iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** IX **Month of publication:** September 2019

DOI: http://doi.org/10.22214/ijraset.2019.9015

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue IX, Sep 2019- Available at www.ijraset.com

Implementation of Critical Path Algorithm for Resource Management in Fog Environment

Antim Malik[1], Vikas Malik[2]
[1, 2]*Mtech Cse, BPS Mahila Vishwavidyalaya, Khanpur Kalan Gohana.*

Abstract: ***Research is considering the concept of resource scheduling in fog environment along with need of Fog computing has been discussed here. The research work focuses on the scheduling algorithm for resource management. The concept of node duplication has been discussed in critical path algorithm. Such algorithms are suppose to minimize the make span time. These algorithms are performing efficiently to manage cloud resources. The Critical Path Algorithm has been used for Real Time Applications in Cloud Environment. In order to show the working algorithm. The operation has been made on the sample graph. The graph involves the six jobs from t1 to t6, and two dummy tasks, tst and tex. Three services are there for every job which executes the task with different QoS. The technical implementation of critical path using java is made in this paper that has been integrated in fog environment. In the future the proposed algorithm is applied to improve the performance of fog environment.***
Keywords: Cloud computing, Fog Computing, Scheduling, Resource scheduling, Critical path algorithm.

I. INTRODUCTION

Using cloud services [2], organizations are capable to deploy their software systems on pool of resources. These organizations depend on their business-critical systems. Their systems are developed over long periods. Such legacy applications are implemented on-premise. Several researches in cloud migration have been carried out in last few decades.

Requirement of storage of data [3] is enhancing every day.

It may be considered as a record or as the memory. Under the traditional [22] way of storing hard disks were used in computers or in the smart phones. Data store is increasing simultaneously with the increase in profiles number of individuals and there was a parallel increase in the store of data. There is elasticity, scalability, efficiency and multi-mobility in cloud computing. Fog computing is an expansion of cloud computing services up to the border of the network. The purpose is to minimize latency and overcrowding of network.

It is comparatively a trend of the present research. Even though same type of resources and services are offered by cloud and fog, the latter is categorized by low latency with a broader spread and geographically circulated nodes for supporting movement and interaction of real-time.The extension of cloud computing has become possible due to Fog computing. Cloud computing offers data, computation, storage, and application services for the end-user.

Fog computing is that dispersed computing networking device which is omnipresent. It is often considered as IoT devices and some new devices and analytics.

It is considered to be happening in Fog infrastructures because of mobility of various applications, allocation of resources and distribution of management .This is done by a fog layer in which present user movement is forwarded to geographically distributed data centers of cloud. More than this, fog layer time can handle the application execution requests distribution of IoT devices at the network edges. Here data generation and processing is not taken into consideration.

II. OBJECTIVE

The objectives of research have been discussed as follow:

A. To study the need and limitations of existing Fog based system.
B. To investigate different scheduling techniques.
C. To review the existing researches related to fog computing along with their achievements and limitations.
D. To consider the concept of resource scheduling in fog based system.
E. To proposed a fog system associated with resource scheduling in order to perform comparative analyses of proposed and traditional system.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue IX, Sep 2019- Available at www.ijraset.com

III. PROBLEM FORMULATION

There are many problems that are faced during fog computing. It has been observed that applications running on devices need centralized point to host application. There are limited computation and storage capacity. However present trends are resolving such issues. These mechanisms have tried to provide more energy-efficient. Such mechanisms based devices are the powerful and efficient devices. There are some improvements that have been approved for non-user appliances. The management of resources on fog environment is a big issue. Having potentially billions of small appliances are arranged, the fog depends on decentralized management mechanisms. Other side the lack of security is a big issue. Data privacy has become a problem. However several encryption and security mechanisms have been proposed. But these mechanisms have influenced the performance as lot of the time is wasted in securing data before transmission.

