© INNOVATIVE RESEARCH THOUGHTS | Refereed | Peer Reviewed | Indexed
ISSN : 2454 – 308X | Volume : 04 , Issue : 04 | January – March 2018

SIMULATION OF LOAD BALANCING IN CASE OF FOG COMPUTING

Nancy Malik, nancymalik339@gmail.com

Abstract: - This paper has represented the simulation of resource allocation in case of Fog computing. The scheduling algorithm has been considered in order to perform resource scheduling. This work has been based on fog based cloud computing and here scheduler has been connected to fog nodes. Several processes are going to be managed by scheduler. This scheduler would serve to several processors from fog nodes which have been connected to IP network. IP network is interface between fog node and cloud storage. This system consists of multiple sub system such Gang scheduler, Fog node, IP network, cloud storage, Distributed intelligence. In this research Distributed intelligence is embarrassingly parallel. This is able to exploit computation of large scale. Spatial distribution of computing resources will also be get by it. The problem would be solved by these features. It needs the processing of very large data sets.

Keyword: - Fog Computing, Scheduling, Cloud models, Execution time, Throughput, Load balancing.

1. INTRODUCTION

Fog computing is a term defined by Cisco. It refers to elaborate cloud computing .It is to the edge of an enterprise's network. It is also known as Edge Computing or fogging. Fog computing makes it available the operation of compute, storage, and networking services between end devices and cloud computing data centres. Fog computing essentially extends cloud computing [1]. Its services to edge of network, bringing advantages & power of cloud closer to where data is created and acted upon. The objective of fog computing of fog is to enhance efficiency and decrease amount of data transported to cloud for processing, analysis and storage. This is often done to improve efficiency, though it may also be used for security and compliance reasons [2].

2. RELATED WORK

The environment for load balancing scheduling algorithm is provided by Fog Computing environment. The intermediate layer between cloud layer and client layer is Fog Layer. It has been introduced to enhance the efficiency of cloud computing environment. It does proper utilization of bandwidth. As the data is transmitted or exchanged. It is between cloud and fog. Due to this processing get reduced [3].

They show an approach to secure business data in the cloud. Once unauthorized access or exposure is suspected, and later verified, with challenge questions for that instance, then They inundate the malicious insider with bogus information in order to dilute the user's real data [4].

This work presents a novel Combined Fog-Cloud architecture .It is consisting of a dual-layer. Fog aim to lessen the Cloud access delay into an IoT scenario. Taking this approach, a service may avail geographically distributed network elements in scenarios. This is such as Smart Cities or Smart Transportation Systems. It is among others. It diminishes the necessity of demanding further Cloud resources. It prevents high delays. The show results prove the advantages of service distribution. It is among multiple low-delays. Fog nodes avoiding the high delay access on upper layer [5].

In this paper, researchers proposed a flexible resource allocation strategy for fog computing .It is based on PTPN to enhance the efficiency of fog resources utilization. It satisfies the users' QoS requirements. It has constructed the PTPN models of tasks in fog computing. It is prepared in accordance with the features of fog resources. These are used to guide flexible resource allocation in fog computing. Saying further algorithms were proposed to estimate completion time of task. It computes the credibility evaluation of resources. It allocates fog resources dynamically. The simulation was carried in a Dawn parallel machine and Linux cluster [6].

3. PROPOSED WORK

This section presents the proposed model which is more convenient to run the proposed algorithm. Next this section also describes the proposed algorithm in detail followed by its description. They are discussed as follows

3.1. PROPOSED MODEL

The proposed model is based on fog based cloud computing where scheduler is connected to fog nodes. The multiple processes will be managed by scheduler and gang scheduler will provide service to multiple processors from fog nodes that are connected to IP network. IP network is acting as interface between fog node and cloud storage. Proposed system is the integration of multiple sub system such Gang scheduler, Fog node, IP network, cloud storage, Distributed intelligence The intelligence of distributed things is embarrassingly parallel. It is able to exploit computation of large scale. It also carries spatial distribution of computing resources. These type of properties allow it to solve problems .It requires the processing of very large data sets. DAI systems comprises of autonomous learning processing nodes .These are distributed at a very large scale.

Algorithm

1. Capture request from multiple processes and transfer to gang scheduler.
2. Gang scheduler would allot resources to different process after getting feedback from fog nodes.
3. Fog node would be the infrastructure implementing the said mini-cloud. Other proposals have their own definition of what a fog node is, usually in relation to a specific edge device, a specific use case or an application.
4. IP network is here to share the load of cloud using distributed intelligence mechanism.
5. DAI is an approach for solving complex planning, learning, and problems of decision making. The nodes of Distributed intelligence mechanism can act independently. Partial solutions are integrated by communication between nodes. These are often asynchronously. Distributed intelligence mechanism systems are elastic and robust by their virtue .These are loosely coupled by necessity.

