VOL. 13, NO. 11, JUNE 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

MULTI OBJECTIVE PARTICLE SWARM OPTIMIZATION FOR PERFORMANCE TESTING IN WEB APPLICATION

B. Shyaamini[1] and M. Senthilkumar[2]
[1]Department of Computer Science, Bharathiar University, PSG College of Arts and Science, Coimbatore, Tamil Nadu, India
[2]PG and Research Department of Computer Science, Government Arts College, Udumalpet, Tamil Nadu, India
E-Mail: shyaaminibalu@gmail.com

ABSTRACT

Web application performance testing is the emerging and most important field of software engineering. The performances of the web applications depends upon several different type of the testing process like load testing, soak testing, smoke testing and stress testing. The load testing is used in this paper to determine how the web application behaves under varying load. In this paper the Multi Objective Particle Swarm Optimization (MOPSO) proposed to optimize the server behaviour for improving the performance in the web application. The MOPSO select the more number of server behaviour such as work load, CPU, bandwidth, throughput, response time, hits per second, database locks, thread count, number of position in the service queue, round trip time, server mean service time. The optimized server parameter tested with JMeter performance tools which return the better services to the user. The result shows that MOPSO increase the performance of the web application in terms of less workload, maximum CPU utilization, less bandwidth and less response time.

Keywords: performance testing, MOPSOPT, server behaviour, work load, CPU utilization, bandwidth, response time.

1. INTRODUCTION

Now a day many companies, enterprises and web sites set up are based on the Web based applications [1], the web scale can be expanded and more peoples pay their awareness to how to improve the accuracy and reliability of the web applications. The performance testing is describes as the process of collecting the information and analyses the collected data which is used to predict the load level will be run out of the resources of the system. The aim of the performance testing in the web applications are simulating the true load to identify the performance bottlenecks, optimizes the performance of the system, and then assure the real time application can provide the satisfied and reliable performance.

The performance testing [2] is the general testing and performs to find how the system performs in terms of the responsiveness and stability based on the specified workload. This technique also investigates the measure, validity or verifies the attributes of the system quality like scalability, reliability and resource usage. The load testing is the one of the approach of the performance testing. The load testing approaches [3] also used to analyses the performances of the web based system. The load testing is used to identify the isolate system bottlenecks, time application components, then predicting the system scalability.

In the early work the genetic algorithm [3] used to analyse the web application which is used to applicable for some applications like e-governance, e-commerce, e-learning, blogs, forums, social networking and group communications. Since the time complexity of this technique quite high in the web applications

In this paper the time is consumed by using the Multi Objective Particle swarm optimization (MOPSO).The MOPSO used to optimize the server behaviour to improve the performances of the web applications.

2. LITERATURE SURVEY

Web Application Testing (Zhongsheng, Q. 2010) proposed [4] the novel approach for web application testing depends upon the user by using the genetic algorithm. The proposed technique used to generating and optimizing the test cases. In this technique the URL trace based reduction algorithm used to the user sessions acquire are lessened efficiently. Then the prioritizing user sessions are presented which is used to improve the efficiency of the test cases. The strategy of the test reuse and concurrence is provided which is used to decrease the testing time and cost. Since the proposed technique only consider the coverage ratio as the performance metrics does not consider the actual running time, loading time, time to save test.

Automated Web application Testing (Alshahwan, N., & Harman, M. 2011) presented [5] the analysis of the search based testing algorithms and introducing the tool SWAT. The SWAT used for the automated web application testing to significantly increase the efficiency of the classical search based techniques for make use of both the static and dynamic analysis. Then the dynamically Mined value seeding (DMV) introduce into the searching process which is used to iteratively increase the overall coverage and reduces the effort. The proposed approach tested with the 6 real world web applications. The performance is evaluated in terms of the branch coverage of the server side code, evaluation of the fitness, execution time and fault finding time. Since the proposed technique can detect only the 60 faults and 424 warnings overall 6 applications.

Automated Testing Web Based Applications (Nivas, T. 2011) presents [6] the general design principles for the test

script. In this analysis the testing includes the both traffic generation and setting up the traffic environment after that verifying the both software and hardware configurations. It's used to provide the accurate prior to sending traffic and creating the report at the end of the test. So the proposed technique requires the part of the complete harness to achieve the tasks. The proposed analysis assure the end to end testing process pre -test, traffic generation and post-test activities complete efficiently. However the proposed analysis needs more investigation.

