. This is the quickest way to check and see if a transaction has cleared your account. This can help you to find out the amount of a transaction after you have lost your receipt. It also allows you to find out about unauthorized transactions more quickly. This can help you to resolve the issues more quickly.

- Transfer Money between Accounts

Online banking also allows you to transfer money between accounts much more quickly. It is more convenient than using the automated phone service, and can save you a trip to the bank. When you apply or set up your online banking, be sure that all of the accounts you have at the bank are listed. This will make it easier to transfer money and make loan payments online.

- Protect Yourself Online

It is important to be careful when banking online. You do not want your safety or privacy to be breached. It is important to clear your cookies after each banking session, if you are at a public computer. Additionally you need to make sure that your password is long enough to prevent it from easily being hacked. Never give your online account information to someone who is not an authorized signer on your account.

3. Load balancing

Load balancing is a computer networking methodology to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. The load balancing service is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server.

One of the most common applications of load balancing is to provide a single Internet service from multiple servers, sometimes known as a server farm. Commonly, load-balanced systems include popular web sites, large Internet Relay Chat networks, high-bandwidth File Transfer Protocol sites, Network News Transfer Protocol (NNTP) servers and Domain Name System (DNS) servers. Lately, some load balancers evolved to support databases; these are called database load balancers.

3.1 Working mechanism of load balancer

For Internet services, the load balancer is usually a software program that is listening on the port where external clients connect to access services. The load balancer forwards requests to one of the "backend" servers, which usually replies to the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting backend servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports. Some load balancers provide a mechanism for doing something special in the event that all backend servers are unavailable. This might include forwarding to a backup load balancer, or displaying a message regarding the outage.

3.2 Various Algorithms to reduce load balancing

- Genetic Algorithm

Genetic Algorithm (GA) is to dynamically schedule heterogeneous task on heterogeneous system in a distributed environment to minimize total execution time. GA uses historical information to exploit the best solution and completes it process in three steps: Selection, Crossover and Random mutations. The GAlgorithm is only performed if there are more unscheduled tasks than processors.

- Fuzzy Logic Algorithm

This algorithm not only effectively reduces the amount of communication messages but also provides considerable improvement in overall performance such as short response times, high throughputs, and short turnaround times.

- Static and Dynamic Algorithms

Static algorithms

- Round Robin and Randomized Algorithms
- Central Manager Algorithm,
- Threshold Algorithm

Dynamic algorithms

- Central Queue Algorithm
- Local Queue Algorithm

The performance of these algorithms is measured by following parameters:

- Overload Rejection
- Fault Tolerant
- Forecasting Accuracy
- Stability
- Centralized or Decentralized
- Nature of Load Balancing Algorithms
- Cooperative
- Process Migration,
- Resource Utilization.

When number of requests increase from a particular site the response time from that website also increases, it's really important to properly handle the slow response time problem otherwise clients will get frustrated and will either refuse connection or will never visit that web site again. A website can handle slow response problem in variety of ways: by increasing server bandwidth (which is not feasible every time), using web proxy caching or mirror websites, answering only text and making use of monolithic or cluster web server. Among various solutions available to solve this slow response problem, cluster web server (when single domain name is served by more than one computer) is the best option. The most important concern of this strategy is to handle incoming load dynamically between all working units.

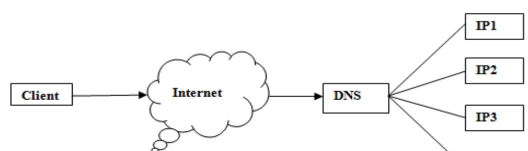

Fig.1:Existing system

Handling load dynamically is a very big challenge but degree of challenge magnifies when we are working with cluster technology.

In this study, we propose load balancing technique framework which consists of a single control server and multiple heterogeneous clusters.

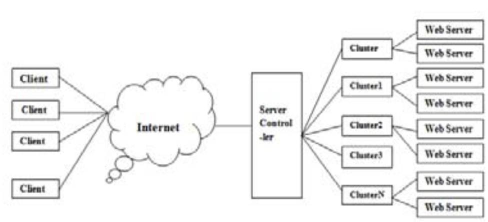

Fig.2:Hetrogeneous cluster

Using Mobile Agent we can distribute the load in a cluster, total load is calculated by queue length, the queue length is measured by the total number of processor in the queue

4. Mobile Agent

A mobile agent is a process that can transport its state from one environment to another, with its data intact, and be capable of performing appropriately in the new environment.