3.2.Proposed algorithm:-

In proposed algorithm the scheduling metrics are divided into two parts namely external cloud and internal cloud.

Description of Proposed Algorithm:-

1. Perform the classification of jobs according to complexity.
2. m1, m2 are used as a scheduling matrixes.
3. Place easy jobs at outer edge of cloud in m1
4. Place complex jobs at internal of cloud in m2
5. The algorithm running on outer edge (Fog cloud)
    a. do{
    b. for I1= Row having least population to the row having most population
    c. for J1=most populated row to least populated row
    d. for all the jobs at the outer edge of cloud in row I1
    e. if they can be moved to row J1,
    f. then move and break
    g. }while matrix m1 changes
6. The algorithm running on internal of cloud system
    a. do{
    b. for I2= from row having maximum population to row with minimum population
    c. for J2= from the row having maximum population to row with minimum population
    d. for all jobs at the internal of cloud in row I2
    e. if they can be moved to row J2,
    f. then move and break
    g. }while matrix m2 changes
7. **Stop**

[4]PROPOSED IMPLEMENTATION

Proposed implementation consist two section

Section 1

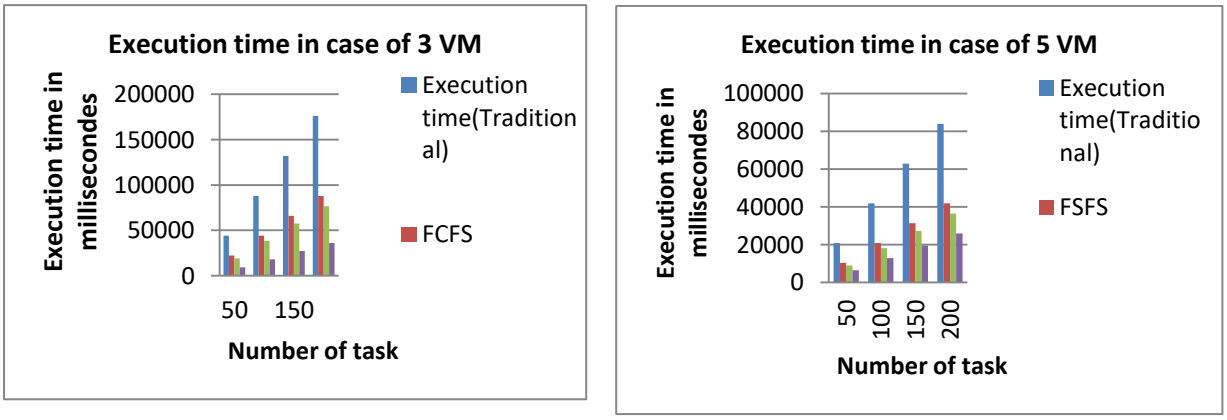
Here in this section cloud simulator based implementation for traditional and proposed work has been made.

Section2

Here in this section the reading would be taken from traditional and proposed work. Then the Matlab based graph would be plotted in both circumstances.

Execution time:-

The following chart represents the comparative analysis of **execution time** in case of Traditional gang scheduling, FCFS, ROUND ROBIN and PROPOSED WORK. We calculate the execution time for different set of task after executing then on the different virtual machines. And fig. 2 (a) to (d) shows the performance of the algorithm. Here the **execution time** of proposed work would be **less** than other approaches in case of number of task in case of 3, 5VM. And the proposed algorithm perform better all three existing algorithm.



2(a) **2(b)**

Fig.2 execution time: (a) 3 VM, (b) 5VM

Throughput:-

The following chart represents the comparative analysis of **throughput** in case of Traditional gang scheduling, FCFS, ROUND ROBIN and PROPOSED WORK. We calculate the throughput for different set of task after executing then on the different virtual machines. And fig. 3 (a) to (d) shows the performance of the algorithm. Here the throughput of proposed work would be **more** than other approaches respect to number of task in case of 3vm. And the proposed algorithm perform better then three existing algorithm.