Web Application Performance by using Component Load Testing (Babbar, C., *et al*. 2011) presents [7] the web application based analysis of the performance testing with different parameters and stress levels. The testing of the entire site is very complex including the behaviour. The sub behaviour includes the query, downloadable objects and meaningful business entities in the web applications. The result shows that the entire web application require the rigorous performance testing with the all performance indicators set for every sub behaviour. The proposed approach considers only limited number of performance parameter such as the request time, response time, and load time etc.

Investigation and Evaluating the Performance testing for web Application (Kalita, M., & Bezboruah, T. 2011) presents [8] analyse the performance testing based on the web application for increasing the virtual users. The proposed approach developed with the .Net application so the author (Kalita, M., & Bezboruah, T. 2011) called as PReWebD. The performance is the most important criteria for web application. The performance parameters are tested with the PreWebN with different stress levels. The analysis result shows that distribution of response time and throughput is normal compared to PreWebN implemented with the Java technique. Since the analyses use only the less investigation.

Performance Analysis of Web Application (Zhou, Q., Ye, H., &Ding, Z. 2012) presents [9] the performance analysis of web applications. The new approach introduces to build the web user behaviour model based on the user's browsing behaviour. The performances are evaluated and obtain from the markov model. The performance measures are based on the service response time, service path length, service utilization, service implementation rate and access error rate. The result shows that the proposed approach improve the design of web application and optimize the services.

Web Application Testing and Analysing (Proko, E., & Ninka, I. 2013) presents the analysis [10] of the performance testing web applications. In this analysis the two performance testing tools are introduced like the Firebug and YSlow which is used to analyse effective web application performance test with the minimal cost. The result shows that the software developer helps to predict the complexity in the performance of the system as well as helps to increase the web application with the high performance.

Load Testing based Web Applications (Wang, X., *et al*. 2013) proposed [11] Realistic usage model (RUM) and Simple Load Model (SLM) and implementing the high automation load testing tools based on the Load Testing Automation Framework (LTAF). The LTAF based on two proposed models it is used to performing the easy and realistic load testing for the web applications. However the proposed model has some limitation such as the RUM supports the only simple data parameters and the SLM not support the AJAX web applications.

Study of Systematic Mapping of the Web Application Testing (Garousi, V., *et al.* 2013) presents [12] the analysis of the systematic mapping study in the area of the web application testing. The analysis based on the genetic classification scheme for classifying the papers in to the web application testing field, the study systematic mapping in the web application field, analysis of the demographic trends and bibliometrics in the web application field, then finally the online repository of the papers are collected and analysed based on the systematic study. Since the proposed analysis need more investigation for the web application testing which includes the automated oracle generation, mutation testing, concolic testing, testing asynchronous client/server interaction, coverage metrics etc.

Automated Testing on Web Application (Zhou, Y., & Evans, D. 2014) proposed [13] the Single Sign on Scan (SSOScan) to detect automatic vulnerabilities checking for web applications. The SSOScan consider the website URL as the input and find the site uses the Facebook uses SSO, then automatically sign in the site by using the face book test accounts and complete the registration process if is require. After that the SSOScan simulating the various attacks in the site when observing the responses and monitor the traffic in the network to automatically determines the applications is the vulnerable toalready the tested vulnerabilities. Since the proposed work focus only the face book SSO, does not consider the identity providers or the other protocols.

Integrated Web Application Testing (Stepien, B., & Peyton, L. 2014) proposed [14] TTCN-3 framework provides the several advantages to the web application testing. The TTCN-3 assures the flexible and powerful platform for the web application testing. The proposed framework utilizes to significant reuse and dramatically reduces the size and complexity of the test logic. In this approach the unit test framework like HtmlUnit to perform the concrete data extraction which provides the creating the instance of the WebResponse Type Object, creating the instance of the appropriate type to each field of the record type, then populating the WebResponseType object instances with object instance creation to the each field. However the lack of memory usage reduces the performance.

