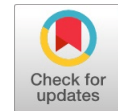


Full Stack Software Development and Multi Aspect Testing



Gayathry G, Thirumalaiselvi R

Abstract: Various statistical methods are used to estimate software reliability, to gain accuracy in software reliability prediction the data sets from the software testing results must be in detail and enormous in size. Existing papers have been acquired the reliability with a limited number of datasets, that too acquired from existing templates. This work attempts to create two layers of software design and testing methods to acquire a real-time error dataset. The first one is single system software design and second one is multiuser server-based testing. Using these two types of software design and by employing automatic data testing using Brute force algorithm, dynamic error occurrences of the datasets with time period will be acquired and tabulated, based on the error results. Instead of predicting the accuracy alone, the proposed work will also predict the possible reasons for the error occurrence scenario and tries to provide an optimal solution, using these types of vigorous testing and solutions, various aspects of the software can be measured like user traffic handling capability, possibility of input error combinations etc. For designing purpose, two different type of software one is java for desktop application and another one is python language for multiuser activity will be used to give different flavors in testing results.

Keywords: client, pattern generator, reliability, server

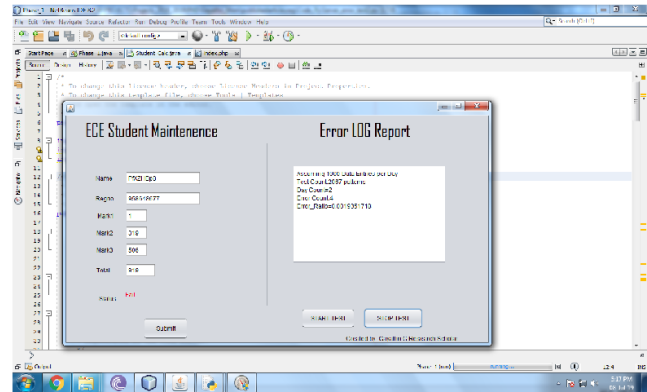


Fig 1: Standalone System: Random Test Pattern

In a single day maximum of 500 student records may be created by manual data entry and faults can be observed and tabulated, using this assumption as a basic threshold, after every 1000 data entries in from the test pattern will be assumed as a Day. Whenever the Software detects an exception an error count will be incremented. Exception may occur due to Mismatch in expected data type or a database response error. An error log report provides the ability of software to record errors. It typically includes details about the problems encountered. Using this technique, we can relate to how a software error may occur and implementing automated testing reduces 90% of manual testing time (i.e. weeks converted into minutes). Error counting based on the number of test patterns will be tabulated.

I. INTRODUCTION

In this computer era it is necessary to develop a highly reliable software to support our day-to-day life. Measuring the desirable properties of a software in quantitative terms is an established part of the reliability engineering. Reliability is one of the most desirable property of most of the software. It is defined as the analysis and study of operational behavior of software-based system based on user requirements.

II. PROPOSED WORK

A. Single System Software Design

A GUI for student database application to record the student exam result is developed and auto tested by employing a Brute force non sequential pattern generator.

B. Server Based Testing

The second type of software testing is server based testing, The reliability of a software service which is designed to handle multiple clients cannot be calculated with respect to the software errors, because the occurrences of the software error itself will be dynamic based on the number of parallel users, to test this scenario a python based grid service is developed with TCP socket protocol, the aim of the grid service is to validate whether a number supplied by the user is prime or not. The selection of prime number service is intentionally crafted due to its mathematical complexity, The time complexity of the Prime number validation is $O(\sqrt{n})$ and it can be visually verified for 8digit input too, whereas the word count program where the time complexity is $o(n)$ cannot be manually verified for large data size. That's why prime number validation is utilized as client service, server will wait for the client connection in an IP and with a specified port value.

Manuscript published on 30 August 2019.

*Correspondence Author(s)

Gayathry G*, Computer Science, Bharathiar University, Coimbatore, India.
Thirumalaiselvi R, Computer Science, Govt. Arts College (Men), Nandanam, Chennai, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.



Whenever a client initiates a connection with server through the IP address, server will immediately create a new thread to operate the client, these parallel processing threads is for servicing the client without actually disturbing the server functionality. This process is applicable for all the newly connected client, a client can either manually supply an integer to the server or it can make the server thread to go into test mode in which a randomly generated variable size integer will be applied to the server at 1 sec time delay.

Through this vigorous method of utilizing server with multiple clients, we can measure the CPU utilization, RAM availability and the Disk Hardware errors of the server with respect to the number of clients. Through this testing instead of predicting the software errors occurred during testing we can predict the load handling capability of the server by means of maximum number of clients it can handle at a time and min and max response time to client based on the queue etc.

Following screens display server that is waiting for client query, client passing data to server and client testing pattern to server.

Fig 2 : Server Waiting for Client Queries

Fig 3 : Client Passing Data to Server

Fig 4 : Client Testing Pattern to Server

The results are tabulated below. The table display the CPU, RAM usage and average response time from server when it handles multiple clients.

Table 1: CPU Usage based on multiple clients

| S NO | No of Clients | CPU USAGE (i3 Core) | RAM USAGE(MB) | Avg Response Time(sec) |
|------|---------------|---------------------|---------------|------------------------|
| 1 | 1 | 25% | 100 | 5 |
| 2 | 2 | 26% | 104 | 5 |
| 3 | 3 | 28% | 105 | 6 |
| 4 | 4 | 31% | 109 | 8 |
| 5 | 5 | 34% | 109 | 12 |

The CPU usage expressed in the form of graph. The graph shows that as the number of clients increases there is an increase in CPU usage.

