FAULT TOLERANCE USING CREDENTIALS MANAGEMENT IN ONLINE TRANSACTION APPLICATION

L. Javid Ali
Research scholar, Sathyabama University, Department of Computer Science and Engineering, St.Joseph's College of Engineering, Jeppiaar nagar, OMR, Chennai, Tamilnadu, India
Email: javidinaug77@gmail.com

G.S. Anandhamala
Professor, Department of Computer science and Engineering, St.Joseph's College of Engineering, Jeppiaar nagar, OMR, Chennai, Tamilnadu, India
Email:gs.anandhamala@gmail.com

Abstract

Web applications play a vital role in the IT field for satisfying the web customer. The customer always depends on the online transaction processing system. The web application has various forms which gives a complete service to the customer. These various forms have options that are used to satisfy the customer's needs because of the attraction over web sites existing in the global market. The traditional web pages will be closed from the current session whenever the customer selects an improper option because of single sign-on property. Selection of wrong option that is not suitable for the current session will lead to reliability problem. If the same user needs the same service, again he has to navigate from home page to the required page, thus adding up extra burden on customer. The customer session should be maintained properly, so that the customer's satisfaction is retained over the online web application. The existing system classifies the user with their access level and also their fault level. The main objective of the proposed work is to manage the credential in all levels in order to keep the valuable customer for a long time of access in the current session. The credential management and session management are used to manage a multilevel credential from web client to web resource level and vice versa. The options selected by the customer can be classified based on the fault and type of access. The credential management also performs the maintenance process for fixing the fault tolerance level to the web user. A complete log is recorded to trace the overall process in the online transaction processing**.**

Keywords**:** Online Processing; Fault Tolerance; Reliability; WWW; Web Applications.

1. Introduction

Online transaction management is a method which is used to share the company resources to the global market. The customer uses an E-Commerce in accordance with websites for placing the order and performing the payment over the product. In traditional way of purchasing a product by the customer has spent long time for complete the entire purchase process because of the unavailability in E-Commerce. This issue has been solved by introducing an online transaction or shopping which supports the customer in all access level. There are various customers who can use the online transaction system with various needs related to the web sites. Normally all users cannot purchase the product in online with in a particular application. The fault identification and classification method has been implemented in order to categorize the user based on their access level and fault level. The customer may select an improper selection will leads the session time out. The customer can be categorized into basic user, normal user and valuable user. The basic user and normal user will perform only a specific application selection whereas the valuable customer performs all the operation over the online application. The tolerance level is fixed to the valuable customer in order to expand the session time. The main objective of the tolerance fixation is to give an importance to the valuable customer who spends more time to perform and complete the processing with high reliability. The mobile agents in the distributed computing environment suffer from threats like security, crashes, unavailability of resources and congestion in the network. These threats are handled by implementing the checkpoint with antecedence graphs in order to achieve the fault tolerance and also to improve the system performance in the mobile agent systems [1]. The recovery manager for enterprise network suffers an overhead to execute and recover from the fault over the agent-based application. This problem can be overcome by implementing a fault tolerant protocol for achieving the objectives like agent's execution, communication and migration [2]. The failure recovery between storage node and coverage node is implemented using the framework which contains three modules such as Process

