ISSN 2347 - 3983

Volume 8. No. 9, September 2020

International Journal of Emerging Trends in Engineering Research

Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter07892020.pdf

https://doi.org/10.30534/ijeter/2020/07892020

Layered Programming Model for Resource Provisioning in Fog Computing using Yet another Fog Simulator

Kolla Bhanu Prakash[1*], Sk. Nazeer[2], Pradeep Kumar Vadla[3], Subrata Chowdhury[4]

[1] Dept. of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, A.P, India

[2]Professor, Dept. of Computer Science & Engineering, Bapatla Engineering College, Bapatla, A.P, India

[3]Dept of Computer Science and Engineering, B.V. Raju Institute of Technology, Narsapur, Medak, Telangana, India

[4] Dept of Computer Science and Engineering, Vels University VISTAS, Pallavaram, India

drkbp@kluniversity.in*

ABSTRACT

The huge usage of IoT devices and their deployment in multiple electronic machines need to address complex computing while processing their information from that device to any other network or cloud. Fog computing address those computing needs of IoT devices by managing the connectivity at the edge of their network by allocating resources virtually in cloud. This paper provides a perspective survey on fog computing providing features on multi-tier or distributed architecture concept of it, along with applicability of programming models and management of resource provisioning among the network of IoT devices.

Key words: Edge network, Architecture concept, Programming models and Resource provisioning**.**

1. INTRODUCTION

The current trends of using IOT devices in daily life has become mandatory for performing many tasks of personal and business operations. Performance of these devices depends on many constraints of network communication where current trending technologies need to evolve to solve their run time needs of internet connectivity, data storage and power backup.

The Figure 1.1 gives the four phases of computing models which are meant for providing resources to IoT devices for processing and send the results to the users who are associated with that device interaction.

CLOUD COMPUTING: It is a central processing model where large amounts of data is processed and can be accessed through internet. It is easily scalable, but there are a lot of disadvantages namely high latency, more bandwidth cost, consumes more power, less secure, it is always online mode, and many other security, privacy and legal issues occur if any raw data is sent over the internet. It also follows service on demand policy to provide resources.

EDGE COMPUTING: It is a model which can work without cloud or fog computing where it is limited to the peripheral layers only. Here wiring is done using programmable automation controllers which are embedded with intelligence. These PAC's are used to find out what type of data should be stored locally or sent to the cloud for further analysis.

Though it has intelligence, it is less scalable when compared to fog computing. And it also doesn't use cloud and so there is no chance of resource pooling [32]. And also it may not be preferred for IT/OT team. It is just a subset of Fog computing and just its intelligence is up to the network edge only.

DEW COMPUTING: this computing provides the users with rich functionality without the cloud services and also has the ability to work using cloud services. It mainly depends on the "micro-services" approach and has to be embedded with all the network technologies and also performs in hybrid behavior..

This dew computing has the power to mitigate its problem by itself and also enhances its computational power accordingly. It can change its nature and also changes its way of computation and processing with respect to the problem. It is highly scalable and is transparent in nature.

FOG COMPUTING: It is the recent most developed distributed computing model which it is an extended form of edge of the network. It is used for overcoming high legacy, limited Quality of Service, poor throughput, bandwidth, latency, connectivity problems that exist in cloud computing. It is used by gateways and cloudlets for better computing.

But the main drawback is that as there is an increase in the number of links, where the data is being moved from physical layer to digital layer there will be a failure in potential point.

Here these fog nodes can be deployed in all private, public and hybrid modes and here data that is less sensitive and which is of not real-time can be sent to cloud for computing.
MIST COMPUTING: It acts as an intermediate layer between the cloud and fog / edge computing architectures

where microchips, microcontrollers [33] are used in the network cables. It can take decisions by itself even in the local mode and works in hand with both the cloud and fog computing. This may not need cloud.

Fog computing in general is an advancement of cloud computing in which these devices computations are managed at the edge of cloud network near to the user premises where these IoT devices are used. Fog computing helps to prevent cloud involvement through data pre – processing [35] and filtration Data [34] ; generated from the increasing number of devices and other IoT connected devices. This rapid increase use of edge resources leads to communication and execution of tasks. Providing mechanisms for federated data processing and for virtualized resource scheduling in fog landscapes [37] is a specific research issue. A creative program to address the problem environment, i.e. a framework for computing fog [36], is needed.

Fog computing introduces a new measurement model in which capacity computing, capacity storage and networking services are positioned at the edge, instead of over the Internet in the cloud. [13], [14] In addition, the management computing, networking and warehousing services and the computing of network infrastructure is possible fog computing by using a coordinated harmonized method. [15], [16] It provides a ubiquitous environment where communicating devices are potentially cooperating for performance store. They process tasks which can be carried out with & without coordination [18] with applications centralized in the cloud. Accordingly, fog computing is a generalized novel computing model for use in Internet of everything Applications [17].

Practical application requires care various aspects, such as design process, implementation design, integration [19] of the software, data protection and computing, management of resources, and networking. Bonomi et al. introduced fog computing as technological advances for IoT in [4], 2012. This innovative distributed prototype embodies the idea of computer & storage efficiency closer to the IoT appliance generating data at the edge.