Mobile agents decide when and where to move. When a mobile agent decides to move, it saves its own state, transports this saved state to the new host, and resumes execution from the saved state

4.1 Comparison: Client/server and mobile agent

By using mobile agent-based we can reduce traffic/congestion on the network. In client/server network, communication is usually done by exchanging messages between the client and the server. For example, if a client needs a particular service, he sends a request message to the server acquiring the needed service. In the other hand, server responds to the client by sending exchanging more messages which may increase network traffic. Because agents are smaller in size and when they reach the server, there is no connection anymore until the agent finishes his task and after he is ready for reconnection.

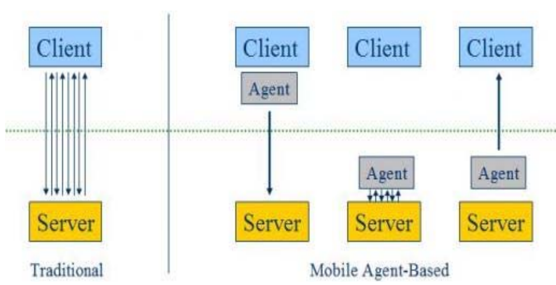

Fig.4:Traditional vs mobile baseed agent

4.2 Advantages which mobile agents have over conventional Client /Server agents

- Computation bundles convert computational client/server round trips to relocatable data bundles, reducing network load.
- Parallel processing asynchronous execution on multiple heterogeneous network hosts
- Dynamic adaptation actions are dependent on the state of the host environment
- Tolerant to network faults - able to operate without an active connection between client and server.
- Flexible maintenance to change an agent's actions, only the source (rather than the computation hosts) must be updated

4.3 Mobile agent versus conventional client server

Fig:3 shows the comparative performance between the agent and conventional client/server requests as the number of requests increases from one to eight, the performance of both approaches when sending a single command is roughly the same. As we increase the number of commands in each experiment, the total completion time for the agent barely varies while the completion time for the client/server application increases rapidly as expected.

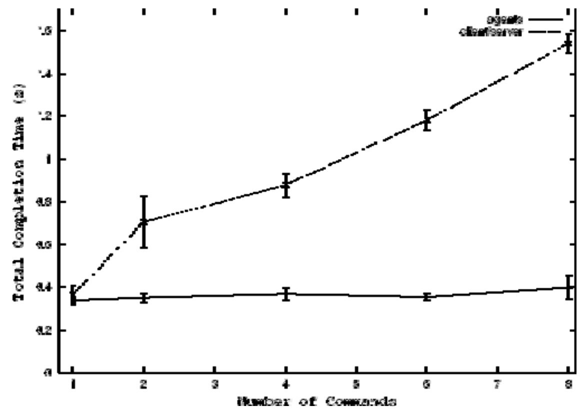

Fig.3:Comparison of mobile agent and client server

4.4 Load Balancing with Mobile Agent

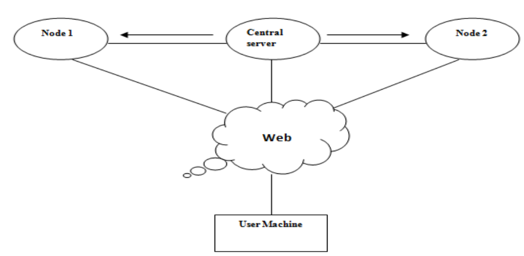

Fig.4:Mobile agent network

4.4.1 Response Time Evaluations

Fig: 4 show a model of a mobile agent network which depicts both the network architecture and the mobile agent behavior. In the network architecture, the processing part is represented as a set of nodes, such as "Server1", and they were capable of hosting and providing resources to the mobile agent to make it execute its task. The network communication is represented by arrows that connect the nodes and support mobile agent migration and communication with the user by sending result messages in Fig: 5

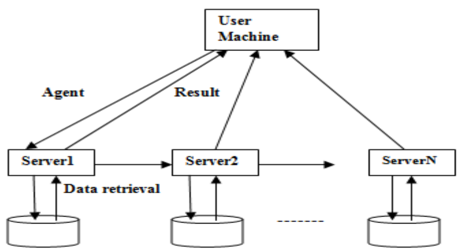

Fig: 5 Mobile Agent Traversal

The user interacts with the mobile agent network by initiating a mobile agent at the user machine. The agent then carries the user's search criteria to request required web documents at remote servers. The user's request and creation of the agent were random events; the agent migrates with a user defined itinerary to the destination node to perform the data retrieval task, which is also a stochastic process. So, the mobile agent can be considered as a network communication packet in the network queuing system that should be served at the server.

Response time is a typical parameter for measuring the performance of a queuing system. It is defined as the time interval between when a request is sent and when a response is returned. Response time is commonly used to evaluate parallel and distributed processing in telecommunications. We also adopted this method for our work with a mobile agent network.