Execution Module, Network Module and Checkpoint Management Module. This framework cannot tolerate a multiple faults without using the checkpoint processor [3]. The pessimistic log-based rollback recovery protocol has been introduced to restart and re-execute the failed process. RADIC (Redundant Array of Distributed Independent Controllers) is architecture especially for fault tolerance over inter-process communication. This approach suffers a problem in fault tolerance middleware which leads a less speed in scalability [4]. A novel fault tolerance mechanism has been implemented with various features for monitoring an organization such as fast failure detection, reducing the monitoring managers, consistent failure detection and execution with actions [5]. FANTOMAS (Fault-Tolerant approach for Mobile Agents) is a framework which provides the fault tolerance support over web application. This framework also preform operations like flexible adaptive and robust on dynamic agent domains [6]. The fault tolerance approach is used to perform an operation with the help of checkpoint which stores the state in a stable storage. The fault tolerance in the distributed systems can be achieved by resuming the already exists global state in the storage [7]. There are seven main reasons to start a mobile agent in fault tolerance they are reduction of network load, elimination of network latency, encapsulation of protocol, asynchronous execution, dynamic adaptability, heterogeneity, robust and fault tolerance in nature. The above reasons are not enough to achieve the maximum reliability in the applications [8]. The information processing system suffers a threat on enterprise assets and policies, so this can be avoided by using efficient information processing systems [9]. The web client gets the access permission only from the trusted web sites in order to prevent the flooding attacks [10]. The existing fault tolerance technique uses various methods with different parameters which are targeted on the mobile agent based fault tolerance system. Nowadays the user can use the web sites for everything like booking an air/train ticket, online shopping etc. The main objective of the proposed system is to provide an efficient way of maintaining and managing the users and their session using fault tolerance with credential management. The credential management manages the user level credential, browser level credential and captcha level for achieving maximum tolerance level to the valuable user. The session is maintained properly with various user access levels, and then only the tolerance level will be assigned to the customer. The fault count is maintained based on the type of access because the valuable user never gets the session time out and service unavailability. The log maintains records for all the information related to the fault tolerance over the online transaction application. The overall objective of the proposed technique is to assign the tolerance level to the valuable customer who spent the session with high availability

2. Basic Concepts of Credential Management

2.1. *Validation*

2.1.1 *Client side validation*

The client side validation of input will provide better user experience and also reduce the network traffic. The input can be restricted by limiting the size, type of characters allowed and type of data allowed. The special characters like single quotes, double quotes, commas, semi-colons, etc., must be avoided by sanitizing the input at the time of entry.

2.1.2 *Server side validation*

There are possibilities that the malicious user can bypass the client side validation and submit data to the server side application. The sanitization and validation of the data has to be done before processing the input data. A server side validation of the submitted data is more important that will prevent most of the attack on application and database

2.2 Authentication

Authentication is one of the best practice methods that the input comes from the trusted user or source. The password is one of the common authentication method used in all kind of application. User authentication has to be done while accessing the services in the web application where data will plays a major role in access.

2.3 Authorization

The user may have different credentials, based on the authentication method in which they are allowed to access and perform operations on a particular service. The data entered at the time of registration process should be verified using the following authorization procedures:

2.3.1 Email Authorization

The email address entered at the time of online submission has to be verified by sending and verifying of mail to that email address for confirming that the request came from the owner of the specific email id. If the user may not confirm the email address within a specific time of interval, the already entered data will be discontinued for further process.

2.3.2 Mobile Authorization

The increasing utilization of online services provided the gateway which helps to perform a financial transactions in online by many users and their count increases day by day. The mobile number authentication can be done for those who are registering their mobile number at the time of registration or in later stage. The integration of SMS gateway with the web application will help to send security passwords for approving the online financial transaction, etc.

2.4 Verification

There are possibilities that the server side application can be attacked by using an automated data submission process. The online submission of data has to be ensured in the client side using the security key that the request comes from the human interaction and not from any automated process. The security key can be generated using random generation of text, numbers or calculations. This will ensure the data are that comes from the user.

3. Architecture of the proposed system

Fig. 1 shows the architecture diagram of the proposed system. The web client uses required credential in different levels to access resources from the web server. The browser level information and its credentials and user credentials are managed by an appropriate level to achieve a maximum reliability. Extra level of security can be achieved by using captcha management in order to validate authentication. The session are managed to