With aim of reducing the distance between edge devices and this paradigm is the closest processing unit and adds an additional resource-rich layer for IoT appliance, that is to say fog cells [11]. These fog cells have their own computing & storage efficiency for processing requests for tasks, filtration, and data pre - processing.

The broad geographical distribution of fog computing is another important extension of these additional devices are designed for smooth and efficient service delivery, even when necessary moving devices, such as smart cars and mobile phones. [4, 5, 11] Fog cells are situated at the network edge in more detail, with the intention of reducing the Jitter, processing time of tasks and the amount of data transmitted over the network.

Fog cells checks with the requests received from IOT appliance and judges whether these are processed locally or through cloud. [6]

2. LITERATURE SURVEY

2.1 Architectures of Fog Computing and its Concepts

In 2012, Bonomi et al.[5] coined first notion of fog computing. In their research they describe the significance of a new fog computing framework.

The main features of the IoT environments are (i) limited bandwidth and location-consciousness, (ii) high geographic distribution, (iii) large sensing applications, (iv) mobility assistance; and (v) heterogeneous unit.

This advised framework targets to address these specifications & different layers of infrastructure from the bottom to the top: IoT Appliances, multi-service edge where the fog cells, the core network and the fog cells are located in the cloud. The framework gets more specific in further research [4] with a complete discussion of technical and operational components and an abstract model of software prototype [38].

A fog abstraction layer, fog service orchestration layer, and north bound APIs were initiated into this software prototype model. The abstraction layer provides an abstract version of the diverse systems, allowing more simplified execution of services. The orchestration layer [39] is used for compilation of programs, the organizing & recording of activities & the installation and destruction of fog cells, the software agents.
North-bound APIs [21] are specified to allow distributed communication between those software agents. North-bound API [40], [41] demonstrates that one unit can communicate with one more unit of a superior level of hierarchy. Flavio Bonomi mentioned that the focus has been on improving the performance of internet using fog computing prototype [42]. Zhu et al. [12] used an identical design and definition to enhance the efficiency of websites as in the preceding articles.

Dastjerdi et al. [7] introduced a layered framework with ideas proportionate to the research carried-out in the previous scientific work. Software defined capacity management which is liable for organizing tasks, observing, reporting, and resource maintenance is implemented. This new panel considers the resource off Fog control nodes managerial responsibility. Along with the divergent concept of architecture, key aspects of application, and directions & enablers for research are outlined.

The schemes mentioned for the applications are healthcare, virtual reality, preprocessing and caching. A fall auditing system in healthcare for inmate with strokes is the key scenario. Augmented reality is used using Google Glasses and brain engagement games [22] in conjunction with cognitive help. The last application scenario emphasizes data preprocessing and caching. The centered ultimatums are in modeling techniques, stability, resource allocation and energy reduction.

