EasyChair Preprint

№ 4159

Comparison Performance of 3 Chosen Load Balancing Algorithms in Cloud Computing using CloudAnalyst

Kai Ze Yeow and Wei Sheng Lim

September 9, 2020

Comparison Performance of 3 Chosen Load Balancing Algorithms in Cloud Computing using CloudAnalyst

Yeow Kai Ze
CA17070
Faculty of Computing
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan, Pahang.
kaizeyeow@gmail.com

Lim Wei Sheng
CA17109
Faculty of Computing
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan, Pahang.
ixywei@hotmail.com

Abstract — In the digital age, cloud computing is the development of distributed computing, parallel computing and grid computing. Cloud computing using resources sharing to accomplish collaboration efficiency and reduced system upgrade costs. Cloud computing is an emerging technology which still needs to optimize. Load balancing is the process of distributing the total load to the individual node to improve efficiency and response time. There are many existing algorithms that are implemented in cloud computing in order to improve the task scheduling issue. In this paper, comparison between 3 chosen Load Balancing Algorithms, which are Round Robin, Equally Spread Current Execution and Throttled in cloud computing will be focused and their respective performance is simulated and recorded using CloudSim.

Keywords — cloud computing, cloudsim, load balancing algorithms.

I. INTRODUCTION

A. BACKGROUND

Cloud computing is a trending computing with internet based have adopted many fields which provide a better and efficient solution to store and access files. Cloud computing's key aim is to provide high efficiency, scalability and quality on-demand computing resources in a distributed environment [1]. Cloud providers deliver flexible cloud services across large data centers. Hierarchical network design is the model most widely seen in data center networks [2]. By using the concept of distributed and parallel computing which use combining and sharing computing resources have been an industry game changer in businesses and information technology communities because enable the decreasing of hardware resource costs. This transforms the traditional data center to a boundary less data center which means no hub. Cloud allows users to access information at anytime from anywhere with an internet connection. In distributed computing, a job or a problem can be divided into many tasks, each of which is solved by one or more computers. Hence, it is essential to research some of the areas in cloud computing to improve the processing time and response time of the user associated with load balancing.

Load balancing is the process to assign load to every node of a distributed system to work faster and efficiently. It is a parallel system strategy that is used to achieve optimal system condition, in which workloads are spread uniformly among computers, which as its effect would decrease the execution time of the programs [3]. It executes load balancing tasks before execution of programs. In a cloud environment, load balancing is ensuring all nodes in the network are given an equal amount of work [4]. The importance of load balancing in cloud computing is distributing the load to reach maximum resource utilization. Load balancing algorithms will help in dividing resources efficiently by ensuring no node will be overloaded. The goals of load balancing are optimum resource utilization, maximum throughput, preventing overload and maximum response time. There is a general downside to static load balancing algorithms is that when the process is established, the final selection of a host for task allocation is done and can not be modified during process execution to adjust the system load [5].

B. PROBLEM STATEMENTS

Cloud computing offers flexibility to access data anywhere, any time. Load balancing is applied in cloud computing to prevent bottleneck of loads. There are a lot of algorithms that are implemented to improve cloud computing. The information of algorithms performance is limited. The processing time of static load balancing algorithms are concerned by people as it concludes the completion of the tasks and also the usage of the system. Furthermore, some of the static load balancing algorithms with better performances. Load balancing algorithms should notice that resource migration time is an important factor which significantly affects system

efficiency [6]. Cloud service provider efficiency is strongly affected by load balancing which is a very critical problem [7]. Therefore, researchers have created many load balancing algorithms to overcome the bottleneck of cloud computing, and there is a range of load balancing algorithms to accomplish better load balancing [8].

C. OBJECTIVE

These are three objectives of this paper:

1. To introduce chosen static load balancing algorithms
2. To review performance of chosen static load balancing algorithms
3. To analysis response times of different types of static load balancing algorithms

D. SCOPE

In this project, we will conduct research about the response time on 3 load balancing algorithms which are Round Robin, Equally Spread Current Execution and Throttled. We will use cloudsim to record their response time and do a comparison between them.

