
ABSTRACT

Many number of software dependent web services are being developed & everything is moving from offline to online world. To maintain these web services & ensure its smooth operations we need some system mainly servers. It is necessary to check the servers systems performance to maintain its smooth functioning. The performance can be checked on number of parameters like their Response time, Memory utilization, CPU utilization etc under different conditions.

KEYWORDS: Features testing, Database server, Integrated Platform, Monitoring & Load testing, tools, Virtual servers

INTRODUCTION

Everyone is moving to the online world at a rapid space & to support these online functionaries we need some infrastructure & Server is one of the main component of this infrastructure. So to maintain a smooth working of the server one needs to look after its maintainance so this is where our tool is useful. The tool concentrates on testing the server based upon different scenarios in different conditions depending upon the different inputs. The tool will check the server on features like Response time, Memory management & CPU utilization. This paper concentrates on the implementation part of the tool. The paper will be a continuation of the paper "Advanced Web Server Testing Tool" published in "International Journal Of Science & Research"

This paper will mostly concentrate on the implementation of the features memory utilization & CPU utilization. When we say Memory utilization we talk about the total memory used by the server its thread, its process, Concurrent process. Same also goes with the CPU utilization the total use of cores by threads, process etc.

Memory of a machine will constitute of total memory. The total memory will be inclusive of Used memory, Free memory, Shared memory, Buffer memory & Cache memory Whereas the total CPU utilization can be divided into number of cores & number of blocks present in each core

MATERIALS AND METHODS**System Introduction**

This tool is used to check the overall health condition of the server. The general setup or architecture of our tool is accordingly as follows:-

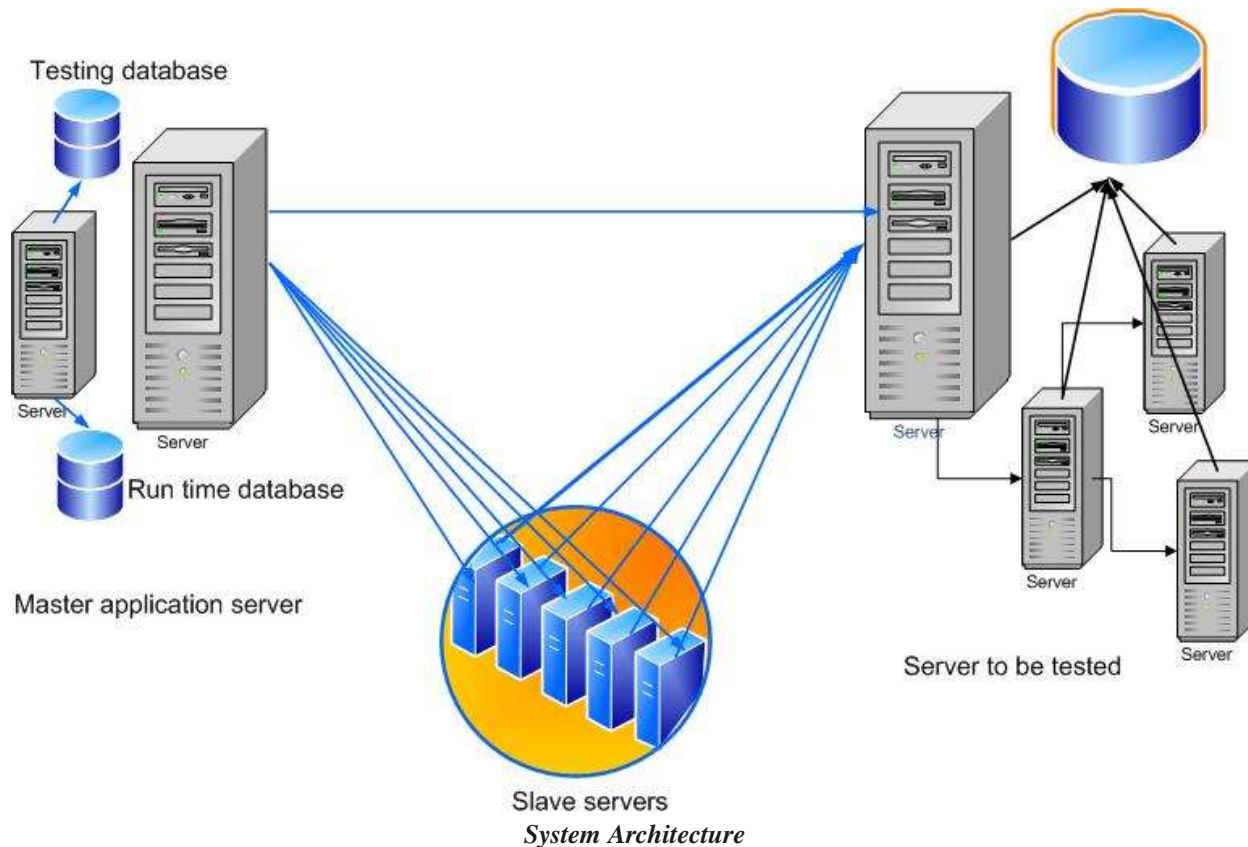
Main server-This is the main system component of our architecture it acts as a main administrator. The user sends http request for reads & writes information to the main server. The main server is responsible for pre-checking. main server collects the user required information and acts accordingly.