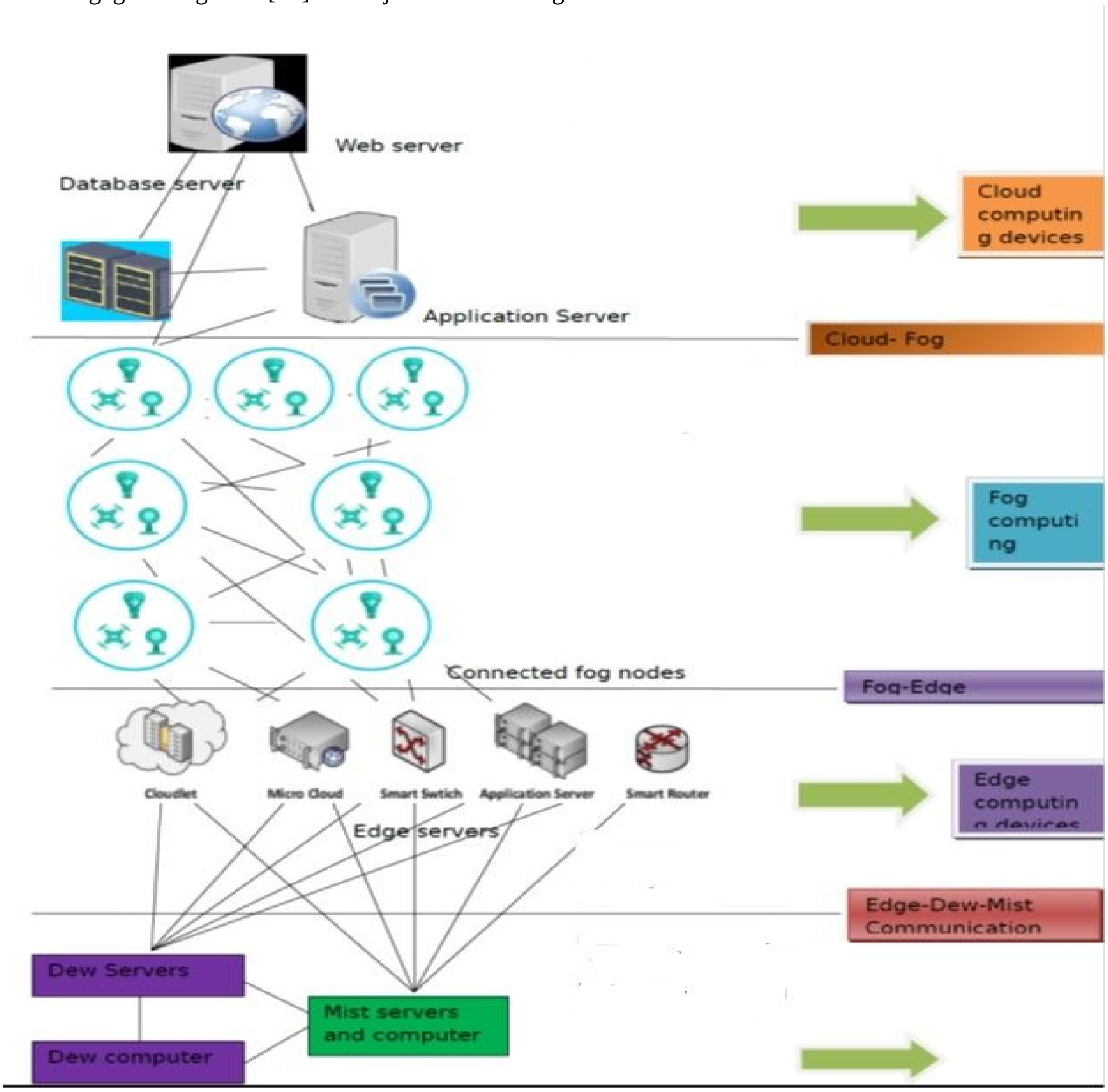


Figure 1.1: Phases of Interaction of Computing from Cloud Computing to Dew & Mist Computing

Sarkar et al.[9] introduce in this paper a detailed theoretical model of a computing environment for fog. Fog computing for IoT applications is presented as a supporting technology. First, they introduce three-tier architecture. Tier 1 stands for cluster-formulated IoT devices. Edge portal, fog-space portal, and fog instances are in Tier 2 followed by Tier 3 with cloud. The role, size, and position of the IoT devices are considered and well explained. Sarkar et al. use two efficiency metrics to measure the model's performance: system latency, and power usage. The execution assessment is examined on a given scheme and demonstrates substantial changes as opposed to a cloud computing approach.

2.2 Fog Computing Programming Models

The related work on the programming models for fog computing is examined in this part. As far as a programming model is concerned, it is essential to check suitable appliance, as not every appliance form suits in a fog landscape. The appliances are investigated based on distributed architecture & need to be detached into different standalone micro-services [23].A high level programming model is provided in Hong et al. [8], including APIs & simple resource allocation study based on novel fog resource criteria [24], [25]. This model of Mobile Fog, is based on a PaaS approach, which is hosted by a cloud service provider & implemented by developers, check this application, and deploy solutions.

During runtime, the model presented takes liability for dynamic escalation of executed appliances. The typical objectives of the work underway are the development of a HLL programming plan [26] and promoting dynamic scalability [27]. Mobile fog appliance is developed on processes configured to distribute computing examples. Such examples are clustered in chunks about level of position & protocol of the network [28]. Any mobile fog cycle [29] deals with demands of area committed. We can create this process by utilizing an image, storing it in a database; add a unique id to recognize the process perfectly.

The segmentation of ambiance is executed by developing instances of on-demand computing. This leveling method is dependent on tracking measures. Spatio-temporal repository [30] involves holding a stockpile container, and handling knowledge about territory and date. Atlast, a simulation of framework is done using developed traffic conditions & vehicular traffic [31] in given area of traffic. Simulated results indicate model enhancement as opposed to a general approach to cloud computing.

2.3 Resource Provisioning in Fog Computing

An optimization in Skarlat et al.'s work [10], along with the system architecture Resource provisioning problem is introduced in a fog landscape. The paper describes four different tools in this architecture-(i) cost and schedule-poor IoT tools, such as sensors and actuators, which submit requests for tasks to higher hierarchies.(ii)Rich-in-resource Internet of Things gadgets, i.e. fog units, for handling task demands got from IoT associated devices.(iii)Fog control hubs to coordinate fog units and deal with the asset gracefully and planning of the assignments in related fog clusters. (iv)To maintain the communication between the control monitor nodes an cloud service providers, a middleware known as Cloud-fog is used for the fog orchestration. The colonies of the fog contain fog units, IoT gadgets and fog orchestration. The fog landscape is simulated in their work by extension of the simulation tool CloudSim3. The assessment contrasts CloudSim's default provisioning policies with the enhanced method and contrasts the entire fog environment execution of tasks with execution of a cloud showing the benefits of fog computing.

A three layer fog computing architecture is proposed by the authors Aazam and Huh [1], [2], in which it consists of cloud, fog landscape, and IoT gadgets with various control hubs in those layers that can arrange the fog cells inside them. The authors embrace an approach Micro DCs orchestrated through smart gates. A detailed theoretical tool [43], in addition it includes management model. The forecast of future requests for assets is based on for sorts of gadgets available, surrender the probabilities created by the recorded information get to, estimating models and kind of service. The draw back in this model is that it was not able to react to dynamic changes in the landscape. The author also presented in their future work an enhancing of the hypothetical asset [45] arranging model as far as usage determination and QoS in the feeling of advanced media IoT devices [3].