II. RELATED WORKS

A. LOAD BALANCING ALGORITHM

Load balancing algorithm is used to distribute requests from larger processing nodes into small processing nodes and then this will improve the performance of the system. In cloud computing, load balancing is used to distribute the computing resource between all the nodes. It can maximize resource utilization and satisfy users because of its fair allocation of computing resources. Therefore, maximizing resource utilization can help in reducing resource consumption and avoiding bottlenecks.

1. Round Robin Load Balancing Algorithm

Round robin is the most common and easy scheduling algorithm. It works based on time quantum and process in round form. In this load balancing, the datacenter controller assigns the request to VMs on a rotational basis. The process will wait for the last process to proceed. The major problem is that the new process will take a lot of time to complete if previous processes are huge and need a large amount of time. The Round Robin technique does not take resource abilities, priority, and assignment duration into account. The higher priority and the lengthy jobs, however, end up with the higher response times [9].

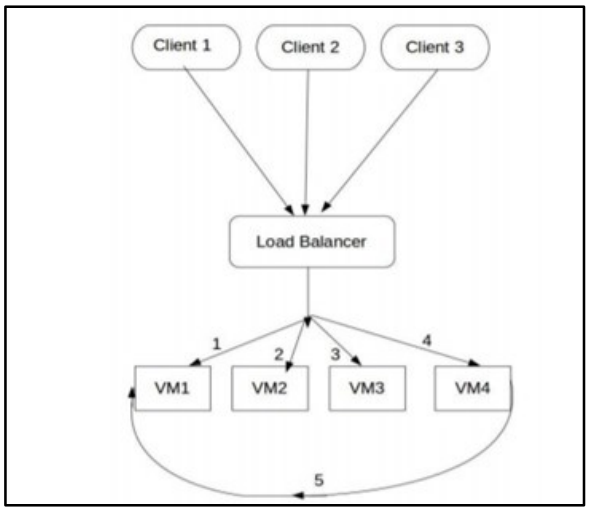


Figure 2.1.1: Round robin load balancing

2. Equally Spread Current Execution

Equally spread current execution algorithm functions on a continuous basis on the queue and randomly passes it to another virtual machine and distributes it by measuring the size of the coming load, then distributes it to a virtual machine that has light work [10].

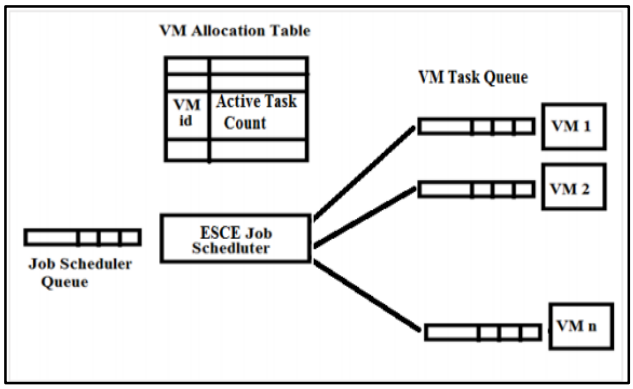


Figure 2.1.2: Equally spread current execution

3. Throttled Load Balancing Algorithm

Throttled load balancing algorithm contains an index of virtual machines and the states, which is available or busy. The load balancer has the list of VMs. When a new request comes in, the load balancer will assign the request to the first available VMs .If no VMs are available, then the request needs to wait in queue for fast processing ,and it returns -1 to the datacenter and this will reduce the cost of usage.