Performance Testing of Web Application (Khan, R., & Amjad, M. 2016) presents [15] the analyses of the performances of web application and also analyses the complexity of hardware, software and resource utilization. The proposed work analysis focus on the performance testing based on the load test. The load is tested with the help of HP ALM (Application Life Cycle Management) tool then the throughput and hits per second observed during the load test. The HP AML tool used to schedule

and run the test for client's requirements for the multiple users. The web application is developed and tested in this tool. However the proposed technique has some limitation such as the only supported the limited number of the size.

3. METHODOLOGIES

In this paper the multi objective particle swarm optimization (MOPSO) used to reduce the time consumption of the performance testing in the web applications.

3.1 Multi Objective Particle Swarm Optimization for Performance testing in web application (MOPSOPT)

The Particle swarm optimization (PSO) algorithm is the optimization algorithm it's based on the movement of the flock of birds. The PSO has reach increasing attention in the many research fields because of the good results and finding the optimal solutions in the complex problems and its ease in configuring the parameters of algorithms. The PSO concepts involves the populations of the particles called the swarm where the each particles to explore other locations in the exploration space.

In this algorithm the particle can communicate between the each other and the trade experience to predict the best search region. The way particles are communicate and the best regions are explored by the each particle then influence the velocity and the direction where the each particle is flying. The fitness value is used to measure the quality of the solution in the investigation space.

In PSO algorithm, for each iteration the particle velocity updated as follow:

$$V_{ik}^{n+1} = V_{ik}^{n} + C_1 r_1^n (pbest_{ik}^n - X_{ik}^n) + C_2 r_2^n (gbest_k^n - X_{ik}^n) \quad (1)$$

V- Represent as the weight which is used to control the overall capacity of the algorithm, c_1- cognitive learning factor, c_2-social learning factor, r_1,r_2 are random numbers between 0 and 1. $\vec{x}_{i_{pbest}}$- best position of the particle i, $\vec{x}_{gbest}$- found best position, based on the equation (1) the new position of the each particle is updated by the following equation.

$$P_{ik}^{n+1} = X_{ik}^{n} + V_{ik}^{n+1} \quad (2)$$

The MOPSO is the adaption of the PSO algorithm to dealing with the multi objective problems. In this paper the MOPSO is used to optimize the performance parameter in the web application. In the beginning of the MOPSO, the initial swarm is randomly generated. Then the set of gbest is initialized by using the non-dominated particles from the swarm. The set of gbest is stored in an external archive. In each iteration, the gbest is chosen then the positions of the particles are updated. The set of gbest is updated after all the processes of all particles have finished. The MOPSO terminates its processes when the number of iteration reaches the objective function. The objective functions are less workload, maximum CPU utilization, less bandwidth, less response time. The performance tools used to optimize the server behaviour parameters which are returns the best services to the user. Since the time consumption in the web application performance testing is reduces by using the MOPSO technique. The web performance testing has been repeated for different server configuration. The MOPSO optimization functions are:

$$f(x) = \sum_{m=1}^{M} w_m . f_m(x) w_m \in [0,1], \sum_{M=1}^{M} w_m = 1 \quad (3)$$

The proposed Multi objective particle swarm optimization performance testing (MOPSOPT) used to optimize the server behaviours. The each server behaviour assumed as the parameter. The parameters are initialized in the initial phase. After that the position are initialized for each parameter. To achieve the objective function such as less workload, maximum CPU utilization, less bandwidth and less response time. The proposed algorithm randomly select the sever behaviour as work load, CPU, bandwidth, throughput, response time, hits per second, database locks, thread count, number of position in the service queue, round trip time, server mean service time based on the each objective function. Calculate the fitness value using Equation (3). Then find the gbest by using the Pareto optimal. The Pareto optimal mainly used to handle the non-dominated solutions. Obtain the each parameter fitness value compared with the old pbest fitness value if the new one is better means it will assign as the local pbest (Particle best value). Finally the select the highest fitness value it compare with the old fitness value if the new value is better means it will replace with the new fitness value that will assign as the global best. Finally update the velocity of the objective function using the equation (1) and position are updated by using the equation (2). The objective functions are tested with the JMeter Performance tool. The performance tools are used to identify the performance of the web application based on the following sever behaviour

a) Work load- used to find number of concurrent user in the particular time.
b) CPU Utilization- used to find how the minimum time of CPU utilized for the application in each second.
c) Bandwidth- used to find the bits per second used by the network interface.
d) Throughput-used to find the rate of which computer or network receives request per second.
e) Response time-used to find the time from when the user start the request till the first character of the response is received.
f) Hits per second-used to find the number of hits in the web server while every second in the load test.
g) Database locks-used to find the locking of the tables and monitoring the database carefully.
h) Thread count-The performance of the application measured based on the how many threads are running and active in the current state.
i) Number of position in the service queue-
j) Round trip time-used to find the outward trip time and return trip time.

k) Server mean service time- used to find the service time of the each application in the networks.

The JMeter Performance testing tools are mainly focuses the

a) Speed- which is used to determine how the application will be response quickly.
b) Scalability- which is used to determine how the applications are stable under the varying loads.
c) Stability which is used to determine how the application is stable under varying tools.

The Advantages of JMeter Performance tools

a) Open Source License- it is the open source free tool which is used to help the developers to provide the rights to access the source code for development.
b) Friendly GUI- its take very less time to familiar for user and easy to use.
c) Platform independent- The JMeter is the fully java desktop applications so it can run on the multiple platforms.
d) Full Multi-threading framework- This tools allows the concurrent and synchronized of the various sampling functions by the several group of the thread.
e) Visualize test results- The test result can be shown in various formats such as table, tree, chart and log file.

Pseudo code of proposed MOPSO

Input: Server behaviour such as work load, CPU, bandwidth, throughput, response time, hits per second, database locks, and thread count, number of position in the service queue, round trip time, and server mean service time.

$C_1, C_2 -$ cognetive and social learning rates
$r_1, r_2 -$ random numbers

Step 1: Initialize the parameter i= 1, 2 ...N
Step 2: Position initialization for each parameter
Step 3: Calculate fitness value based on the objective function using MOPSO.

Optimize objective function f (x) = $\sum_{m=1}^{M} w_m . f_m(x) w_m \in [0,1], \sum_{M=1}^{M} w_m = 1$

Step 4: Select gbest using Pareto optimal
Step 5: calculate each parameter fitness value compared with old pbest fitness value if new one is better means it will assign as the local pbest (particle best) value.
Step 6: select the highest fitness value it will compare with old fitness value if the new fitness value better means it will replace with the new fitness value that will assign as the gbest (global best)

If (iteration >1)
If ($gbest_{old}$ >$gbest_{new}$)
gbest = $gbest_{old}$
else
gbest = $gbest_{new}$

Step 7: update objective function velocity by Equation (1) and new position of the objective function by Equation (2) to all server behaviour parameter then again move to step 3.

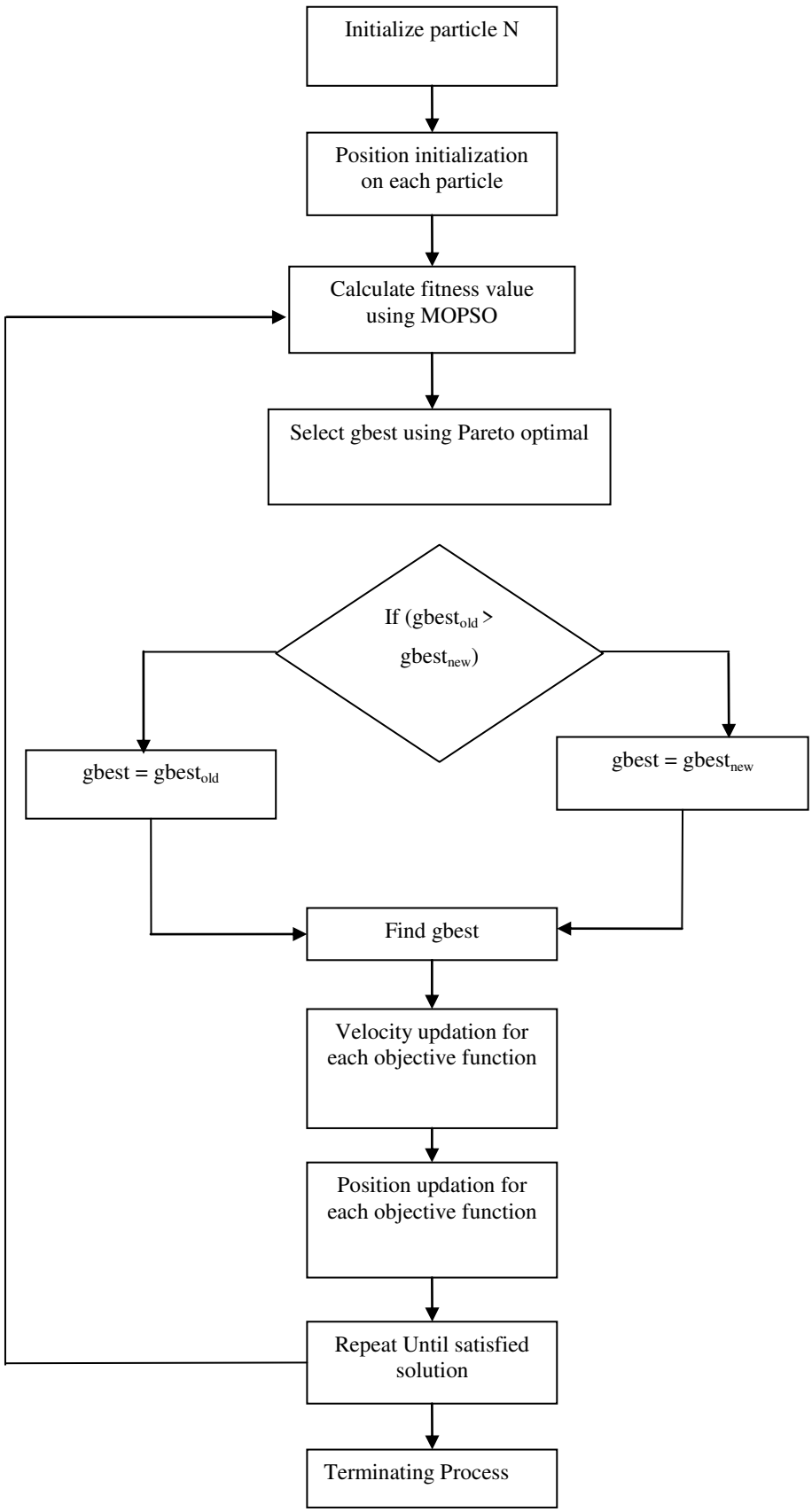
$$V_{ik}^{n+1} = V_{ik}^{n} + C_1 r_1^n (pbest_{ik}^n - X_{ik}^n) + C_2 r_2^n (gbest_k^n - X_{ik}^n) \quad (1)$$

$$P_{ik}^{n+1} = X_{ik}^{n} + V_{ik}^{n+1} \quad (2)$$

k- Number of objective function
i- Position of the parameter

Step 8: stop until criterion is satisfied (maximum iteration) repeat 2 to 7

3.2 Architecture diagram



3.3 Advantages of MOPSOPT approach

The advantages of using the MOPSO algorithm are:

- Effectively utilize the system resource to reduce the time.
- Only less number of parameters is used to optimization.
- Simple implementation
- Easily parallelized for concurrent processing

4. RESULT AND DISCUSSIONS

The performances of Proposed Multi object particle swarm optimization performance testing obtain in terms of workload, CPU Utilization, Bandwidth and Response Time.

A. Workload

The workload is defined as the number of concurrent user in the specified time. The unit is measure as the number.

Workload = Task X Time X Workload

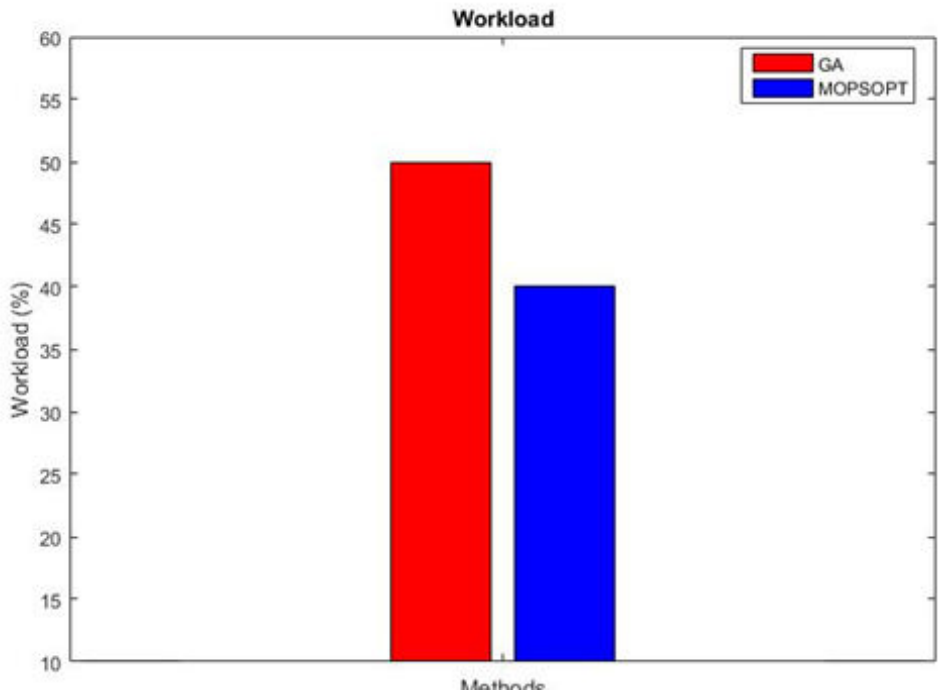


Figure-1. Comparison of Workload.

The Figure-1 shows that the comparison of workload between existing genetic methods and proposed multi object particle swarm optimization performance testing (MOPSOPT). The result shows that the proposed MOPSOPT minimize the workload compare to existing genetic algorithm.

B. CPU utilization

The CPU utilization is defined as the how much amount of the CPU is utilized for the application in each second. The unit is measured as the milliseconds.

CPU time (millisecs)/elapsed time

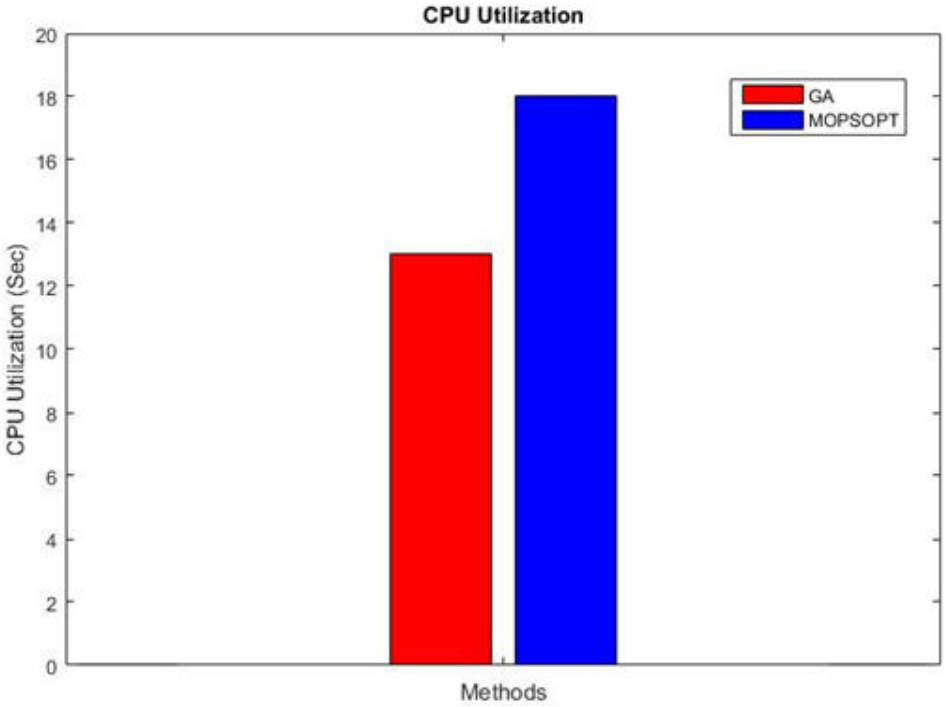


Figure-2. Comparison of CPU utilization.

The Figure-2 shows that the comparison of CPU Utilization between existing genetic methods and proposed multi object particle swarm optimization performance testing (MOPSOPT). The result shows that the proposed MOPSOPT increase the CPU utilization compare to existing genetic algorithm.

C. Bandwidth

The Bandwidth is defined as the bits per second used by the network interface. The Unit is measured as the bytes.

(Number of visitors X Pages X Page Size) /1000

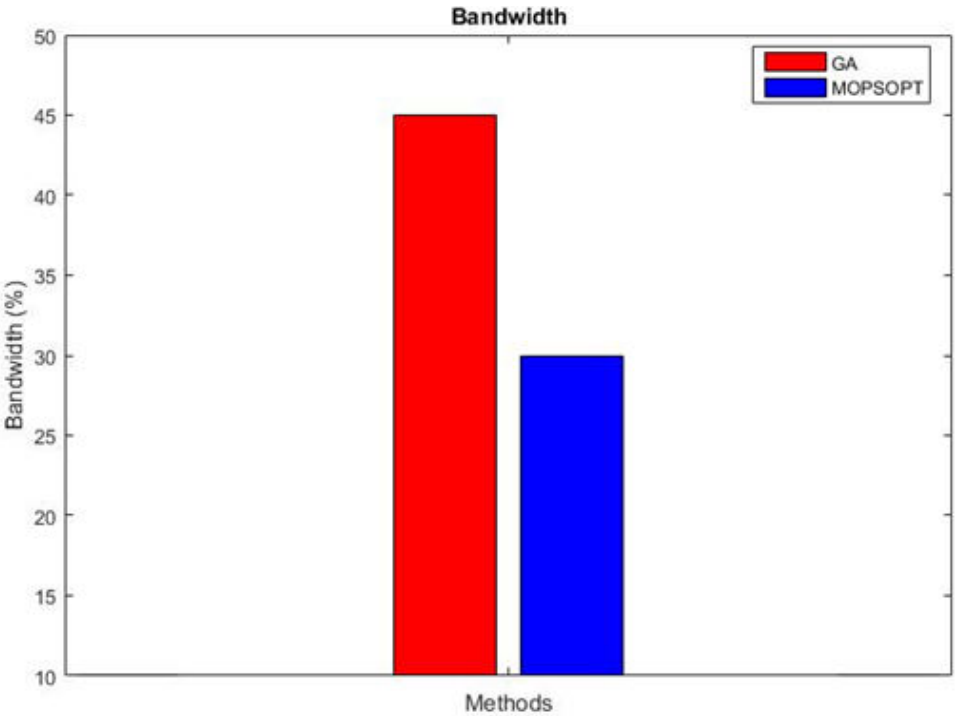


Figure-4.3. Comparison of bandwidth.

The Figure-4.3 shows that the comparison of Bandwidth between existing genetic methods and proposed multi object particle swarm optimization performance testing (MOPSOPT). The result shows that the proposed MOPSOPT minimize the bandwidth compare to existing genetic algorithm.

D. Response time

The response time is defined as the time from when the user enters in to the request till the first character of the response is received. The unit is measured as the millisecond

$T_{response} = n/r - T_{think}$

n = Number of concurrent users,

r = Number of request per second the server receives

T_{think} = Average link time (in seconds)