Standardization of the fog environment is another problem as the system is using different hardware from different vendors. Some time it becomes difficult to configure routers and other devices due to compatibility issues. There is need of task scheduling. It is capable to consider the balancing of workload over client device as well as computation servers. Resource management considers the placement of task images on storage servers. Balancing of input and output interrupt requests in case of storage servers may resolve issues related to cloud storage. The process of Task scheduling is required to balance the workload over client device as well as computation servers. There is need of management of resource as it considers the placement of task images on storage servers. Balancing of input and output interrupt requests in case of storage servers might be helpful in resolve issues related to resource management.

IV. RESULT AND DISCUSSION

The first critical path technique has been applied to manage the resources in this research. The real technique has been introduced for construction work. It has been applied for any task running on cloud where there are interdependent activities. During resource scheduling it would be easy to allot resources to task that are running on cloud, after getting status of their dependency. With critical path technique, the critical actions in a program have been classified. Such activities have an effect directly on the finishing time of operation on cloud.

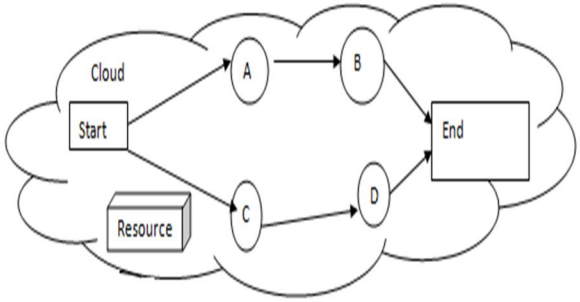


Fig 1Resource Management on Cloud

A. Proposed Work In Process Flow

The process flow of proposed work in resource scheduling phase has six steps.

1) *Step 1:* Specific the activity
2) *Step 2:* Establish the activity sequence
3) *Step 3:* Network diagram
4) *Step 4:* Every activity estimation
5) *Step 5:* Identify the critical path
6) *Step 6:* Use the critical path diagram to indicate the project progresses

B. The Way To Identify The Critical Path

To identify the critical path, four parameters of each task of the fog are considered.

1) Earliest start time (ES) - The earliest time and activity can start once the previous dependent actions.
2) Earliest finish time (EF) - ES + duration of activity.
3) Latest finish time (LF) - The latest time may end an activity lacking of delay of any type in the project.

Latest start time (LS) - LF - duration of activity

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue IX, Sep 2019- Available at www.ijraset.com

C. Critical Path Diagram Showing The Task Progresses

Critical path diagram is a live. Therefore, this diagram should be updated with actual values once the task is completed. This gives more realistic figure for the deadline and the fog management can know whether they are on track regarding the deliverables resources.

D. Critical Path Algorithm for Real Time Applications in Cloud Environment

1) In today's scenario cloud computing is gaining more and more popularity as we are moving towards digitization. With the evolution of cloud computing technique there are issues are faced to get better the service quality.
2) In cloud computing environment there are various parameters as performance, availability, reliability, on demand cost etc.
3) In this research proposed an algorithm that finds the availability of the services for the real time applications.
4) The algorithm would find the new critical path and optimal deadline for the task workflow model based on earliest beginning time and latest ending time.

E. Mathematical Formulation for Task Dependency

1) A workflow application W = (T,E) has been modeled in the form of directed acyclic graph (DAG) here T={T1; T2; . . . ; Tn} has been considered her the set of tasks. here E is the set of directed edges.
2) An edge Eij of the form (Ti; Tk) situates in the situation of data dependency between Ti and Tj, in the case if Ti is the parent task of Tj and Tj Tj and Tj, will be the child task of Ti.
3) As per the directed acyclic graph a child task will be completed after the completion of its entire parent tasks.
4) In addition, every flow of work W has a deadline connected to it. A deadline has been expressed as a time limit for the implementation of the workflow.
5) Precedence relation " < " defines if Ti < Tk ie when Ti is completed then Tk executes and Ti is immediate predecessor of Tk while Tk is immediate successor of Ti.
6) Tp represent parent node for any task node, Tc represent Child node for any task and D is the deadline for the workflow schedule.