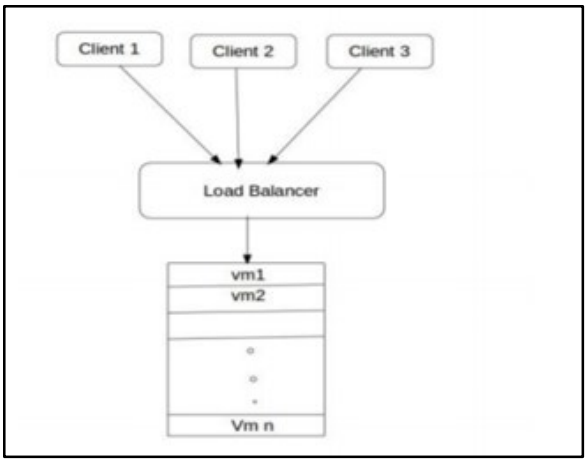


Figure 2.1.3: Throttled load balancing

B. RESULTS OF SIMULATION

Key setup is set as shown in the figure 3.1, 3.2, 3.3 and figure 3.4 reliant on which we have found the result for the three figuring that is RR, ESCE, and TLB. The parameters secluded are response time. The response time was tested in 2 cases, which is 3 hour case and 6 hour case[11][12].

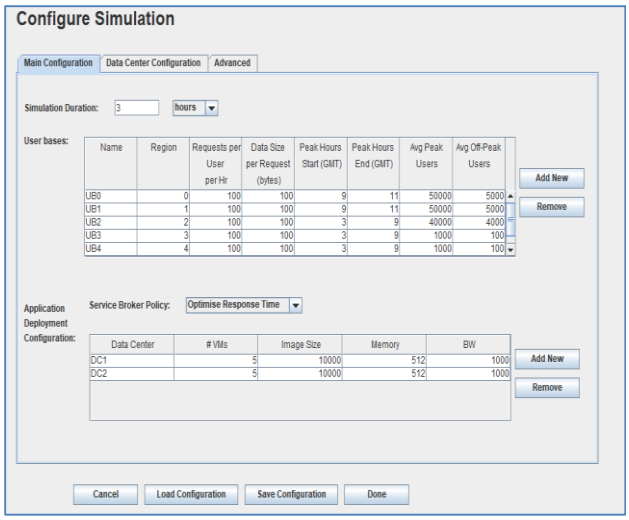


Fig.2.2.1: 3 Hours Main Configuration

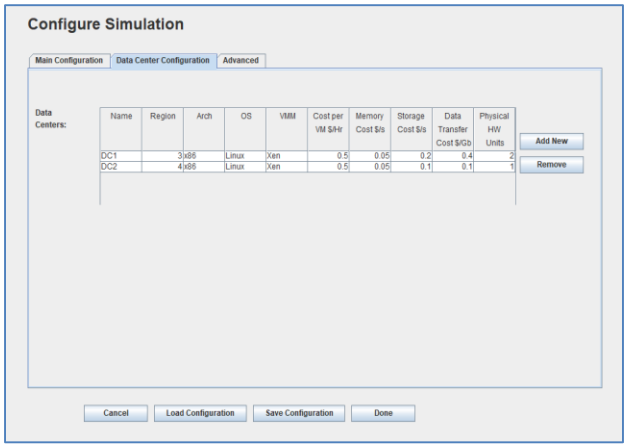


Fig.2.2.2: 3 Hour Data Centre Configuration

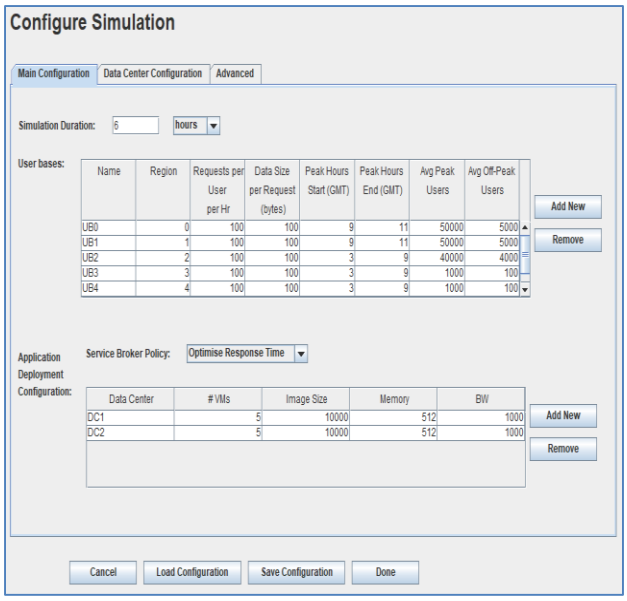


Fig.2.2.3: 6 Hour Main Configuration

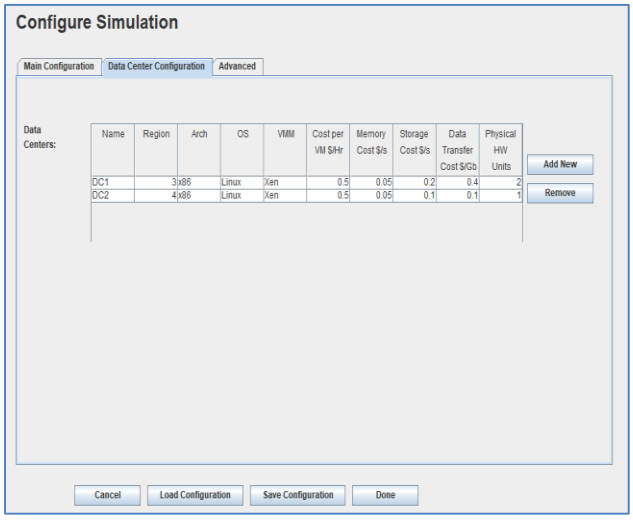


Fig.2.2.4: 6 Hour Main Configuration

A) Case 1:
• Simulation run 3 hours
• Six customer bases UB0, UB1, UB2, UB3, UB4, UB5 in territory 0, 1, 2, 3, 4 and 5 correspondingly.
• Application sending setup data centers DC1 in area 3, DC2 in locale 4[13].

B) Case 2:
• Simulation length 6 hours
• Six customer bases UB0, UB1, UB2, UB3, UB4, UB5 in zone 0, 1, 2, 3, 4 and 5 correspondingly.
• Application sending setup data centers DC1 in zone 3, DC2 in locale 4.