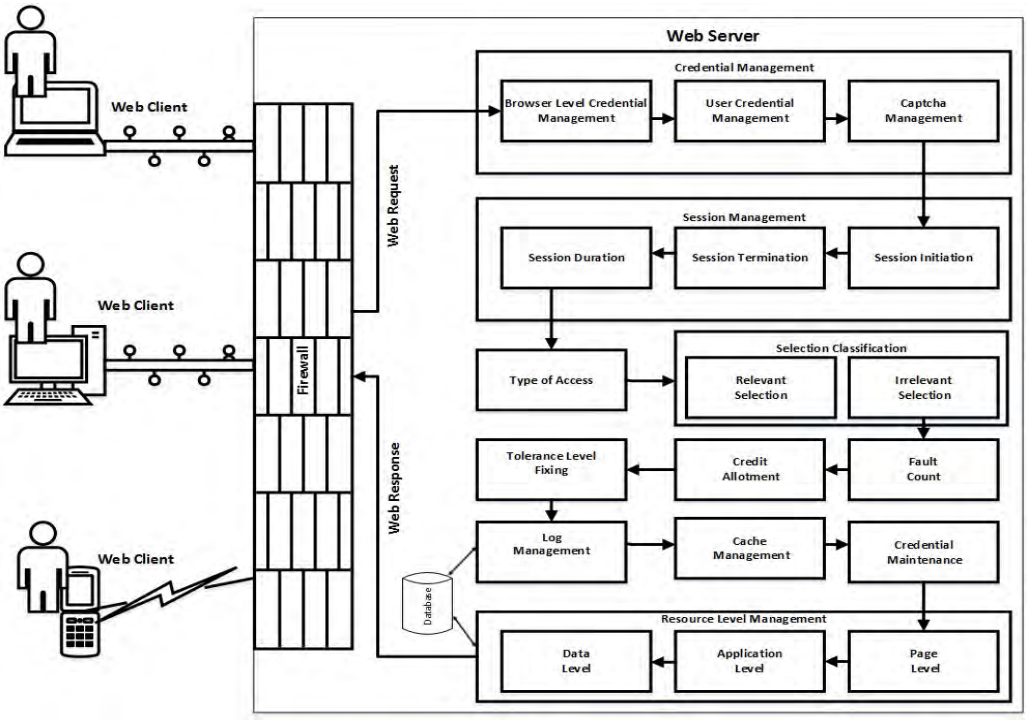

Fig 1. Architecture diagram of the proposed system

categorize and fix the tolerance level for the customer. The session duration can be identified through the starting and finishing session time. The tolerance level is fixed based on the activeness and how long the user uses the resource. This level also consists how much of access level over expensive resources and valuable operations are performed by the user. The user can be categorized based on the type of access they are, sensitive user and insensitive user. The customer may access a web resource by selecting predefined options which exists in the web application. These selections can be categorized into relevant and irrelevant selection type. In relevant type of selection the user can able to select what they actually need so, the tolerance level will be set to high value whereas the irrelevant selection will get minimum tolerance level. The fault count is maintained in order to allocate a credit to the customer for accessing exact resources from the web server. If the user has a high tolerance level then the fault count will be incremented to extend the session time. The insensitive user may quit the web application as soon as the irrelevant selection has been selected. The application level credential management is used to keep the information within the boundary. The credential information is restricted only to a particular web page will be recorded in a log. The database level credential is also managed with in a relevant boundary. The log is maintained to trace and store all information exists in all level. The credential

maintenance are used to keep all information related to the user along with credential for an efficient processing of web application.

4. Algorithm

Determine the client credential list as $C_{L;}$

Let C_{BL}, C_{UL}, C_{CAP} are credential levels such as Browser Level, User Level and Captcha respectively;

while stopping condition not true do

for each $C_{BL} \varepsilon C_L$ do

for each $C_{UL} \varepsilon C_L$ do

for each $C_{CAP} \varepsilon C_L$ do

Calculate the Threshold and set to $ft(C_{BL}, C_{UL}, C_{CAP})$;

end

end

end

select the n_T of the highest threshold as H;

for each $C_{BL} \varepsilon H$ do

for each $C_{UL} \varepsilon H$ do

for each $C_{CAP} \varepsilon H$ do

Create an n_B clones of $C_{BL;}$

Create an n_U clones of $C_{UL;}$

Create an n_C clones of CCAP;