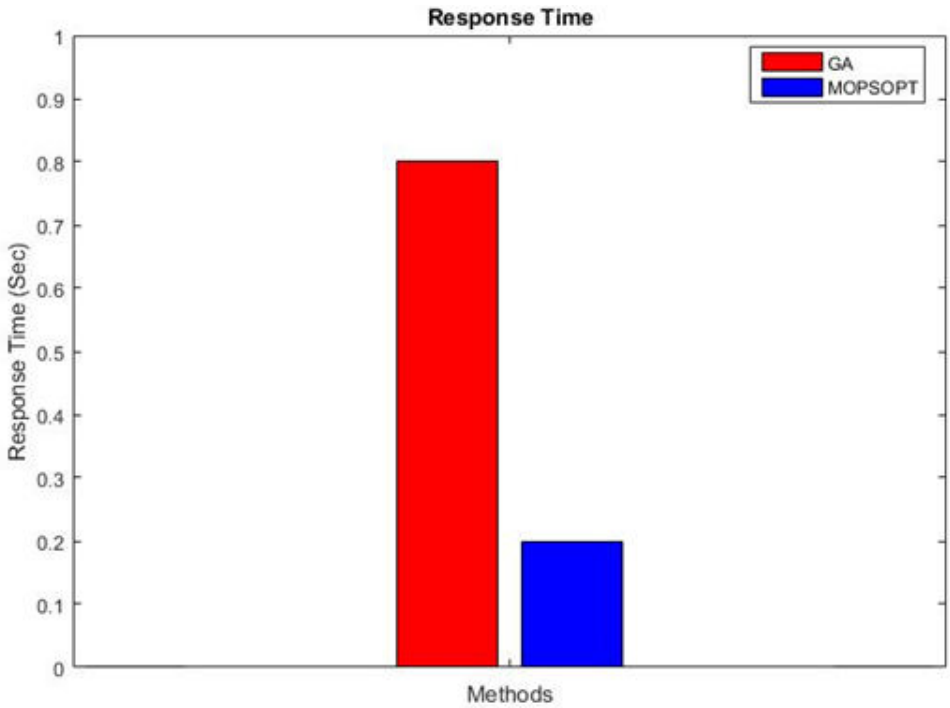


Figure-4. Comparison of response time.

The Figure-4 shows that the comparison of Response time between existing genetic methods and proposed multi object particle swarm optimization performance testing (MOPSOPT). The result shows that the proposed MOPSOPT minimize response time compare to existing genetic algorithm.

5. CONCLUSIONS

The proposed multi objective particle swarm optimization algorithm focuses the performances of the web services for the several web applications. In the MOPSOPT approach, the each particle consider as the server behaviour parameter. Then the fitness value calculated to optimize the parameter. It's used to achieve the objective functions such as less work load, maximum CPU utilization, less bandwidth and less response time. The JMeter performance testing is used to measure the performance by means of considering the various web services for web applications.

REFERENCE

[1] Zhu K., Fu J. & Li Y. 2010, June. Research the performance testing and performance improvement strategy in web application. In Education Technology and Computer (ICETC), 2010 2nd International Conference on. 2: V2-328, IEEE.

[2] Khan R. & Amjad M. 2016, April. Web application's performance testing using HP Load Runner and CA Wily introscope tools. In Computing, Communication and Automation (ICCCA), 2016 International Conference on pp. 802-806, IEEE.

[3] Sharma C., Sabharwal S. & Sibal R. 2014. A survey on software testing techniques using genetic algorithm. arXiv preprint arXiv:1411.1154.

[4] Zhongsheng Q. 2010. Test case generation and optimization for user session-based web application testing. Journal of Computers. 5(11): 1655-1662.

[5] Alshahwan N. & Harman M. 2011, November. Automated web application testing using search based software engineering. In: Proceedings of the 2011 26th IEEE/ACM International Conference on Automated

Software Engineering (pp. 3-12), IEEE Computer Society.

[6] Nivas T. 2011, July. Test harness and script design principles for automated testing of non-GUI or web based applications. In Proceedings of the First International Workshop on End-to-End Test Script Engineering (pp. 30-37), ACM.

[7] Babbar C., Bajpai N. & Sarmah D. K. 2011. Web Application Performance Analysis based on Component Load Testing. International Journal of Technology and Applied Science. 2: 22-28.

[8] Kalita M. & Bezboruah T. 2011. Investigation on performance testing and evaluation of prewebd: a. net technique for implementing web application. IET software. 5(4): 357-365.

[9] Zhou Q., Ye H. & Ding Z. 2012. Performance Analysis of Web Applications Based on User Navigation. Physics Procedia. 24: 1319-1328.

[10] Proko E. & Ninka I. 2013. Analyzing and Testing Web Application Performance. International Journal of Engineering and Science. 3(10): 47-50.

[11] Wang X., Zhou B. & Li W. 2013. Model-based load testing of web applications. Journal of the Chinese Institute of Engineers. 36(1): 74-86.

[12] Garousi V., Mesbah A., Betin-Can A. & Mirshokraie S. 2013. A systematic mapping study of web application testing. Information and Software Technology. 55(8): 1374-1396.

[13] Zhou Y. & Evans D. 2014, August. SSOScan: Automated Testing of Web Applications for Single Sign-On Vulnerabilities. In USENIX Security Symposium. pp. 495-510.

[14] Stepien B. & Peyton L. 2014. Innovation and evolution in integrated web application testing with TTCN-3. International Journal on Software Tools for Technology Transfer. 16(3): 269-283.

[15] Khan R. & Amjad M. 2016. Performance testing (load) of web applications based on test case management. Perspectives in Science. 8: 355-357.