C) Comparison of response time for the load balancing algorithm that is Round Robin, Equally Spread Current Execution, Throttled Load Balancing [14].

Case 1: Response Time in Simulation 3 Hours

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	174.65	34.37	466.13
Data Center processing time:	1.87	0.02	11.59

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB0	54.33	34.37	70.83
UB1	200.90	141.71	264.00
UB2	300.56	205.94	466.13
UB3	50.16	38.86	61.36
UB4	50.06	38.65	62.91
UB5	50.11	37.55	62.38

Fig.2.3.1: Round Robin Algorithm Overall Response Time

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	174.60	34.37	466.13
Data Center processing time:	1.81	0.02	10.87

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB0	54.20	34.37	70.24
UB1	200.87	141.71	264.00
UB2	300.56	205.71	466.13
UB3	50.13	38.86	61.36
UB4	50.08	38.65	63.90
UB5	50.09	37.55	62.38

Fig.2.3.2: Equally Spread Current Execution Algorithm Overall Response Time

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	173.94	34.37	462.13
Data Center processing time:	1.14	0.02	8.04

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB0	52.33	34.37	69.58
UB1	200.85	141.71	264.00
UB2	300.52	205.94	462.13
UB3	50.14	38.86	61.36
UB4	50.08	38.65	63.90
UB5	50.07	37.55	62.38

