Failure Prediction in Distributed Network

P. Madaalasa, S.Akshhayaa, N. Sairam
School of Computing, SASTRA University,
Tanjavur, Tamil Nadu, INDIA
madaalasapanyam@gmail.com
akshhayaakrishnan.94@gmail.com
sairam@cse.sastra.edu

Abstract

In order to improve the reliability and availability in distributed networks, the occurrence of failures are to be predicted. Fault Prediction aims at predicting failure before they occur in Distributed Systems. This is possible by closely observing the behavior of the network. Network packets get latest information of network for every instant. The proposed work uses standard method to analyze network packets and simple mathematical calculations and timed automata in order to estimate Time-To-Failure (TTF). This method is able to detect only observable faults.

Key Words: Reliability, Availability, Distributed network, Failure, Fault, Failure prediction, Timed Automata, Time-To-Failure (TTF), Observable failure.

I. INTRODUCTION

Most of the networks, now-a-days, are distributed networks (Internet, grid and even latest model cloud is also a distributed network basically). These distributed networks become unreliable and unavailable under certain circumstances because of failures. Once network becomes unavailable, then it also loses reliability, consistency and data integrity. If failures are prevented, users can able to escape from results of lack of above quality attributes.

To avoid failures or to take precautions to prevent them before they occur, failures should be predictable. The prediction of failure can be done by observing the network behavior. If any abnormal behavior is found, that can be treated as a fault. We should keep track of this observed fault for further symptoms of the failure.

The paper structure is as follows: Section 2 describes classification of failures in distributed systems. Section 3 provides detailed system model. Section 4 contains results and discussions. Section 5 discusses performance analysis, section 6 contains future work and finally section 7 gives conclusion.

II. BACKGROUND

We can define a failure as a resource turning into useless resource at a certain point of time. There are basically two kinds of failures i.e. observable and non-observable. Observable failures are failures those produce some symptoms or signs before they occur. Non-observable failures occur all of a sudden.

For example, if a person observes rise in temperature and throat turning in to sore, it may be fever and gets severe by certain time. This resembles observable failures. And non-observable failures resemble like accidents, occur suddenly without symptoms before occurring. In both the cases, damage is same but the ways they occur differs.

The proposed method is completely based on the way how a failure occurs. The failure prediction model can only predict the observable failures but not the non-observable failures.

A network inevitably needs a communication layer for communication among the nodes in the network. Every node needs to communicate through packets. These packets carry the current status of network along with them. If we concentrate on packet information we can get the latest information about the network status.

A failure is always preceded with faults in the case of observable failures.

Fig 1: Failure time line

As we can observe in fig 1, time gap between faults keeps on decreasing as time gets closer to failure time. To get the latest information about network status, network packets are better because they give information about nodes of the network for each instance.

A. Related works:

In the literature, (Roberto Baldoni et. al) proposed Failure prediction technique in mission critical distributed systems by using CASPER architecture. Faults are predicted in real time discrete event systems using timed automata by (Ahmed Khomsi et. al). A method, that can predict faults automatically in large scale distributed systems has been proposed by (Marco Canini et. al). (Andrew W.Williams et. al) have developed a

technique which can predict faults using black box prediction in distributed systems. (Shu-ping Chang et. al) have proposed predictive failure management for distributed systems. A mechanism of estimating time to failure in distributed systems is proposed by (Saurabh Bagchi et. al). (Siva Perraju et. al) specified fault tolerance in mission critical systems using I/O Automata. A review report is given on fault tolerance aspects in Distributed Systems by (Yevgeniy Gershtey et. al). (Arvind kumar and Ramshakar Yadav et. al) investigated different techniques of fault tolerance in real time distributed systems.

B. Distributed systems:

A network or a computing network is known as 'distributed' when computer programs and data reside in numerous computers interconnected and are distributed geographically. This model is more efficient than centralized where the computing is based on only one system.

Many kinds of distributed networks are available. Among them, client and server model is popular. A server and many clients make a distributed network in this model. The principle of client and server model is communication held as request and reply.

C. Timed Automata:

Timed automaton is an extension of finite automaton with finite set of real valued clocks. All clocks of automaton increases at same speed. Clock values are compared to integers named guard values. A transition in automaton is taken based on this comparison. Further, clock values are reset for other transitions. Timed automaton is used to model and analyze the timing behavior of computer system.

Formally timed automaton is defined as a timed automaton is a tuple $T = (Q,\Sigma,c,E,q_0)$ that consists of following elements:

- Q is a finite set of states.
- Σ represents input symbols or alphabets.
- c is set of clock values.
- $E \subseteq Q \times \Sigma \times c \times Q$ is set of transitions of T.
- $q_0 \in \Sigma$ is initial state.