Slave servers- Virtual copies are created of the existing server. These copies act as Slave server & these slave server acts as mediator. These servers handle number of requests sent by user. It handles the information sent by main server. slave servers provides formulas and logic for providing result.

Database-database consists of various MB, KB files.

Server to be tested-it is the actual server which needs to be tested. the graphical result of all the features are provided to this server.

Figure:



Implementation

1. Memory Utilization:

When we say memory utilisation it refers to the total memory being used by the system process & the subsidiary process. The total memory constitutes of the Used memory, Free memory, Shared memory, Buffer memory, Cache memory. So this is what we propose for memory utilization of our tool

Main server -Main server will stimulate to the server to be checked

Server to be checked

Check the-Total memory, Used memory, Free memory, Shared memory, Buffer memory, Cache memory

Total memory utilization=total memory-(used+free+shared+buffer+cache)

Main server receives the total memory utilization

1. CPU Utilization:

When we talk about CPU utilisation it is the overall use of the resources being provided by the system. These resources are used by the system process & its subsidiary process

Main server

n=no. of request to be sent

n.s=no of slave servers

$n/n.s = x$ (no. of requests to be sent by slave servers to server to be tested)

we will initialize the slave servers to send the x no. of request to server(test)

Slave server

Send x no. of request to the server to be tested

Server at testing
Calculate cpu usage(u)
 $U=100\%-(\% \text{ of time spent to idle task})$
%time in idle task= avg period of background task with no load/avg period of background task with some load *100
Send to main server

RESULTS AND DISCUSSION

This paper presents a general outlook for testing of servers
Characteristics of the proposed approach are summarized as follows:

1. Integrated testing of a server with minimal resource usage
2. Specific parameter testing depending upon the users requirement

In the future, we can implement testing as well as monitoring of servers

CONCLUSION

The evergrowing popularity of the Internet & its outreach to the people has given rise to a new age a Digital Age. As a human needs to take care of his health for better lifestyle likewise we also need to take care of the health condition of the infrastructure to ensure a smooth working of the Digital Age.

ACKNOWLEDGEMENTS

There are many people to thank who have contributed in the development of this paper. We owe our deep regards to express our gratitude to all the faculty members of RMD Sinhgad School of Engineering for supporting, guidance, help, and inspiration all through this paper.

REFERENCES

1. Nadia Alshahwan ,Mark Harman ,Alessandro Marchetto and Paolo Tonella “Improving Web Application Testing Using Testability Measures” 2009 IEEE
2. Hasliza Sofian, Raihana Md Saidi, Rozita Yunus, Siti Arpah Ahmad “Analyzing server response time using testing power web stress tool” 2010 International Conference on Science and Social Research (CSSR 2010), December 5 - 7, 2010, Kuala- Lumpur, Malaysia
3. Srinivas Shenoy, Nur Asyikin Abu Bakar, Rajashekara Swamy “An adaptivr framework for web services testing automation using Jmeter” 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications