International Conference on New Scientific Creations in Engineering and Technology- ICNSCET21

An Analysis study of website Testing in various Environment using JMeter Tool

Arul Jothi S[1]

1AssistantProfessor, Department of Computer Science and Engineering Nadar Saraswathi College of Engineering and Technology,Theni,Tamilnadu

Vedhanayaki[2], Valli[3], Naganandhini[4]

UG Students

[2,3,4] Department of Computer Science and Engineering Nadar Saraswathi College of Engineering and Technology,Theni,Tamilnadu

Abstract— For Advertisement and marketing,website plays a major role in day to day life and also in IT sectors too. Web development is essential to the success of online marketing and branding campaigns. But it requires to invest limited resources, and its success (or failure) can have lasting impacts on law firm's bottom line. The importance of testing in web development is overlooked by many of today's law firms. Testing can help you avoid costly mistakes and get the most out of web development efforts. For efficient building of websites, we need an efficient language and environment. In this project, we made various tests to find which language is efficient to build a website. Here,we build websites in two different languages namely python with visual studio Environment and Node.js. Using JMeter tool can able to find the performance of two build website has been analysed with the parameters like throughput, latency, navigation speed, uploading and downloading time.

Keywords—website analysis, performance tesing, load testing,

1.INTRODUCTION

The Web has evolved into a global environment for delivering all kinds of applications, ranging from small-scale and short-lived services to large-scale, enterprise workflow systems distributed over many servers. Applications that use HTML-based front ends benefit from the Introducing a Web application in an organization causes a paradigm shift, because it has the potential to significantly change an organization's work practices and procedures. As a result, we need corporate commitment to manage the shift and drive the development.

Web analytics is the collection, reporting, and analysis of website data. The focus is on identifying measures based on your organizational and user goals and using the website data to determine the success or failure of those goals and to drive strategy and improve the user's experience. Frequent website analysis is an essential part of ensuring your site is efficient, competitive, and useful for your users, and specific analysis checklists can help you stay on task.

Website analysis is the practice of testing and analyzing a website's performance in relation to SEO, speed, competition, and traffic.Any site can benefit from some form of website analysis if the results are then used to improve it—for example, by reducing page size to increase overall speed or optimizing a landing page with lots of traffic for more conversions.

2. LITERATURE SURVEY

2.1.Design and Implementation of Web Based Real Time Chat Interfacing Server

Chat application is a feature or a program on the Internet to communicate directly among Internet users who are online or who were equally using the internet. Chat applications allow users to communicate even though from a great distance[1]. Therefore, this chat application must be real-time and multi platform to be used by many users. The development of information and communication technologies are rapidly making one of the reasons for Indonesia, especially Bandung to develop this chat application. That's because Indonesia does not always rely on outsiders. It is important for Indonesia to develop this chat application for themselves. This chat application in the manufacture begins with the collection of relevant data that will be displayed in the web and mobile versions. The programming language used to build server is Node.js with express framework and MongoDB database.

2.2.Android Chat Application Development using AWS

Online chatting is a kind of communication that is done over the Internet, which uses an online chat application to it. The online chatting offers the users to experience real-time conversation with the people who are located geographically in a different location that is it offers real-time transmission of messages from one person to other[2]. Chat messages are usually short so that the other participant responds to them quickly. Thereby, it makes the users of the online chatting app, feel the chat like a real-time conversation. The online chat application allows the user to address the single message to an individual receiver or multiple receivers that is by using the point to point communication and multicast communication techniques. All these parameters must be considered while designing a chat application along with these parameters the other important parameter that should be considered is the security. There is a growing need for providing security for the mobile chat applications, with a large user number it is very important for a chap application to provide all security for the data they store and transmit for better resistance to cyber-attacks. So to provide the security all the messages that are being transmitted and the data stored in the database must be encrypted. For the storage of the user's details and their data, the AWS (Amazon Web service) is used. AWS is a cloud platform that provides several IT infrastructure like database, SQL, virtualization and many more. It allows storing data from the online application and maintains all the user's detail which helps in for the registration and login of the user to the online chat application.

3. EXISTING SYSTEM

In the existing system, Chat application built with Node.js, MongoDB and Socket.io running much faster in order to achieve real-time chat applications compared with the chat application built with PHP and MySQL[3]. Test result of the chat application shows the same results when tested on localhost, in the network or when in hosting. Here is a comparison table of chat application built with Node.js, MongoDB, and socket.io with chat application built with PHP and MySQL. Based on the results of tests performed, it can produce an conclusion that the chat application built with Node.js, Socket.io, and MongoDB is running in real time very well with a speed less than one second (6 times faster) than the chat application built with PHP and MySQL[4]. So it can be ascertained if data more than 200 chat, Node.js, Socket.io and MongoDB will have a speed 1200 faster than PHP and MySQL. CombSort Strict CPU Test.

4. PROPOSED SYSTEM

The main objective of the proposed system is to analysis the performance of the website by its building architecture. In our project, we consider two kinds of websites like Node.js and Python. To find the performance of the running website we using an efficient tool JMeter. Using JMeter we can able to find the performance of our website. In our analysis, we consider some parameters like throughput, latency, navigation speed, uploading and downloading time. We analysis these parameters using testing tool and in the final we made a report on which language is efficient and good to build a website. In this performance testing, we are getting the result “Python” is best when compare to Node.js.

5. SYSTEM ANALYSIS

The creative and challenging phase of the system life cycle is system analysis. The term analysis describes the initial stage of the system. The term design describes the final system generated and process by which it is developed. It refers to the technical specification that will be applied in implementing the system. It also includes the construction of programs and program testing. System analysis and design is a solution a "how to" approach to the creation of a new system. This important phase is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study.

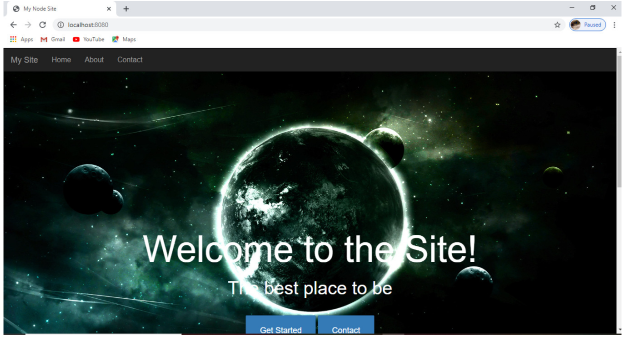
6.MODULES

1. Creation of website using Python and Node.js.
2. Comparing two websites using performance testing tool "JMETER".
3. Analysis the Login time.
4. Analysis the navigation speed.
5. Analysis the throughput.
6. Critical section controller.
7. Calculating uploading and downloading time.

MODULE DESCRIPTION

6.1 CREATION OF WEBSITE

We created two website in different environment and also in different language. Website development by python language and in visual studio environment. Another website was created in node.js.



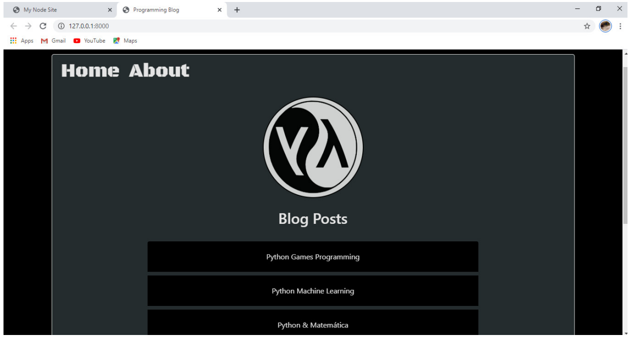
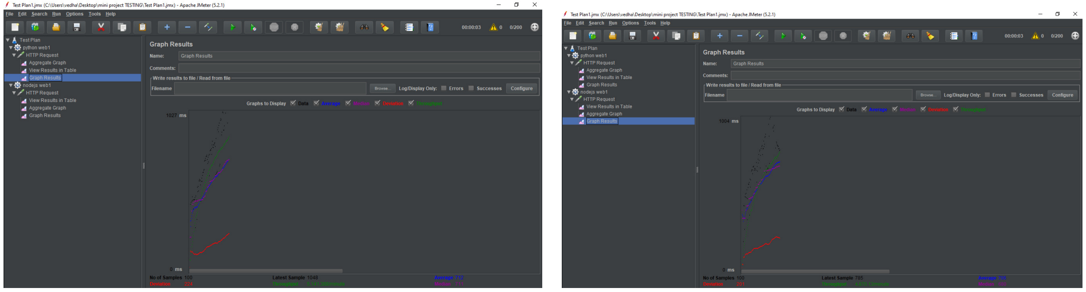


Fig. 1.WEBSITE BUILD BY PYTHON

Fig. 2 WEBSITE BUILD BY NODEJS

6.2 PERFORMANCE TESTING

Performance testing is the process of determining the speed, responsiveness and stability of a computer, network, software program or device under a workload. Performance testing can involve quantitative tests done in a lab, or occur in the production environment in limited scenarios.


Test Plan1.jmx (C:\Users\vedha\Desktop\mini project TESTING\Test Plan1.jmx) - Apache JMeter (5.2.1)

Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 100 Latest Sample 1048 Average 712

Deviation 224 Throughput 5,141.388/minute Median 711

Test Plan1.jmx (C:\Users\vedha\Desktop\mini project TESTING\Test Plan1.jmx) - Apache JMeter (5.2.1)

Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 100 Latest Sample 785 Average 706

Deviation 201 Throughput 4,870.13/minute Median 690

Fig.3 FINDING THROUGHPUT

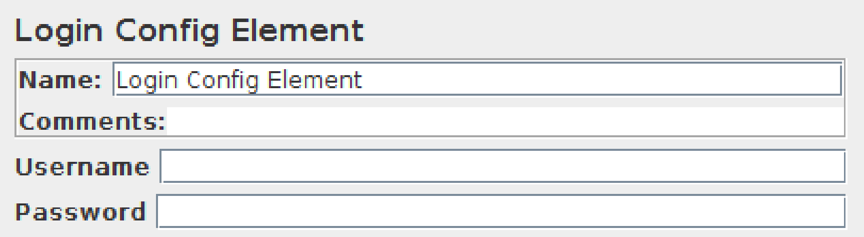
6.2.1 JMETER

JMeter Metrics are widely used in the following section, thus it's better if you're comfortable with their definition:

- Elapsed time: Measures the elapsed time from just before sending the request to just after the last chunk of the response has been received,
- Latency: Measures the latency from just before sending the request to just after the first chunk of the response has been received,
- Connect Time: Measures the time it took to establish the connection, including SSL handshake,
- Median: Number which divides the samples into two equal halves,
- 90% Line (90th Percentile): The elapsed time below which 90% of the samples fall,
- Standard Deviation: Measure of the variability of a data set. This is a standard statistical measure,
- Thread Name: Derived from the Thread Group name and the thread within the group. The name has the format groupName + “ ” + groupIndex + “-” + threadIndex where:
    - groupName: name of the Thread Group element,
    - groupIndex: number of the Thread Group in the Test Plan, starting from 1,
    - threadIndex: number of the thread within the Thread Group, starting from 1.
- Throughput: Calculated as requests/unit of time. The time is calculated from the start of the first sample to the end of the last sample. The formula is: Throughput = (number of requests) / (total time).

6.3 ANALYSIS THE LOGIN TIME

The Login Config Element lets you add or override username and password settings in samplers that use username and password as part of their setup.