III. SYSTEM DESIGN

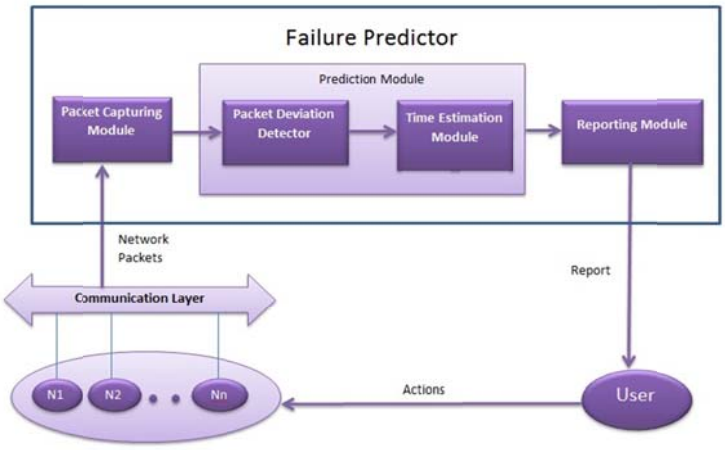

Fig 2: Failure Prediction Model

N1, N2...Nn are nodes in the network which have communication layer for communicating among them. User performs actions on the nodes in the network.

A. Packet Capturing Module:

Packet capturing module captures the packets from communication layer while the nodes are communicating. Packets are captured time-to-time while communication is going on for fastening the task. These packets are given to prediction module for further analysis.

B. Packet deviation detector:

This is the first sub module in prediction module. This module finds out deviated packets in given communication packet stream. For finding out deviated packets, we are comparing each and every packet format with standard packet format. If any deviation is found out, then number of packets deviated, current communication time and current number of packets in communication are given to next module for estimation of time to failure. If no deviated packet found, compared packets are discarded.

C. Time estimation module:

This is second sub module of prediction module. A simple calculation has been done to estimate the time to failure. By using the three variables sent by the packet deviation detector. . Current communication time is attained by using timed automata. Calculations are as follows:

Need to find out the average time gap between the deviations that are observed. The time is measure in seconds. Current communication Time or elapse time (ET) divided by number of Deviated Packets (DP) to get Average Time (AT) between deviations.

$$AT = \frac{ET}{DP} \qquad \text{(Eq.1)}$$

Possible number of Maximum Deviations (MD) can occur before failure is estimated by multiplication of number of the Current Packet (CP) and threshold value (30% of current number of total packets) and dividing obtained value by 100. Here the threshold value is considered as 30.

$$MD = \frac{30*CP}{100} \qquad \text{(Eq.2)}$$

Finally, Time to failure (TTF) is estimated by multiplying Average Time between deviations (AT) which is obtained from Eq.1 and possible number of Maximum Deviations (MD).

$$TTF=AT*MD \text{ sec} \qquad \text{(Eq.3)}$$

D. Reporting module:

This module reports the failure prediction with all the available details to the user in presentable way. The users of the network are informed by broadcasting these details to every node in the network. The prediction report includes details like number of deviated packets found, time to failure.

E. Algorithm:

```
Procedure Prediction()
{
Initialize  start_time=0 ,pkt _num=0, current_time=0, current_pkt=0, deviated_pkt=0;
Capture();
}
   Procedure Capture()
    {
If (Any two nodes starts communication) then
        Start capturing packets;
Set start _time to communication start time;
Set current _time to clock time;
Assign current_time and pkt_num to each captured packet;
Increment pkt_num;
Deviation_detect(packet content);
     }
Procedure Deviation_detect(packet content)
{
current_pkt = Packet content;
 If (current_pkt == standard_pkt) then
   Discard  current_pkt;
Else If (deviated_pkt > (30*pkt_num/100) )
{
Increment deviated_pkt;
time_estimation(current_time, pkt_num, deviated_pkt);
 }
 Procedure  time_estimation(current_time, pkt_num, deviated_pkt)
{
Calculate  Avg_time = current_time/ deviated_pkt;
Calculate  Max_deviation = (30*pkt_num)/ 100;
Calculate TTF =  Avg_time *  Max_deviation;
```

report(TTF, deviated_pkt);

}

Procedure *report(TTF, deviated_pkt)*

{

Broadcast *("found" deviated_pkt "number of deviated packets among" pkt_num. "TTF is" TTF);*

}

IV. RESULTS

The proposed method produces following results. A time slot of 100 with 40 total packets and 3 deviated packets are considered.

According to eq.1, AT = $\frac{ET}{DP}$ 100/3 = 33.33

According to eq.2, MD = $\frac{30*CP}{100}$ (30*40)/100 = 12

According to eq.3, TTF=AT*MD 33.33 * 12 = 4066.26 sec

The TTF is 4066.26 sec and the failure occurred at the time 4188.03 sec.