3. METHODOLOGY

The proposed model of layered architecture of fog in figure 1.2 provides the framework for building a programming model for the connectivity of IoT devices which are less in computational speed and processing of data at the low level layer. These IoT devices utilize very low amount of resources for managing their tasks. Fog cells and fog control nodes are part of fog computing layer managing the tasks request from IoT devices and performing orchestration to handle the resource management and task management . The cloud fog control middleware provides the connectivity of these fog orchestration control nodes with cloud hosting layer which hosts large amount of resources for storing data in data centers and increase the power of availability of processing and throughput by enabling VM placement, task execution and sending the results of decision back to IoT devices.

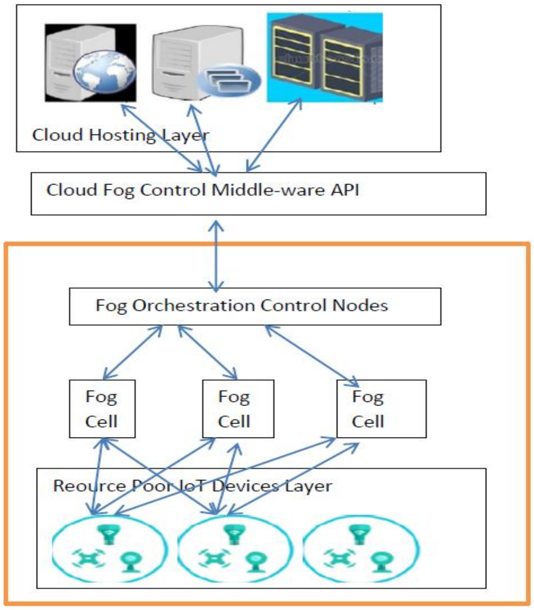

Cloud Hosting Layer
Cloud Fog Control Middle-ware API
Fog Orchestration Control Nodes
Fog Cell
Fog Cell
Fog Cell
Reource Poor IoT Devices Layer

Figure 1.2 Layered Architecture of Fog

4. IMPLEMENTATION AND EVALUATION

The Layered Architecture of fog is simulated through YAFS (yet another Fog Simulator) using the routetopolgy.json file where the end device represents the Fog devices the Gateway device will acts a Fog orchestration controller node and connecting to cloud for getting resource provisioned are shown as in figure 1.3

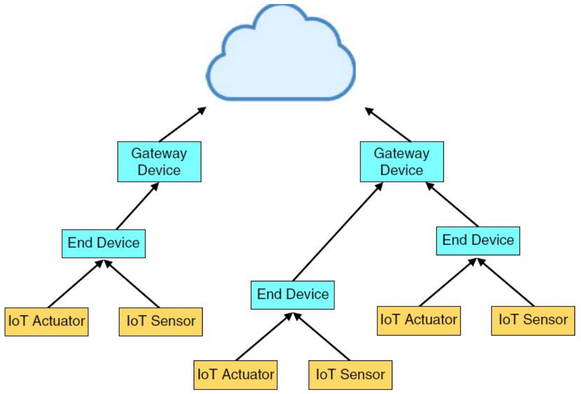
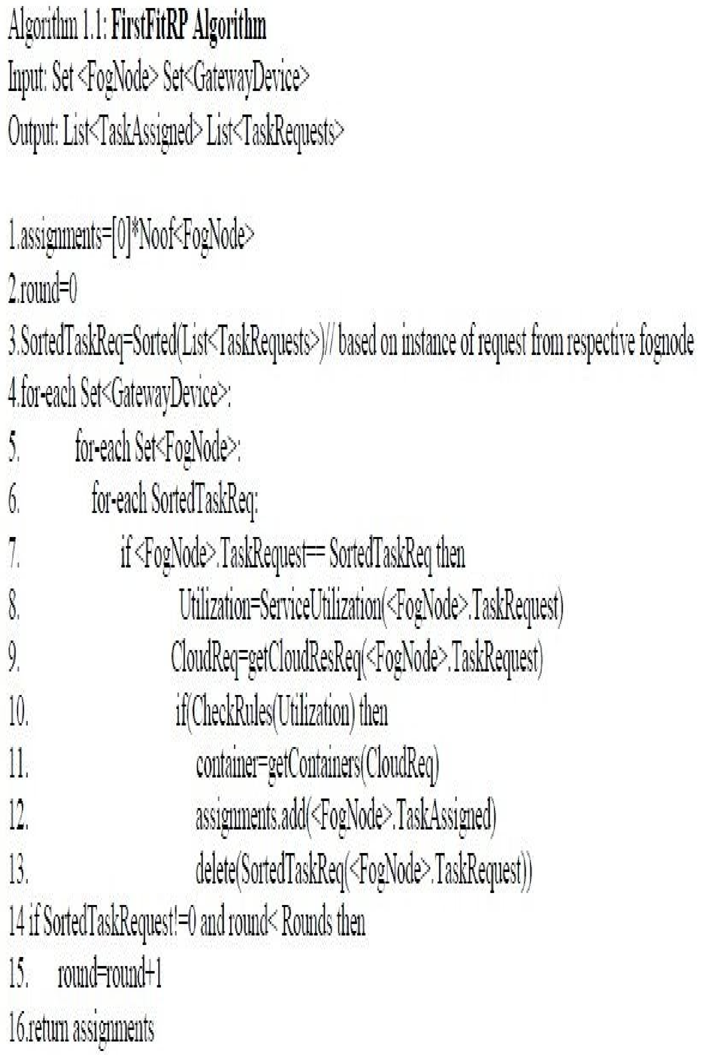


Figure 1.3 Layered Implementation model of Fog using YAFS.



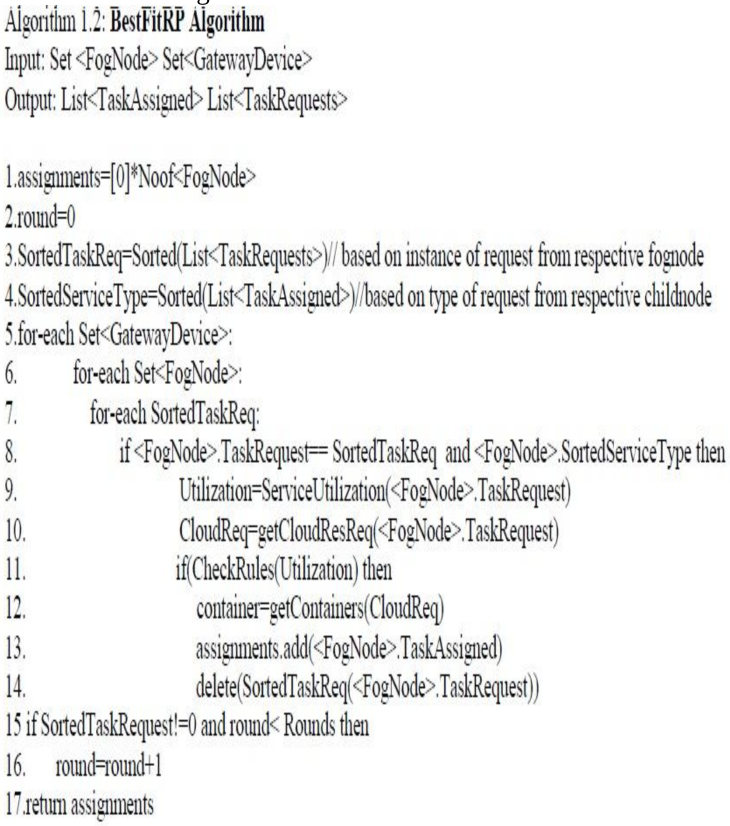
```
Algorithm 1.1: FirstFitRP Algorithm
Input: Set <FogNode> Set<GatewayDevice>
Output: List<TaskAssigned> List<TaskRequests>

assignments=[0]*Noof<FogNode>
round=0
SortedTaskReq=Sorted(List<TaskRequests>)// based on instance of request from respective fognode
for-each Set<GatewayDevice>:
        for-each Set<FogNode>:
            for-each SortedTaskReq:
                if <FogNode>.TaskRequest== SortedTaskReq then
                        Utilization=ServiceUtilization(<FogNode>.TaskRequest)
                        CloudReq=getCloudResReq(<FogNode>.TaskRequest)
                        if(CheckRules(Utilization) then
                            container=getContainers(CloudReq)
                            assignments.add(<FogNode>.TaskAssigned)
                            delete(SortedTaskReq(<FogNode>.TaskRequest))
if SortedTaskRequest!=0 and round< Rounds then
        round=round+1
return assignments
```

Figure 1.4 First Fit Resource Provisioning Algorithm

The figure 1.4 provides the first fit resource provisioning algorithm where the instance of task requests made by the fog node devices are collected at gateway node and processed based on the first arrival request for assigning the task to the cloud. The figure 1.5 provides the details about best fit resource provisioning depending on the priority of the service request they are allocated those tasks which need to be get serviced with high priority is allotted. The table 1.1 gives the results of execution of requests for both first fit and best fit resource provisioning algorithm [44]. The figure 1.6 highlights the results which show that the execution time for Best fit is higher as it needs to check for each tasks request and check the service type of assigning cloud services. The table 1.2 provides the details of throughput of the both best fit and first fit algorithms. The figure 1.7 highlights the results of the no of jobs handled by the algorithms.

Best Fit and First Fit algorithms are part of API of Dynamic workload balancing in YAFS



```
Algorithm 1.2: BestFitRP Algorithm
Input: Set <FogNode> Set<GatewayDevice>
Output: List<TaskAssigned> List<TaskRequests>

assignments=[0]*Noof<FogNode>
round=0
SortedTaskReq=Sorted(List<TaskRequests>)// based on instance of request from respective fognode
SortedServiceType=Sorted(List<TaskAssigned>)//based on type of request from respective childnode
for-each Set<GatewayDevice>:
        for-each Set<FogNode>:
            for-each SortedTaskReq:
                if <FogNode>.TaskRequest== SortedTaskReq  and <FogNode>.SortedServiceType then
                        Utilization=ServiceUtilization(<FogNode>.TaskRequest)
                        CloudReq=getCloudResReq(<FogNode>.TaskRequest)
                        if(CheckRules(Utilization) then
                            container=getContainers(CloudReq)
                            assignments.add(<FogNode>.TaskAssigned)
                            delete(SortedTaskReq(<FogNode>.TaskRequest))
if SortedTaskRequest!=0 and round< Rounds then
        round=round+1
return assignments
```

Figure 1.5 Best Fit Resource Provisioning Algorithm

Table1.1: Results of Execution of Requests of both BestFit and FirstFit Resource Provisioning Algorithms

Simulation Time	Execution of Request (First Fit)(msec)	Execution of Request (Best Fit)(msec)
200	23	35
300	45	54
400	63	78
500	89	92
600	120	130

Table 1.2: Results of Throughput measured for both BestFit and FirstFit Resource Provisioning Algorithms

Simulation Time	Through Put (First Fit)	Through Put (Best Fit)
200	2	4
300	5	8
400	11	15
500	18	20
600	22	23

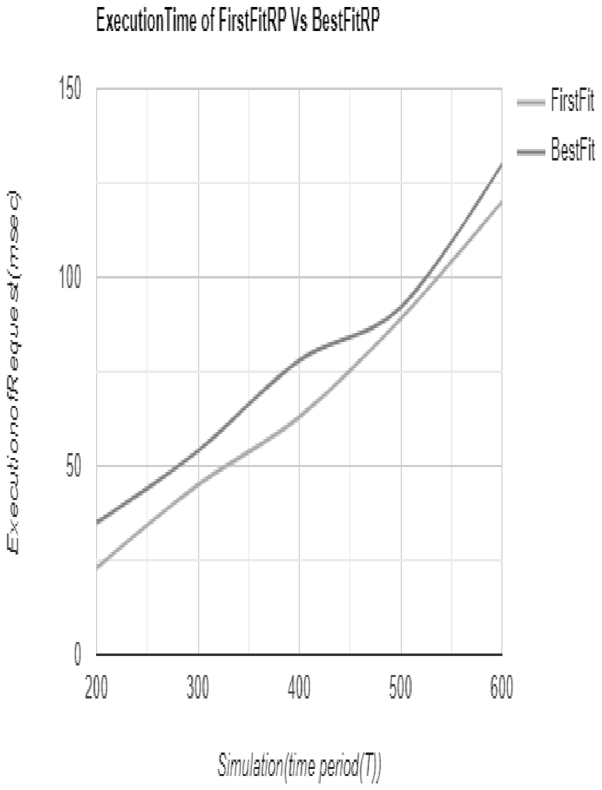


Figure 1.6: Execution of Requests of BestFit Vs FirstFit Resource Provisioning Algorithms.

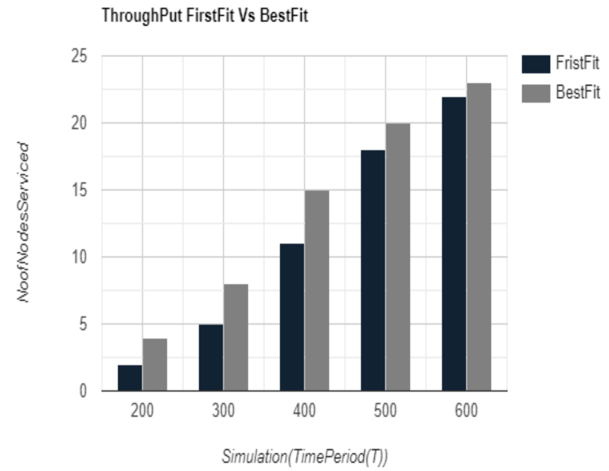


Figure 1.7: Throughput of BestFit Vs FirstFit Resource Provisioning Algorithms.

5. CONCLUSION AND FUTURE WORK

An extensive study of different design points of view in fog figuring is given. These design points of view integrally expand on physical just as intelligent segments and modules in a haze processing condition what's more, their particular jobs and functionalities. A programming model has a system wide view and some attention to hub level design. Related calculations and advancements should be created in a steady and vigorous way to understand the full favorable position of fog registering. The proposed model are implemented in YAFS by IoT devices and build middle ware framework to perform orchestration among fog nodes to service their communication with the cloud .The after effects of this study can help the exploration network to tune their endeavors towards the current holes to help push this innovation to development.

REFERENCES

1. Mohammad Aazam and Eui-Nam Huh. **Dynamic Resource Provisioning throughFog Micro Datacenter**. In Pervasive Computing and Communication Workshops,PerCom’15, Sydney, NSW, Australia, March 2015. IEEE ,pp. 105-110.
2. Mohammad Aazam and Eui-Nam Huh. **Fog Computing Micro Datacenter BasedDynamic Resource Estimation and Pricing Model for IoT**. In 29th IEEE InternationalConference on Advanced Information Networking and Applications, AINA’15,. IEEE, March 2015, pp. 687–694.
3. Mohammad Aazam, Marc St-Hilaire, Chung-Horng Lung, and IoannisLambadaris.**MeFoRE: QoE based Resource Estimation at Fog to enhance QoS in IoT**. In23rd IEEE International Conference on Telecommunications, ICT’16, Thessalonica, Greece, May 2016. IEEE, pp.1-5.

4. FlavioBonomi, Rodolfo Milito, PreethiNatarajan, and Jiang Zhu. **Fog Computing:A Platform for Internet of Things and Analytics**, volume 546 of Studies in ComputationalIntelligence, pages 169–186. Springer International Publishing, Cham, Switzerland, March 2014, pp.169-186.
5. FlavioBonomi, Rodolfo Milito, Jiang Zhu, and SateeshAddepalli**. Fog Computingand Its Role in the Internet of Things**. In 1st ACM SIGCOMM Workshop on MobileCloud Computing, SIGCOMM'12,Helsinki, Finland, August 2012.ACM, pp.13-16.
6. Amir VahidDastjerdi and RajkumarBuyya. **Fog Computing: Helping the Internetof Things Realize Its Potential.** Computer, 49(8) August 2016. pp. 112-116.
7. Amir VahidDastjerdi, Harshit Gupta, Rodrigo N. Calheiros, Soumya K. Ghosh,and RajkumarBuyya. **Fog Computing: Principles, Architectures, and Applications,** Elsevier, Burlington, MA, USA, January 2016 , pp. 61-75 ch. 4.
8. Kirak Hong, David Lillethun, UmakishoreRamachandran, BeateOttenwälder, andBoris Koldehofe. **Mobile Fog: A Programming Model for Large-Scale Applicationson the Internet of Things.** In 2nd ACM SIGCOMM Workshop on Mobile CloudComputing, SIGCOMM'13, Hong Kong, China, August 2013. ACM, pp.15-20
9. SubhadeepSarkar and SudipMisra. **Theoretical Modelling of Fog Computing: agreen Computing Paradigm to Support IoT Applications**. IET Networks, 5(2),March 2016, pp.23-29.
10. OlenaSkarlat, Stefan Schulte, Michael Borkowski, and Philipp Leitner. **Resource Provisioning for IoT Services in the Fog**. In 9th IEEE International Conference on Service Oriented Computing and Applications, SOCA'16, Macau, China, November 2016. IEEE, pp.32-39
11. Luis M. Vaquero and Luis Rodero-Merino. Finding Your Way in the **Fog: Towardsa Comprehensive Definition of Fog Computing.** ACM SIGCOMM ComputerCommunication Review, 44(5), October 2014, pp.27-32.
12. Jiang Zhu, Douglas S. Chan, Mythili S. Prabhu, PreethiNatarajan, Hao Hu, and FlavioBonomi. **Improving Web Sites Performance Using Edge Servers in FogComputing Architecture**.In 2013 IEEE 7th International Symposium on, SOSE'13, IEEE, March 2013, pp.320-323.
13. S. Yi, C. Li, and Q. Li, **A survey of fog computing: Concepts, applications and issues**,in Proc. Workshop Mobile Big Data, New York, NY,USA, 2015, pp. 37-42.
14. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, **Fog computing and its role in the Internet of Things,**in Proc. 1st Ed. MCC Workshop MobileCloudComput., New York, NY, USA, 2012, pp. 13-16.
15. S. Yangui, P. Ravindran, O. Bibani, R. H. Glitho, N. Ben Hadj-Alouane,M. J. Morrow, and P. A. Polakos, **A platform as-a-service for hybridcloud/fog environments,** in Proc. IEEE Int. Symp. Local Metrop. AreaNetw. (LANMAN), Jun. 2016, pp. 1-7.
16. M. Aazam and E.-N. Huh, **Dynamic resource provisioning through fogmicro datacenter,** in Proc. IEEE Int. Conf. Pervas. Comput. Commun.Workshops (PerCom Workshops), Mar. 2015, pp. 105-110.
17. O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, **Edge computing enabling the Internet of Things**, in Proc. IEEE 2ndWorld Forum Internet
18. Kodali, S., Dabbiru, M., Rao, B.T., **Q-HeteLearn: A progressive learning approach for classifying meta-paths in heterogeneous information networks**,2020,International Journal of Emerging Trends in Engineering Research,
19. Sreelatha, K., Krishna Reddy, V ,**A comprehensive review of security challenges for data deduplication and integrity auditing**, , 2019, International Journal of Emerging Trends in Engineering Research, 7(11),52, pp.725-732
20. Dey, N.S., Gunasekhar, T, **Comprehensive Survey of Load Balancing Strategies Using Hadoop Queue Scheduling and Virtual Machine Migration**,.2019,IEEE,7,8762113,pp.2259-92284
21. Prakash, K.B., Nagapawan, Y.V.R., Kalyani, N.L., Kumar, V.P , **Chatterbot implementation using transfer learning and LSTM encoder-decoder architecture**,.2020,International Journal of Emerging Trends in Engineering Research, 8(5),pp.1709-1715.
22. Dharmapuri, S.L., Dandamudi, P.K., Botcha, V.M., Kolla, B.P, **Detecting central nervous system disorder using machine learning technique (XGB classifier)**, 2020, International Journal of Emerging Trends in Engineering Research,8(4), pp. 1142-1147
23. Prakash, K.B., Imambi, S.S., Ismail, M., Pavan Kumar, T., Naga Pawan, Y.V.R ,Analysis**, prediction and evaluation of covid-19 datasets using machine learning algorithms**,.2020, International Journal of Emerging Trends in Engineering Research, 8(5),117,pp.2199-2204
24. Mahmud, R., Toosi, A.N., Ramamohanarao, K., Buyya, R ,**Context-Aware Placement of Industry 4.0 Applications in Fog Computing Environments**,2020, IEEE Transactions on Industrial Informatics 16(11),8894519,pp.7004-7013.
25. Ranjan, R., Villari, M., Shen, H., Rana, O., Buyya, R **,Software tools and techniques for fog and edge computing**,.2020, Software - Practice and Experience,50(5), pp.473-475.