Login Config Element

Name: Login Config Element

Comments:

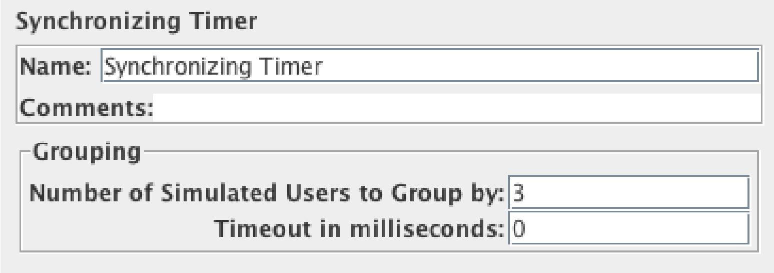
Username

Password

Fig.4 analysis the login time

6.4 ANALYSIS NAVIGATION SPEED

The purpose of the SyncTimer is to block threads until X number of threads have been blocked, and then they are all released at once. A SyncTimer can thus create large instant loads at various points of the test plan.



Synchronizing Timer

Name: Synchronizing Timer

Comments:

Grouping

Number of Simulated Users to Group by: 3

Timeout in milliseconds: 0

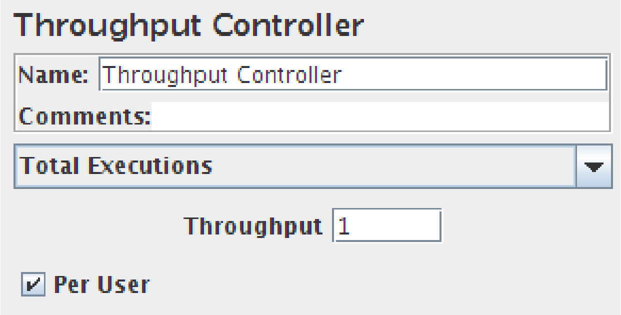
Fig.5 analysis the navigation speed

6.5 ANALYSIS THROUGHPUT

The Throughput Controller allows the user to control how often it is executed. There are two modes:

- percent execution
- total executions

  Percent executions causes the controller to execute a certain percentage of the iterations through the test plan. Total executions causes the controller to stop executing after a certain number of executions have occurred. Like the Once Only Controller, this setting is reset when a parent Loop Controller restarts.


Throughput Controller

Name: Throughput Controller

Comments:

Total Executions

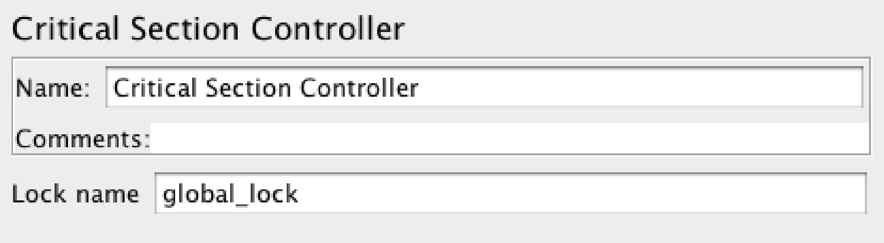
Throughput 1

☑ Per User

Fig.6.Measuring Throughput

6.6. CRITICAL SECTION CONTROLLER

The Critical Section Controller ensures that its children elements (samplers/controllers, etc.) will be executed by only one thread as a named lock will be taken before executing children of controller.