V. CONCLUSION AND FUTURE WORK

By the obtained results, we can conclude that the proposed failure prediction model gives user enough breathing time for TTF. The method is proposed by simple calculations and network packet information. The timed automaton is used for the clocks in network.

In future, this can be enhanced with classification of failure, location detection of failure, causes of failure and suggest user methods those can be able to prevent failure from occurring.

VI. REFFERENCES

[1] Ahmed Khoumsi, "Fault Prognosis In Real-Time Discrete Event Systems" International Workshop On Principles Of Diagnosis 2009.

[2] Rajeev Alur , David L. Dill. 1994 A Theory of Timed Automata. In Theoretical Computer Science, vol. 126, 183-235

[3] Andrew W. Williams, Soila M. Pertet And Priya Narasimhan "Tiresias: Black-Box Failure Prediction In Distributed Systems", IEEE International Parallel And Distributed Processing Symposium, 2007, Pages: 1-8.

[4] Arvind Kumar, Rama Shankar Yadav, Ranvijay, Anjali Jain "Fault Tolerance In Real Time Distributed System", International Journal On Computer Science And Engineering, 2011, Pages: 933-939.

[5] Felix Salfner, Maren Lenk, And Miroslaw Malek "A Survey Of Online Failure Prediction Methods", ACM Computing Surveys, 2010, Pages: 168.

[6] Libor Waszniowski, Jan Krakora , Zdenek Hanzalek "Case Study On Distributed And Fault Tolerant System Modeling Based On Timed Automata", The Journal Of Systems And Software ,2009, Pages: 1678-1694.

[7] Lucien Ouedraogo, Ahmed Khoumsi, Mustapha Nourelfath "Setexp: A Method Of Transformation Of Timed Automata" Into Finite State Automata Real-Time Systems, 2010 Pages: 189-250.

[8] Marco Canini, Dejan Novakovi C, Vojin Jovanovi C, And Dejan Kosti C "Fault Prediction In Distributed Systems Gone Wild", ACM4th International Workshop On Large Scale Distributed Systems And Middleware, 2010, Pages: 7-11.

[9] Marius Bozga, Oded Maler, And Stavros Tripakis, Verimag "Efficient Verification Of Timed Automata Using Dense And Discrete Time Semantics", Correct Hardware Design And Veri_cation Methods, 1999, Pages: 125-141.

[10] Michal Knotek A, Zineb Simeu-Abazi, Frantisek Zezulkaa "Fault Diagnosis Based On Timed Automata: Diagnoser Veri_cation", IMACS Multiconference On Computational Engineering In Systems Applications(CESA), 2006, Pages: 4-6.

[11] Naveed Arshad, Dennis Heimbigner, Alexander L. Wolf "A Planning Based Approach To Failure Recovery In Distributed Systems", First ACM Workshop On Self- Managed Systems, 2004,Pages: 8-12.

[12] Nawanol Theera-Ampornpunt, Bowen Zhou, Saurabh Bagchi "Predicting Time To Failure For Large Scale Distributed Systems", 42nd Annual IEEE/IFIP International Conference On Dependable Systems And Networks (DSN 2011), 2011, Pages: 27-30.

[13] Patricia Bouyer, Fabrice Chevalier, And Deepak DSouza "Fault Diagnosis Using Timed Automata" FOSSACS 2005 Springer, LNCS 3441, 2005, Pages 219-233.

[14] Roberto Baldoni, Giorgia Lodi, Luca Montanari, Guido Mariotta And Marco Rizzuto "Online Black-Box Failure Prediction For Mission Critical Distributed System", Springer 31st International Conference, 2012, Pages: 185-197.

[15] Stavros Tripakis "Fault Diagnosis For Timed Automata", Springer 8th International Conference FTRTFT 2002, 2002, Pp. 205221.

[16] Tawhid Bin Waez, Juergen Dingel, Karen Rudie, "Timed Automata For The Development Of Real-Time Systems", 2011, Pages: 1-63.

[17] Thomas Robert, Jean-Charles Fabre, Matthieu Roy "Application Of Early Error Detection For Handling Degraded Modes Of Operation", 12th European Workshop On Dependable Computing, EWDC 2009 .

[18] Tolety Siva Perraju "Specifying Fault Tolerance In Mission Critical Intelligent Systems Knowledge-Based Systems", Verizon Communications, 2001 Pages: 385-396.

[19] Xiaohui Gu , Spiros Papadimitriou, Philip S. Yu, Shu-Ping Chang "Toward Predictive Failure Management For Distributed Stream Processing Systems" The 28th International Conference On Distributed Computing Systems, 2008, Pages 825-832.

[20] Xiaohui Gux, Spiros Papadimitriouz, Philip S. Yu, Shu-Ping Changz "Online Failure Forecast For Fault- Tolerant Data Stream Processing", IEEE 24th International Conference On Data Engineering, 2008, Pages:1388-1390.