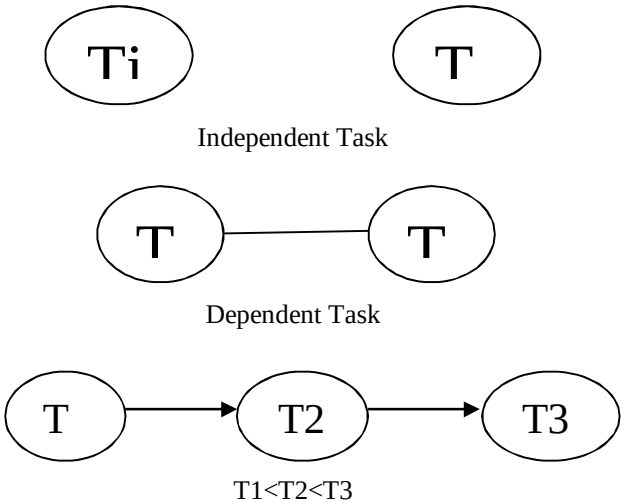


Fig 2 Mathematical Formulation for Task Dependency

F. Performance Analysis

1) To show the works process of algorithm, its operation has been drawn on the sample graph. The graph involves the six tasks. Such tasks are from t1 to t6, and it also includes two dummy tasks, tst and tex. Three separate services are there for each task. Such perform the task with several QoS.
2) One thing is clear that the cost of a faster service than a slow service. It has come to know from the analysis of every job, it has been supposed that all services are completely available at desired time.
3) For making the example very simle, it has been supposed that the calculated information sending time and cost between two adjacent jobs are fixed

Table 1Analysis of EST and LFT

Tasks	Initial State			After removing task T4			
		EST	LFT			EST	LFT
T1		0	0	T1		0	0
T2		0	0	T2		0	0
T3		0	0	T3		0	0
T4		3	3	T4		-	-
T5		3	4	T5		3	3
T6		6	6	T6		5	5

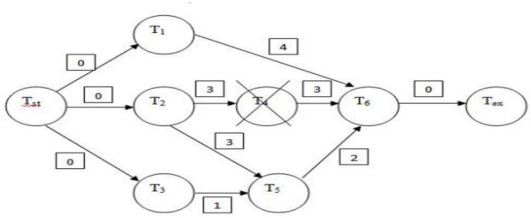


Fig 3 Work flow Schedule on Task Removal

G. In Resource Management Advantages of Critic

There are several advantages of critical path method which have been given below:

1) It provides a visual representation related to actions on cloud and fog.

2) It presents the time in order to complete the jobs and to complete the whole execution.

3) It helps to allot the resource.

V. IMPLEMENTATION OF CRITICAL PATH

Here in this section the java based code to implement critical path has been specified that has been integrated in fog environment in order to manage resources. The CPM depends on the task dependency.