Create a user list as UL;

Mutate the n_B clones and add to set H_1;

Mutate the n_U clones and add to set H_2;

Mutate the n_C clones and add to set H_3;

end

end

end

for each $C_{BL'} \varepsilon H_1$do

for each $C_{UL'} \varepsilon H_2$ do

for each $C_{CAP'} \varepsilon H_3$ do

Calculate the threshold as set to $ft(C_{BL'}, C_{UL'}, C_{CAP'})$;

end

end

end

for each $C_{BL'} \varepsilon H_1$do

for each $C_{UL'} \varepsilon H_2$ do

for each $C_{CAP'} \varepsilon H_3$ do

Calculate the session and set to $fs(C_{BL'}, C_{UL'}, C_{CAP'})$;

end

end

end

fault=0;

set the maximum threshold to the users as maxT;

set the Session value to the users as maxS;

L1: if $ft(C_{BL'}, C_{UL'}, C_{CAP'})$ > maxT then

if $fs(C_{BL'}, C_{UL'}, C_{CAP'})$ > maxS then

```
        for each user ε UL do
       find out the selection type i.e. relevant or irrelevant;
      if Selection_Type==’relevant’ then
       continue;
       else if Selection_Type==’ irrelevant’ then
       Session close;
Exit(0);
else if Selection_Type==’ irrelevant’ and UserLevel=
       ’Advanced’ then
if  --maxT != 0 then
records the statistical information into the log file;
keeps the information into the cache at the web client and
       server;
Maintains a credential of the web application in a
    particular session;
end
goto L1;
       end
     end
  end ;
end
```

5. Analysis of the proposed system

The analysis of the fault tolerance using credential management are described as follows, the web user without any classification **U** is referred in 1.

$$U = \begin{pmatrix} U11 & U12 & ..U1m \\ U21 & U22 & ..U2m \\ U31 & U32 & ..U3m \\ & \vdots & \\ Un1 & Un2 & ..Unm \end{pmatrix} \quad \ldots 1$$

The user classification **Uij** is based on the access level which is referred in the equation 2.

$$User\ Classification = \begin{cases} Uij = Basic, user\ \varepsilon\ U\ and\ Access\ Level = ' Basic' \\ Uij = Normal, user\ \varepsilon\ U\ and\ Access\ Level = ' Normal' \\ Uij = Advanced, user\ \varepsilon\ U\ and\ Access\ Level = ' Basic' \\ false\ , otherwise \end{cases} \quad \ldots 2$$

The credential are maintained for the users **Uij** with request level are identified based on Browser Level, User Level and Captcha Level which is referred in 3.

$$Credential\ Management = \begin{cases} Uij\ ,\ \ Request\ Level = \{BL \cup UL \cup Ccap\} \\ false, otherwise \end{cases} \quad \ldots 3$$

The tolerance of an advanced user can be fixed based on the type of selection either it can be relevant or irrelevant. If the selection type is irrelevant then the current session of the user will be closed otherwise the threshold should be fixed. This specification is referred in 4.

$$Selection\ Type = \begin{cases} Selection = relevant, \quad Fix\ the\ Tolerance\ Level\ \uparrow \\ Selection = irrelevant, \quad Session\ Time\ out \\ false, otherwise \end{cases} \quad \ldots 4$$

The tolerance is calculated with fitness value to the session management which is referred in 5.

$$\Upsilon = \begin{cases} fs(Cbl, Cul, Ccap) > 0, & Session\ Continue \\ fs(Cbl, Cul, Ccap) < 0, & Session\ Timeout \end{cases} \quad \ldots 5$$

The level of tolerance is identified based on the fitness value to the session management with user credentials which is referred in 6.

$$fs(Cbl, Cul, Ccap) = \begin{cases} Tolerance\ Level\ \Upsilon = high, user\ \varepsilon \{ AL\ and\ SL\ and\ ST \} \\ Tolerance\ Level\ \Upsilon = low, otherwise \end{cases} \quad \ldots 6$$

6. Conclusion and Future Work

The online transaction application gives important features to the customer who purchases anything with the support of internet. If the user needs a web resources by using the login based authorization technique. There are various ways to verify and validate the customer they are basic user and password, captcha validation etc., Traditional technique lacks in customer retention because the customer may leave from the current session by following single sign-on property. If a customer selects an improper option in the web page will leads the session time out. If a user needs the same service, once again he has to login on and navigate to the particular page for further processing. The main objective of the proposed technique is to retain the customer by applying fault tolerance using credentials which exists in various levels from web client end to the web server resource end. The proposed technique uses a credential management which manages the credential in the online transaction application. The session should be maintained properly to the valuable customer with fault tolerance level. The customer can spend long time in the resource access until the tolerance level gets elapsed. The credential and log maintenance are used to keep the detail of user session in a proper level. In future this technique can be extended to the web services with advanced technological implementation.

References

[1] Rajwinder Singh and Mayank Dave,"Antecedence Graph Approach to Checkpointing for Fault Tolerance in Mobile Agent Systems (2013). IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 2, FEBRUARY 2013, Pp. 247-258.

[2] Taha Osman, Waleed Wagealla, and Andrzej Bargiela (2004). An Approach to Rollback Recovery of Collaborating Mobile Agents . IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 34, NO. 1, FEBRUARY 2004,Pp 48-57

[3] Dipali B Parase,Sulabha Apte (2012). Handling Multiple Processor Failure Using Diskless Checkpointing Approach. International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 9,ISSN: 2278-0181,2012,Pp 1-4.

[4] Marcela Castro, Dolores Rexachs and Emilio Luque (2012). RADIC-based Message Passing Fault Tolerance System , IARIA, ISBN: 978-1-61208-237-0, ADVCOMP 2012, Pp 59-64.

[5] Jinho Ahn (2010). Fault-tolerant Mobile Agent-based Monitoring Mechanism for Highly Dynamic Distributed Networks. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 3, May 2010,ISSN (Online): 1694-0784,ISSN (Print): 1694-0814,Pp 1-7

[6] Hodjatollah Hamidi,Abbas Vafaei and Seyed Amirhassan Monadjemi (2010). Evaluation and Checkpointing of Fault Tolerant Mobile Agents Execution in Distributed Systems. ,JOURNAL OF NETWORKS,Vol. 5, No. 7, 2010, ACADEMY PUBLISHER, doi:10.4304/jnw.5.7.800-807, Pp 800-807.

[7] Rachit Garg, Praveen Kumar (2010). A Review of Checkpointing Fault Tolerance Techniques in Distributed Mobile Systems . International Journal on Computer Science and Engineering, Vol. 02, No. 04, 2010,ISSN : 0975-3397, Pp 1052-1063.

[8] Danny B. Lange and Mitsuru Oshima .Seven Good Reasons for Mobile Agents, Vol. 42, No. 3 COMMUNICATIONS OF THE ACM,Pp 1-2.

[9] Dianxiang Xu, Senior Member, IEEE, Manghui Tu, Michael Sanford, Lijo Thomas,Daniel Woodraska, and Weifeng Xu, Senior Member, IEEE (2012). Automated Security Test Generation with Formal Threat Model. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 4, JULY/AUGUST 2012, Pp 526-540

[10] Sanjeev Khanna, Santosh S. Venkatesh, Member, IEEE, Omid Fatemieh, Fariba Khan, and Carl A. Gunter, Senior Member, IEEE, Member, ACM (2012). Adaptive Selective Verification: An Efficient Adaptive Countermeasure to Thwart DoS Attacks",IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 20, NO. 3, JUNE 2012, Pp 715-728.