26. Mukherjee, A., Deb, P., De, D., Buyya, R ,IoT-F2N: **An energy-efficient architectural model for IoT using Femtolet-based fog network**,.2019, Journal of Supercomputing, 75(11),pp.7125-7146.
27. Sinha Roy, D., Behera, R.K., Reddy, K.H.K., Buyya, R., **A context-aware fog enabled scheme for real-time cross-vertical IoT applications**, 2019, IEEE Internet of Things Journal, 6(2),8457200, pp. 2400-2412
28. Mahmud, R., Ramamohanarao, K., Buyya, R ,**Latency-aware application module management for fog computing environments**, 2018, ACM Transactions on Internet Technology, 19(1),9
29. Babitha, D., Ismail, M., Chowdhury, S., Govindaraj, R., & Prakash, K. B. (2020). **Automated road safety surveillance system using hybrid cnn-lstm approach.** International Journal of Advanced Trends in Computer Science and Engineering, 9(2), pp.1767-1773.
30. Babitha, D., Jayasankar, T., Sriram, V. P., Sudhakar, S., & Prakash, K. B. (2020). **Speech emotion recognition using state-of-art learning algorithms**. International Journal of Advanced Trends in Computer Science and Engineering, 9(2), pp.1340-1345.
31. Bharadwaj, Y. S. S., Rajaram, P., Sriram, V. P., Sudhakar, S., & Prakash, K. B. (2020). **Effective handwritten digit recognition using deep convolution** neural network. International Journal of Advanced Trends in Computer Science and Engineering, 9(2), pp.1335-1339.
32. Prakash, K.B., Rangaswamy, M.D. and Ananthan, T.V., 2014. **Feature extraction studies in a heterogeneous web world**. International Journal of Applied Engineering Research, *9*(22), pp.16571-16579.
33. Ismail, M., Prakash, K.B. and Rao, M.N., 2018. **Collaborative filtering-based recommendation of online social voting**. International Journal of Engineering and Technology *(UAE)*, *7*(3), pp.1504-1507.
34. Prakash, K.B. and Rajaraman, A., 2016. **Mining of bilingual Indian Web documents. Procedia Computer Science**, 89, pp.514-520.
35. Prakash, K.B., Rangaswamy, M.D., Ananthan, T.V. and Rajavarman, V.N., 2015. **Information extraction in unstructured multilingual web documents**. Indian Journal of Science and Technology, *8*(16), pp.1-8.
36. Prakash, K.B., Rangaswamy, M.D. and Raman, A.R., 2012, November. **ANN for multi-lingual regional web communication**. In *International Conference on Neural Information Processing* ,Springer, Berlin, Heidelberg, pp. 473-478.
37. Prakash, K.B., Rangaswamy, M.D. and Raman, A.R., 2012, August. **Statistical interpretation for mining hybrid regional web documents**. In *International Conference on Information Processing* Springer, Berlin, Heidelberg, pp. 503-512.
38. Prakash, K.B., 2016, July. **Content extraction studies using total distance algorithm**. In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)* IEEE, pp. 673-679.
39. Prakash, K.B., 2015. **Mining issues in traditional Indian web documents**. *Indian Journal of Science and Technology*, *8*(32), pp.1-11.
40. Prakash, K.B., RajaRaman, A. and Lakshmi, M., 2017, March. **Complexities in developing multilingual on-line courses in the Indian context**. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)* IEEE, pp. 339-342.
41. Prakash, K.B., Kumar, K.S. and Rao, S.U.M., 2016, July. **Content extraction issues in online web education.** In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)* IEEE, pp. 680-685.
42. Prakash, K.B., RajaRaman, A., Perumal, T. and Kolla, P.,**Foundations to frontiers of big data analytics**. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, IEEE, pp. 242-247.
43. Prakash, K., Lakshmi Kalyani, N., Vadla, P.K. and Naga Pawan, Y.V.R., 2020. **Analysis of mammography for identifying cancer cells using convolution neural networks.** *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(2), pp.1184-1188.
44. Vadla, P.K. and Prakash, K.B., 2020. **Residue based adaptive resource provisioning through multi-criteria decision and horizontal scaling of vm's in agent-based model for federated cloud.** *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(2), pp.1610-1622.
45. Kavuri, M., & Prakash, K. B. (2019). **Performance comparison of detection, recognition and tracking rates of the different algorithms**. International Journal of Advanced Computer Science and Applications, 10(6), pp.153-158