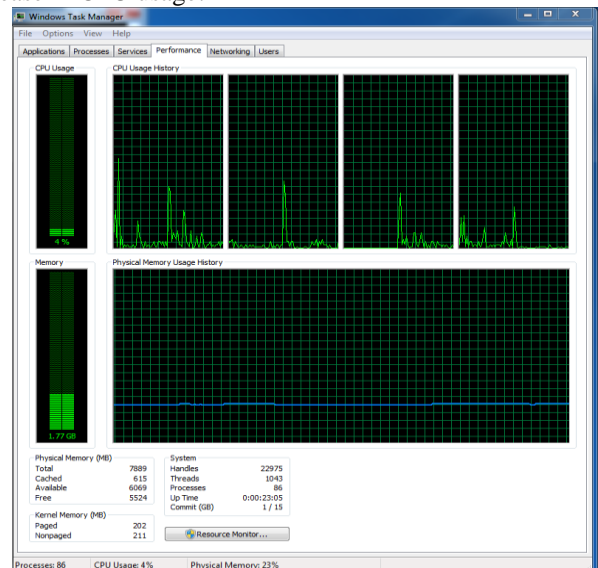


Fig 4: Client 1

The above graph display the cpu usage of single client.

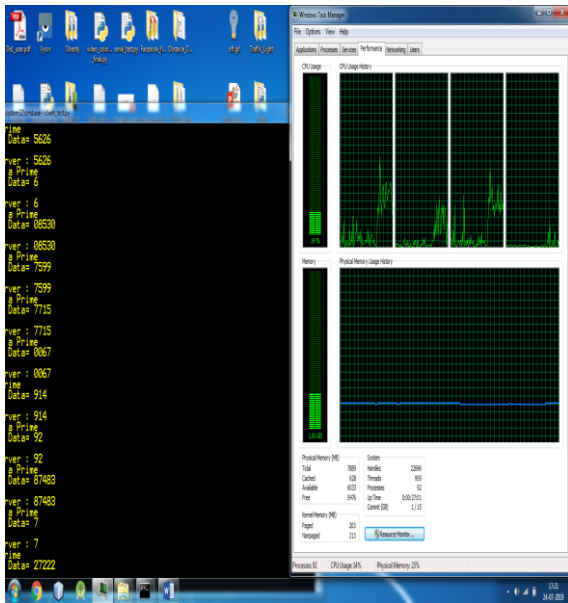


Fig 5 : Client 3

The above graph display the cpu usage of three different clients.

III. CONCLUSION AND FUTURE WORK

This proposed work attempts to redefine the deciding metrics of software reliability. Still now software reliability is calculated based on the number of software or hardware faults occurred over a period of time. Current methods to measure the reliability of software are usually focused on manual data collection method. For such products, unique issues arise in obtaining the failure data, and in analyzing this data to determine reliability. Whereas in the era of full stack development (Software, Webservers, APP) predicting reliability is a multi-faced task. The factors which impact the reliability of software can no longer be identified using traditional data collection methods. This proposed standalone application will read the software description, and will generate test patterns according to the input types and collects test results, predict the software reliability in real time and suggest the possible ways to improve the software. This quantitative information pro-actively focus on analyzing the reliability of software. In server-based testing we can measure the CPU utilization, RAM availability and the Disk Hardware errors of the server with respect to the number of clients.

Thus, the general formula for predicting the software reliability based on number of software error need to be modified to include various dynamic variable like, no of CPU cores, minimum RAM requirement and reliability score of the software may varies based of the different system resources.

From the above-mentioned issues, it is necessary to develop an automated tool using vb.net. In future, the results and data collected from software at the time of real time testing would be tabulated in excel and it will be passed to vb.net software to calculate reliability of software.

REFERENCES

1. IEEE. IEEE Standard Glossary of Software Engineering Terminology. Number STD 610.12-1990. IEEE, 1990.

2. Xue, M. and Zhu, C., "The Socket Programming and Software Design for Communication Based on Client/Server", 2009 Pacific-Asia Conference on Circuits, Communications and Systems, pp. 774-777, 2009
3. IDG Communications (2017). Sockets Programming in Java World from IDG. [Online]. Available at: <https://www.javaworld.com/article/2077322/core-java/core-java-sockets-programming-in-java-atutorial.html> [Accessed 17-July-2017]
4. Z. Yuqing, W. Wu, D. Li, "Efficient Client Assignment for ClientServer Systems", IEEE Transactions on Network and Service Management, Vol. 13 Issue 4, pp. 835-847, August 2016.
5. Brian K. Williams and Stacey C. Sawyer. Using Information Technology: A Practical Introduction To Computers & Communications. McGraw-Hill Irwin, Montreal, 2003
6. <http://www.mysql.com>
7. <http://www.javaworld.com>
8. APTECH Team (2005): Java Simplified – 1. Aptech Limited, Mumbai, India.

AUTHORS PROFILE



Ms.G.Gayathri is working as Assistant professor in the Department of Computer Science, Mar Gregorios Arts and Science College, Chennai. She is pursuing her Ph.D in the area of software engineering from Bharathiar University, India. She has 14 years of teaching experience



Dr. (Mrs.) R.Thirumalaiselvi is currently the Research Supervisor and Assistant Professor in the Department of Computer Science, Govt. Arts College (Men) (Autonomous), Chennai. She has over 20+ years of experience in various arts and science colleges and as a research supervisor. She is guiding many PhD students registered under various universities.