Critical Section Controller

Name: Critical Section Controller

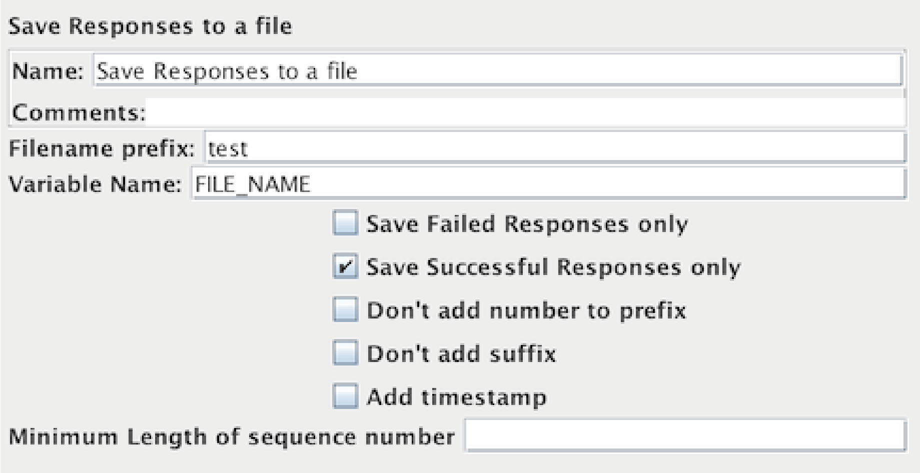
Comments:

Lock name global_lock

Fig.7.Controller

6.7.MEASURING UPLOADING AND DOWNLOADING TIME

This test element can be placed anywhere in the test plan. For each sample in its scope, it will create a file of the response Data. The primary use for this is in creating functional tests, but it can also be useful where the response is too large to be displayed in the View Results Tree Listener. The file name is created from the specified prefix, plus a number (unless this is disabled, see below). The file extension is created from the document type, if known. If not known, the file extension is set to 'unknown'. If numbering is disabled, and adding a suffix is disabled, then the file prefix is taken as the entire file name. This allows a fixed file name to be generated if required. The generated file name is stored in the sample response, and can be saved in the test log output file if required. The current sample is saved first, followed by any sub-samples (child samples). If a variable name is provided, then the names of the files are saved in the order that the sub-samples appear. See below.


Save Responses to a file

Name: Save Responses to a file

Comments:

Filename prefix: test

Variable Name: FILE_NAME

□ Save Failed Responses only

☑ Save Successful Responses only

□ Don't add number to prefix

□ Don't add suffix

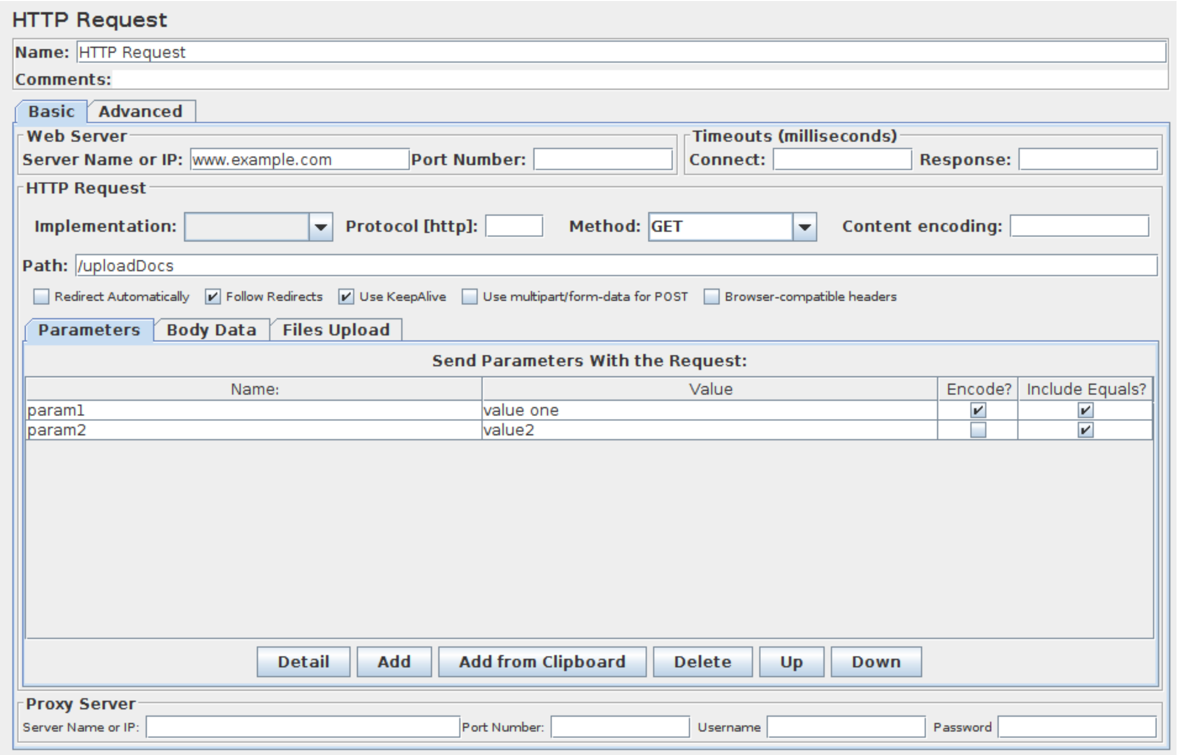
□ Add timestamp

Minimum Length of sequence number

Fig.8.Measuring upload and download time

6.8 LOAD TESTING

A load test is type of software testing which is conducted to understand the behavior of the application under a specific expected load. Load testing is performed to determine a system's behavior under both normal and at peak conditions.



Name: HTTP Request

Comments:

Server Name or IP: www.example.com

Method: GET

Path: /uploadDocs

□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data for POST □ Browser-compatible headers

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
param1	value one	☑	☑
param2	value2	□	☑

Fig.8.Load Testing

7.CONCLUSION

In this paper, we make a report on performance testing between to websites each one is made by different language and also with different environment. One website is build by python with visual studio environment. Another website is build by Node.js. Now we are going to analysis the performance of these two website created in different environments. The performance of the running time of different website parameter has been evaluated using an efficient tool JMeter. Using this .it is possible to evaluate under the merics like throughput, latency, navigation speed, uploading and downloading time.

8.EXPERIMENTAL RESULTS

Below table shows the comparisons of two languages in website creation.The performance has been evaluated under the metrics.

S.NO	LANGUAGE	THROUGHPUT	DEVIATION	NAVIGATION SPEED
1.	Python	5141.388/min	224	1061
2.	NodeJS	4870.13/min	201	814

REFERENCES

[1] Croucher, T. H., & Wilson, M. (2012). Node: Up and Running. United States.

[2] Haryadi, M. F. (2010). ANALISA DAN PERANCANGAN APLIKASI CHATTING BERBASIS WEB MENGGUNAKAN FLASH CS3. 1-2.

[3] Kurniarin, K. (2012). PENERAPAN MONGODB DAN CODEIGNITER PADA PERMAINAN KOKOLOGY. 3-4.

[4] Purnomosidi, B. (2013). Pengembangan Aplikasi Cloud Computing Menggunakan Node.js.

[5] Rahmadini, A. (2013). Pembangunan Sistem Informasi Pengelolaan Inventaris Barang Divisi Pustekin Berbasis Web. Bandung: Politeknik Telkom.

[6] Sidik, B. (2011). JavaScipt. Bandung: Informatika.

[7] Kiessling, Manuel. 2012. The Node Beginner Book. lulu.com, United Stated.

[8] Mardan, Azat. 2012. Practical Node.js: Building Real-World Scalable Web Apps. Appress.

[9] Teixeira, Pedro. 2012. Professional Node.js: Building Javascript Based Scalable Software Kindle Edition. Wrox.

[10] Teixeira, Pedro. 2012. Hands-on Node.js. Wrox.

[11]Android studio developer link, https://developer.android.com/studio/features

[12] Amazon web service providers, https://aws.amazon.com/about-aws/

[13] L.T. Duffy, DoD collaboration and chat systems: Current status and way ahead, in: Proceedings of the International Conference on Semantic Computing, IEEE Computer Society, 2008, pp. 573–576.