A. Source code

```
import java.util.*;
public class CriticalPath {
public static void main(String[] args) {
    HashSet<Task> allTasks = new HashSet<Task>();
  Task end = new Task("End", 0);
  Task F = new Task("F", 2, end);
  Task A = new Task("A", 3, end);
  Task X = new Task("X", 4, F, A);
  Task Q = new Task("Q", 2, A, X);
  Task start = new Task("Start", 0, Q);
  allTasks.add(end);
  allTasks.add(F);
  allTasks.add(A);
  allTasks.add(X);
  allTasks.add(Q);
  allTasks.add(start);
  System.out.println("Critical Path: "+Arrays.toString(criticalPath(allTasks)));
 }

//A wrapper class to hold the tasks during the calculation
 public static class Task{
  //the actual cost of the task
  public int cost;
  //the cost of the task along the critical path
  public int criticalCost;
  //a name for the task for printing
  public String name;
  //the tasks on which this task is dependant
  public HashSet<Task> dependencies = new HashSet<Task>();
  public Task(String name, int cost, Task... dependencies) {
    this.name = name;
    this.cost = cost;
    for(Task t : dependencies){
```

```
                this.dependencies.add(t);
                }
                }
                @Override
                public String toString() {
                return name+": "+criticalCost;
                }
                public boolean isDependent(Task t){
                //is t a direct dependency?
                if(dependencies.contains(t)){
                return true;
                }
                //is t an indirect dependency


  for(Task dep : dependencies){
    if(dep.isDependent(t)){
      return true;
    }
  }
  return false;
 }
}

public static Task[] criticalPath(Set<Task> tasks){
 //tasks whose critical cost has been calculated
 HashSet<Task> completed = new HashSet<Task>();
 //tasks whose ciritcal cost needs to be calculated
 HashSet<Task> remaining = new HashSet<Task>(tasks);

 //Backflow algorithm
 //while there are tasks whose critical cost isn't calculated.
 while(!remaining.isEmpty()){
  boolean progress = false;

  //find a new task to calculate
  for(Iterator<Task> it = remaining.iterator();it.hasNext();){
   Task task = it.next();
   if(completed.containsAll(task.dependencies)){
    //all dependencies calculated, critical cost is max dependency
    //critical cost, plus our cost
    int critical = 0;
    for(Task t : task.dependencies){
     if(t.criticalCost > critical){
      critical = t.criticalCost;
     }
    }
    task.criticalCost = critical+task.cost;
    //set task as calculated an remove


    completed.add(task);
     it.remove();
     //note we are making progress
     progress = true;
    }
   }
   //If we haven't made any progress then a cycle must exist in
   //the graph and we wont be able to calculate the critical path
  if(!progress) throw new RuntimeException
  ("Cyclic dependency, algorithm stopped!");
  }

  //get the tasks
  Task[] ret = completed.toArray(new Task[0]);
  //create a priority list
  Arrays.sort(ret, new Comparator<Task>() {
```

```
@Override
public int compare(Task o1, Task o2) {
  //sort by cost
  int i= o2.criticalCost-o1.criticalCost;
  if(i != 0)return i;

       if(o1.isDependent(o2))return -1;
  if(o2.isDependent(o1))return 1;
  return 0;
 }
});

 return ret;
}
}
```

B. Output

```
C:\Java\jdk1.6.0_10\bin>javac CriticalPath.java

C:\Java\jdk1.6.0_10\bin>java CriticalPath
Critical Path: [Start: 9, Q: 9, X: 7, A: 3, F: 2, End: 0]

C:\Java\jdk1.6.0_10\bin>
```

VI. CONCLUSION

The real time applications over the cloud environment have a challenge of finding a new critical path so as to evaluate deadline for a workflow schedule. The proposed algorithm evaluate the optimized deadline with the new critical path for a given workflow schedule when a task or job is being removed from the schedule. For the real time applications rate monotonic schedule is used to set static priority for every task for best service for each task so as to perform the task before the deadline of the schedule. The presented algorithm can be used to improve the QoS performance of cloud environment. Each virtual machine task has different services suppose where availability of critical path for real time applications can improve the performance of cloud environment that would take a step ahead for further execution of execution time and cost. Furthermore, the algorithm can be modified to set priority of the task using different and feasible technique.

VII. SCOPE OF RESEARCH

In the future the proposed algorithm would be used to improve the performance of fog environment. Each virtual machine task has different services suppose where availability of critical path for real time applications can improve the performance of cloud environment that would take a step ahead for further execution of execution time and cost. Furthermore, algorithm can be modified to set priority of the task using different and feasible technique. It is found from the analysis for each task a faster service costs more than a slower one. But the proposed work would provide all services completely available on desired time.