The response time for a mobile agent network is expressed as:

$$T = t_{process} + t_{migration} \qquad (1)$$

- T: mobile agent response time
- $t_{process}$: time taken to perform tasks at nodes
- $t_{migration}$: time taken to migrate within the network

Depending on the situation, $t_{process}$ will change. The factors that affect $t_{process}$ include the size of the mobile agent, the query size and complexity, and the size of the records inside the database. $T_{migration}$ is determined by the agent size and by the network bandwidth between agent hosts .

With these factors taken into consideration, the above Equation is expanded as:

$$T = t_{agent} + \sum_{i=1}^{N} (t_{process(i)} + t_{agent}(i, i+1)) + t_{N+1} \qquad (2)$$

With reference to the generic mobile agent network model (the agent hosts were numbered from 1 to N), the parameters in Equation are expressed below:

- T: Mobile agent response time
- t_{agent} : Time taken to create a mobile agent and migrate it to the first host
- $t_{process}$ (i): Duration of the mobile agent execution at i_{th} server
- t_{agent} (i, i+1): Time taken by the mobile agent to travel from node i to node i+1
- t_{N+1} : Time taken by the mobile agent to travel back from the last host and destroy itself

The mobile agent-based searching was tested with a network configuration consisting of three agent hosts. Compared with our mobile agent system, the equivalent client-server system as shown Fig.6

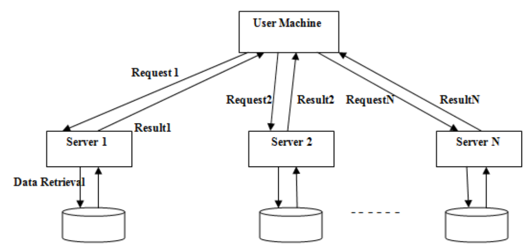

Fig:6 Client-Server Architecture

Since the client-server paradigm is not synchronous, the user has to perform the search by himself, key in the search criteria, and wait for the request to be sent back from the server. If the user wants to search the three data servers one by one for required documents, he has to perform the same search operations three times. Those operations include opening the E-learning application, entering the search criteria, clicking the "start" button, etc.

The client-server network configuration is shown in Figure. Its response time is expressed as follows.

$$T = \sum_{i=1}^{N} (t_{query}(i) + t_{transmission}(i, 0)) + \sum_{i=1}^{N} t_{user}(i) \qquad (3)$$

- T: Client-server response time
- t_{query} (i): Time taken for the server i to perform the query on its database
- $t_{transmission}$ (i, 0): Time taken for server i to send query results to the user machine
- t_{user} (i): Duration user performs operations during each search session

The client-server-based search was tested with the network configuration consisting of three servers, and the response time for each server was recorded in the table below. Since t_{user} is subject to the user's typing speed and reaction ability, we took t_{user} as 10 seconds for the average time a user takes to perform a search session. These values were summed up to get the response time to calculate the response time T.

The database used in testing was the same as the one used to test the mobile agent system. The query and query results were also the same. Thus, we performed a response time comparison between the two systems with the equivalent hardware and software components .

From the data collected, it was shown that, as the records in the database increase in size, the response time for both the mobile agent and the client-server systems increased. However, the response time of the client-server system was more sensitive to the database record size. Figure 9 shows the response time versus record size comparison for the two systems.

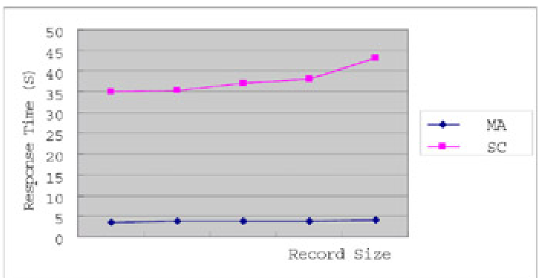

Fig.7: Comparison Graph

4.4.2 Response Time Comparison

It shows that the response time of the mobile agent-based search is much faster than that of the client-server search. This is mainly due to the asynchronous and autonomous nature of mobile agent. Mobile agent performs information searching tasks alone once a user creates it, and completes the task autonomously, without interaction or communication with the user. The user doesn't need to monitor the agent during its work, which brings the user tremendous time savings and reduces communication costs.

From our experiment data, mobile agent saved an average of 30 seconds compared to completing the same task with client-server system. The efficiency in performance can be attributed to the following reasons:

Mobile agent is not a stand-alone process. It is a thread that needs to work together with other programs in the agent host to complete the task. Hence, it is flexible and small in size. In our system, the search agent is only around 10K with small variation depending on user query size. Hence, the amount of time taken to transport mobile agent is small.

The mobile agent moves query computations to the data server, the repetitive request/response handshaking involved in the client-server system is eliminated. The mobile agent performs data selection at the remote server; only the selected documents are sent to user. In the client server system, intermediate results and information need to be passed through the network between server and client. Hence, mobile agent reduces network traffic and improves performance in terms of response time.

5. Conclusion and Future work

We have examined that, by using mobile agent the load can be balanced in the internet banking based on the request made by the client and so the response time is increased. The demerit of mobile agent is Security. Security risk in mobile computing environment is two parts. First of all a malicious mobile agent can damage a host. On the other hand malicious host can be tamper with the functioning of the mobile agent. Most of the experts say that, this kind of risk is difficult to deal with. Some server will even try to steal credit card number from the mobile agent. Mobile agent tools are still new and they have security bugs and vulnerabilities. Since Mobile agents are not so mature technology and most agent development tools are alpha or beta versions. Although an agent's ability to travel throughout the network introduces fault-tolerant properties to the security tool, it also expose the agents to new security threats and risk is that host based security do not encounter. The agent host environment must be installed and maintained on each node.

6. Reference papers

[1] Xu Zhang and Chung-Horng Lung "Improving Software Performance and Reliability with an Architecture-based Self-Adaptive Framework", 34th Annual IEEE Computer Software and Application Conference, 2010

[2] George H.Fairbanks, "An Architecturally-Evident Coding Style: Making Your Design Visible in Your Code", ACM, 2010.

[3] Neil B.Harrison and Paris Avgeriou, "Pattern-Based Architecture Reviews", IEEE, 2010.

[4] Hui Li, Jingiun Zhang and Yuejuan Chen "Aspect-Oriented Modeling in Software Architecture Pattern Based on UML", IEEE, 2010.

[5] Julie Street Fant,"Building Domain Specific Software Architecture from Software Architectural Design Patterns", ACM, 2011.

[6] Vallidevi Krishnamurthy and Chitra Babu "Effective Self Adaptation by Integrating Adaptive Framework with Architectural Patterns", ACM, 2010.

[7] B.Fernandez and Rouyi Pan, "A pattern language for security models", Eduardo, Plop Conference, 2001.

[8], Zhu Zemin,"Study and Application of Patterns in Software Reuse" , IEEE, 2009

Web reference:

[9] Gustavo Rossi, Fernando Lyardet, and Daniel Schwabe,"Patterns for E-commerce applications". http://www.google.co.in/#hl=en&output=search&sclient=psyab&q=Patterns+for+Ecommerce+applications&pbx=1&oq=Patterns+for+E-ommerce+application

[10] Douglas C.Schmidt, Aniruddha Gokhale, and Balachandran Natarajan,"Frameworks: Why They Are Important and How to Apply Effectively" http://www.google.co.in/#hl=en&cp=63&gs_id=i&xhr=t&q=Frameworks%3A+Why+They+Are+Important+and+How+to+Apply+Effectively&pq=patterns+for+e-commerce applications

[11] Deepti Sharma and Archana B.Saxena, "Framework to Solve Load Balancing Problem in Heterogeneous Web Server"

http://www.google.co.in/#hl=en&sclient=psy-ab&q=Framework+to+Solve+Load+Balancing+Problem+in+Heterogeneous+Web+Server&pbx=1&oq=Framework+to+Solve+Load+Balancing+Problem+in+Heterogeneous+Web+Server

A. Meiappane is a research scholar of Pondicherry University, Pondicherry, India. He has received his M.Tech. in Computer Science & Engg. From Pondicherry University (Central), Pondicherry, India. Currently he is working as Associate Professor in Sri Manakula Vinayagar Engineering College, Pondicherry India. His different research areas are Computer Networks, Distributed Systems and Software Architecture Patterns.

Dr. V. Prasanna Venkadesan received his M.Tech. and PhD degree in Computer Science & Engineering from Pondicherry University, Pondicherry, India. He is currently working as Associate Professor, Dept. of Banking Technology at Pondicherry University, Pondicherry. His research interests are in the areas of Software Architecture, Object-Oriented Systems, Multilingual-based systems, Programming Languages, Banking Technology Mgt.

V. Jegatheeswari is pursuing her B.Tech. degree in Information Technology, Sri Manakula Vinayagar Engineering College, Pondicherry University, Puducherry, India. Her research area includes Architecture Patterns and Computer Networks. She presented in various National and International conferences.

B. Kalpana is pursuing her B.Tech. degree in Information Technology, Sri Manakula Vinayagar Engineering College, Pondicherry University, Puducherry, India. Her research area includes Software Engineering and Architecture Patterns. She presented in various National and International conferences.

U. Sarumathy is pursuing her B.Tech. degree in Information Technology, Sri Manakula Vinayagar Engineering College, Pondicherry University, Puducherry, India. Her research area includes Architecture Patterns and DBMS. She presented in various National and International conferences.