Fig.2.3.3: Throttled Algorithm Overall Response Time

Case 2: Response Time in Simulation 6 Hours

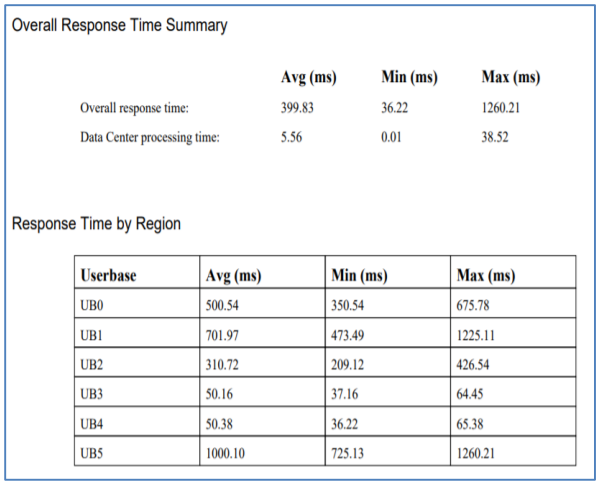
Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	399.97	36.22	1260.27
Data Center processing time:	5.70	0.01	41.79

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB0	500.56	350.55	675.76
UB1	702.04	473.49	1225.10
UB2	310.91	209.85	425.66
UB3	50.16	37.16	64.69
UB4	50.39	36.22	65.38
UB5	1000.16	725.13	1260.27

Fig.2.3.4: Round Robin Algorithm Overall Response Time



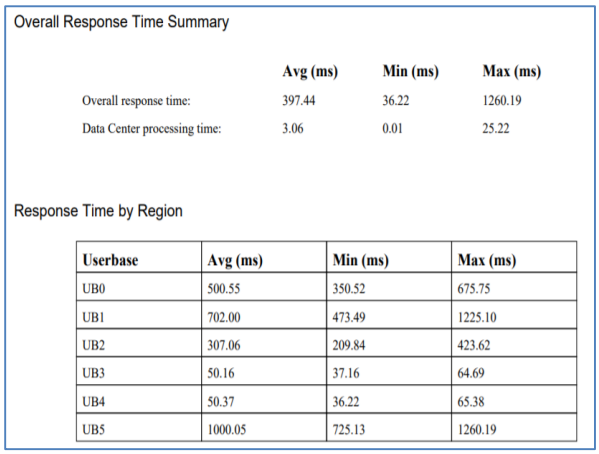
Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	399.83	36.22	1260.21
Data Center processing time:	5.56	0.01	38.52

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB0	500.54	350.54	675.78
UB1	701.97	473.49	1225.11
UB2	310.72	209.12	426.54
UB3	50.16	37.16	64.45
UB4	50.38	36.22	65.38
UB5	1000.10	725.13	1260.21

Fig.2.3.5: Equally Spread Current Execution Algorithm Overall Response Time



Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	397.44	36.22	1260.19
Data Center processing time:	3.06	0.01	25.22

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB0	500.55	350.52	675.75
UB1	702.00	473.49	1225.10
UB2	307.06	209.84	423.62
UB3	50.16	37.16	64.69
UB4	50.37	36.22	65.38
UB5	1000.05	725.13	1260.19

Fig.2.3.6 Throttled Algorithm Overall Response Time

C. RESULTS ANALYSIS AND SUMMARY

Table 1: Overall Response Time in Simulation 3 Hours

Load Balancing Algorithm	Overall Response Time(ms)		
	Avg	Min	Max
RR	174.65	34.37	466.13
ESCE	174.60	34.37	466.13
TLB	173.94	34.37	462.13

Table 2: Overall Response Time in Simulation 6 Hours

Load Balancing Algorithm	Overall Response Time(ms)		
	Avg	Min	Max
RR	399.97	36.22	1260.27
ESCE	399.83	36.22	1260.21
TLB	397.44	36.22	1260.19

In summary, we can see that in table 1 which Throttled load balancing algorithm has the best average response time in case 1. In table 2, Throttled load balancing algorithm also has the best average response in case 2.So we can conclude that the throttled load balancing algorithm has the best overall response time among these 3 load balancing algorithms.