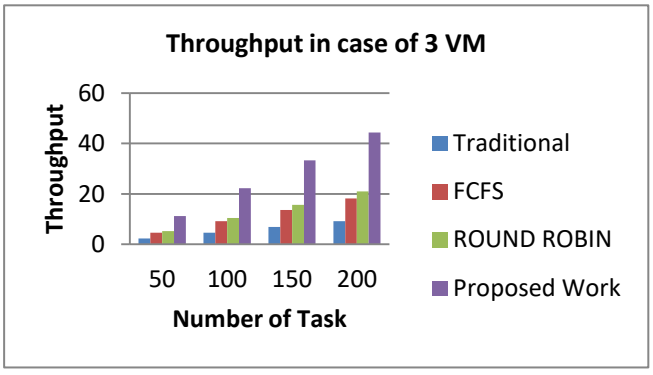
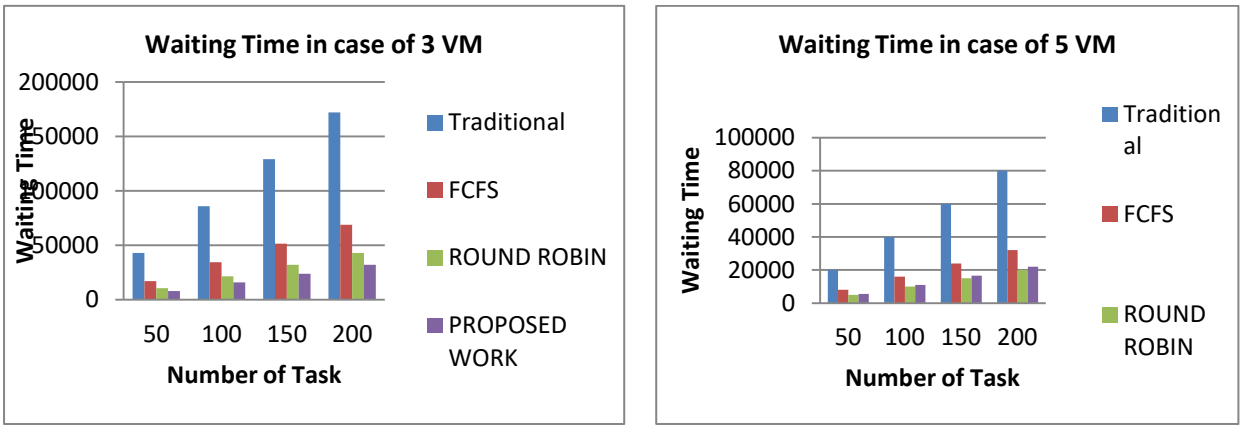


Fig.3 Throughput: (a) 3 VM

Waiting time:-

The following chart represents the comparative analysis of **waiting time** in case of Traditional gang scheduling, FCFS, ROUND ROBIN and PROPOSED WORK. We calculate the waiting time for different set of task after

executing then on the different virtual machines. And fig. 4 (a) to (d) shows the performance of the algorithm. Here the **waiting time** of proposed work would be **less** than other approaches in case of number of task are 3, 5, VM. The proposed algorithm perform better then all three existing algorithm.

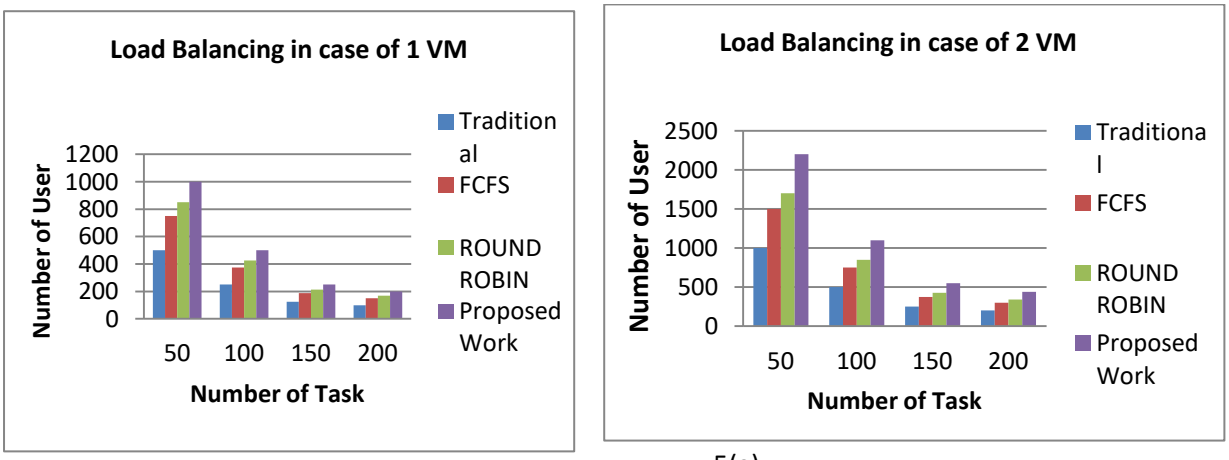


4(a) 4(b)

Fig.4 Waiting time: (a) 3 VM, (b) 5VM

Load balancing: -

In the fundamental **load balancing** setup, the requests of the clients are sendt to the IP address of a virtual server that is configured. This virtual server allocates them to the **load**-balanced application servers. This is according to a preset pattern. It is called the **load balancing** algorithm. **Load balancers** are used to increase capacity or concurrent users and reliability of applications. The following chart represents the comparative analysis of **load balancing** in case of Traditional gang scheduling, FCFS, ROUND ROBIN and PROPOSED WORK. We calculate the load balance for different set of task after executing then on the different virtual machines. And fig. 5 (a) to (d) shows the performance of the algorithm. Here the **load balancing** of proposed work would be **more** than other approaches in case of number of task in case of 1, 2, 3, 5VM. And the proposed algorithm perform better then all three existing algorithm.



5(a)

Fig.5 Load balancing: (a) 1 VM, (b) 2 VM

[5] CONCLUSION

In proposed work, burden of jobs have been shared in internal-cloud and external-layer of cloud. The proposed work schedule arrived jobs according to nature. Some task need to be processed by internal cloud. But some task processed externally. This has lead to increase in performance. The proposed work has thus played a

significant role for increasing the overall performance during fog computing operations. Here the compilation time, waiting time, load balancing, though put and execution time , resource utilization has been considered. In case of proposed work the waiting, compilation and execution time has been decreased. But the throughput and load balancing got increased. The proposed work has been proven better approach as compare to FCFS, ROUND ROBIN and traditional work.

[6] FUTURE SCOPE

In proposed work has increased the efficiency by reducing the burden of jobs. Such work is beneficial in case of internal-cloud and external-layer of cloud. The proposed work would handle the job by scheduling arrived jobs. It would be done according to type of jobs arrived. The proposed work would play a significant role in increasing the overall performance in case of fog computing operations. This system would take less compilation time, waiting time. Proposed work would boost the capability of load balancing and though put. The proposed work would be better approach from FCFS, ROUND ROBIN and traditional work.

REFERENCE

[1]. P.V. Patil, "Fog Computing", International Journal of Computer Applications (0975 – 8887), pp.1-6, 2015.
[2].I.Stojmenovic, S.Wen, "The Fog Computing Paradigm: Scenarios and Security Issues", Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, IEEE, ACSIS, Vol. 2, pp.1-8, 2014, DOI: 10.15439/2014F503.
[3]. M.Verma, N.Bhardawaj, A.K.Yadav, "architecture for Load Balancing Techniques for Fog Computing Environment", I.J. Information Technology and Computer Science Volume 6 No.2, pp.269-274, April 2016, DOI:10.090592/IJCSC.2015.627.
[4]. J.Khandagale, D.Fodse, P.Sul, P.Patil, "FOG Computing", International Journal of Engineering Science and Computing Volume 6 Issue No. 3, pp.2705-2709, March 2016, DOI:10.4010/2016.634.
[5]. V.B.C.Souza, W.Ramírez, X.Masip-Bruin, E.Marín-Tordera, G.Ren, G.Tashakor, "Handling Service Allocation in Combined Fog-Cloud Scenarios", IEEE - Next-Generation Networking and Internet Symposium, 2016.
[6]. L.Ni, J.Zhang, C.Jiang, C.Yan and K.Yu, "Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets", IEEE Internet of Things Journal VOL. *, NO. *, pp.1-14, 2017, DOI: 10.1109/JIOT.2017.2709814.
[7].Sudhir Singh, "Performance Optimization in Gang Scheduling in Cloud Computing", IOSR Journal of Computer Engineering (IOSRJCE) Volume 2, Issue 4, pp.49-52, August2012.
[8]. Sawati, Nikita," Enhancing Capability of Gang Scheduling by Integration of Multi Core Processors & Cache", IJSRE Volume 4 Issue 8, pp.5625-5633, August 2016, DOI: http://dx.doi.org/10.18535/ijsre/v4i08.06.
[9].Y.Zhang, H.Franke, J.E Moreira, A.Sivasubramaniam, "Improving Parrallel Job Scheduling by Combining Gang Scheduling and Backfilling Techniques",IEEE,pp. 133-142,2000.