III. OPEN RESEARCH ISSUES

However, the paper only consists of the comparison of performance between limited load balancing algorithms and based on the two scenarios. Many of the scopes beyond this report can be discussed in the future, namely:

I. Analyzing the performance of load balancing algorithm affect by user base usage pattern

II. Discussing the performance of the centralized and distributed data center at each scenario.

IV. CONCLUSION

Load balancing helps distribute the total request to individual nodes to enhance resource utilization and response time. It also helps avoid heavy loads on individual nodes because it has fair allocation .To compare the response time of each load balancing algorithm, we choose cloud analysts to conduct our research. The result showed throttled load balancing algorithm has the better response time in cloud analysts. However, other algorithms like round robin algorithm and equally spread current execution algorithm are also efficient for user requests load distribution among the VMs [15].

V. REFERENCES

[1] X. Xu, “From cloud computing to cloud manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 1, pp. 75–86, 2012.

[2] M. Devine and G. Lin, “The Role of Networks in Cloud Computing,” 2010.

[3] H. Rahmawan and Y. S. Gondokaryono, “The simulation of static load balancing algorithms,” *2009 International Conference on Electrical Engineering and Informatics*, 2009.

[4] K. Garala, N. Goswami and P. D. Maheta, "A performance analysis of load Balancing algorithms in Cloud environment," *2015 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, 2015, pp. 1-6, doi: 10.1109/ICCCI.2015.7218063.

[5] S. Sharma, S. Singh, and M. Sharma, “Performance Analysis of Load Balancing Algorithms,” *World Academy of Science, Engineering and Technology International Journal of Civil and Environmental Engineering*, vol. 2, no. 2, 2008.

[6] V. R. Kanakala, V. Reddy, and K. Karthik, “Performance analysis of load balancing techniques in cloud computing environment,” *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Mar. 2015.

[7] V. K. Reddy, K. D. Surya, M. S. Praveen, B. Lokesh, A. Vishal, and K. Akhil, “Performance Analysis of Load Balancing Algorithms in Cloud Computing Environment,” *Indian Journal of Science and Technology*, vol. 9, no. 18, pp. 1–7, 2016.

[8] R. Ramya, S. Puspalatha, T. Hemalatha and M. Bhuvana, "A Survey on and Performance Analysis of Load Balancing Algorithms using Meta Heuristics approach in Public Cloud-Service Provider’s Perspective," *2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW)*, Erode, India, 2018, pp. 380-385, doi: 10.1109/I2C2SW45816.2018.8997138.

[9] D. C. Devi and V. R. Uthariaraj, “Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks,” *The Scientific World Journal*, vol. 2016, pp. 1–14, Feb. 2016.

[10] I. N. Falisha, T. W. Purboyo, and R. Latuconsina, “Experimental Model for Load Balancing in Cloud Computing Using Equally Spread Current Execution Load Algorithm,” *International Journal of Applied Engineering Research*, vol. 13, no. 2, pp. 1134–1138, 2018.

[11] Patel, Sandip & Patel, Ritesh & Patel, “Hetal & Vahora, Seema. (2015). CloudAnalyst : A Survey of Load Balancing Policies.”, *International Journal of Computer Applications*, 117. 975-8887. 10.5120/20679-3525.

[12] Durgesh Patel , Mr. Anand S Rajawat, “Efficient Throttled Load Balancing Algorithm in Cloud Environment’, *International Journal of Modern Trends in Engineering and Research*, pp. 463 - 480, 2015.

[13] Nuaimi, K. A.; Mohamed, N.; Nuaimi, M. A. Al-Jaroodi, J., “A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms”, *Second Symposium on Network Cloud Computing and Applications (NCCA)*, IEEE, Print ISBN: 978-1-4673-5581- 0, pp. 137- 142, 2012.

[14] V. Bagwaiya and S. K. Raghuwanshi, "Hybrid approach using throttled and ESCE load balancing algorithms in cloud computing," *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, Coimbatore, 2014, pp. 1-6, doi: 10.1109/ICGCCEE.2014.6921418.

[15] Mahalle, Hemant S., Kaveri, Parag R. Chavan, Vinay, “Load Balancing On Cloud Data Centres”, *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, ISSN: 2277-128x, vol. 3.