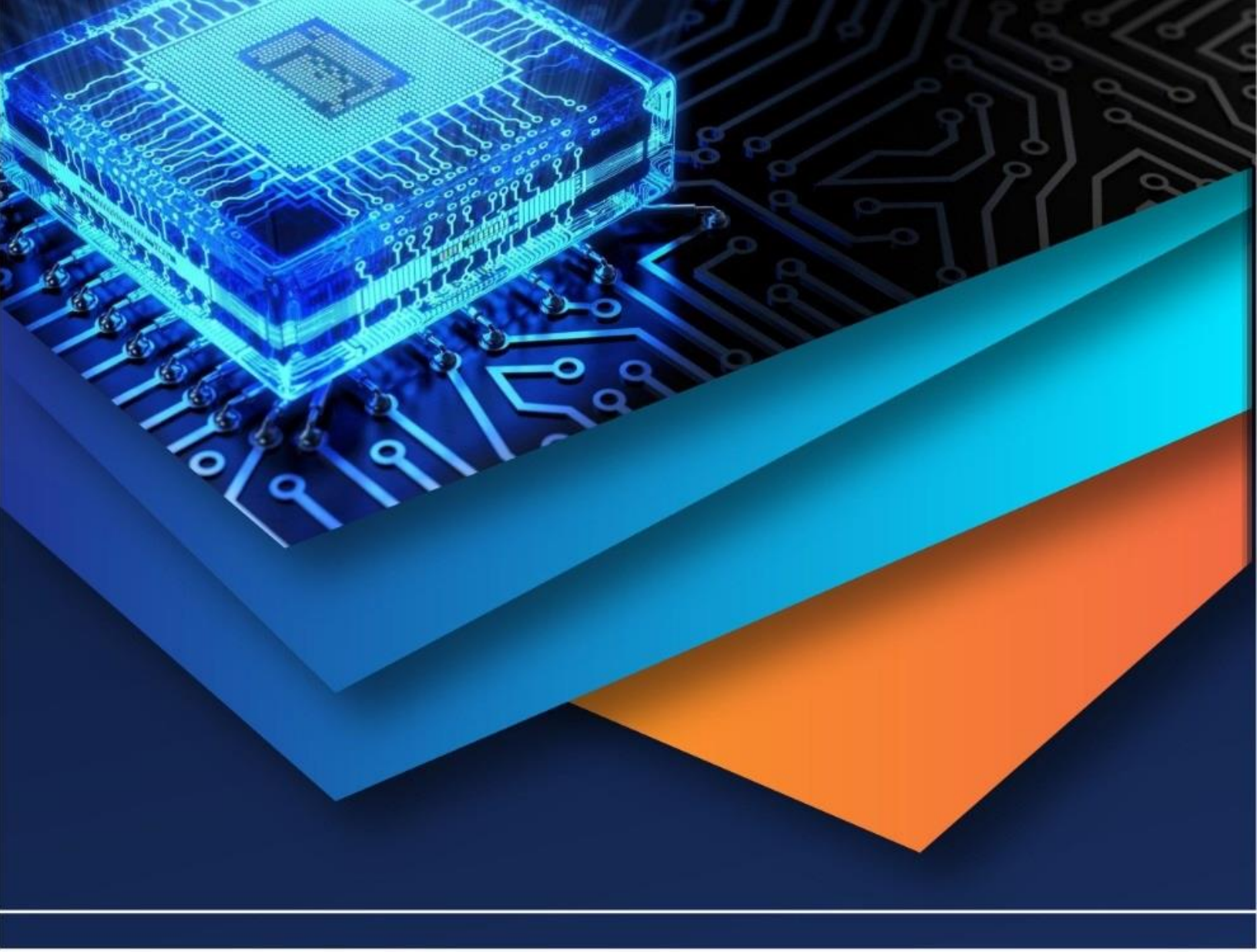
VIII. ACKNOWLEDGMENT

I would especially like to thank my worthy guide Guide name, whose supervision dissertation work has been carried out. Her technical advice, ideas, and consecutive criticism and motivation contributed to the success of this research. She suggested me many ideas and solved my puzzles when I was in need. Her motivation and help have been a great inspiration to me.
I would also like to thank our Vice-Chancellor, college name, Vice-Chancellor name for his continuous support and guidance from the institution.
I convey my thanks to respected Chairman Department of CSE, NAME OF CHAIRMAN and all the faculty members of CSE Department for providing me the opportunities, support and the necessary help to complete this project.
I would like to express a deep sense of gratitude and thanks profusely to our dissertation coordinator, Name of Assistant Professor Assistant Professor for his wise counsel and able guidance.
At last, I would like to express my gratitude towards my teammates, my friends, my family members who have directly or indirectly helped me in the completion of the dissertation work.

REFERENCES

[1] J. Deng, J. L. Hu, A. C. M. Liu, and J. Wu, "Research and application of cloud storage," Proc. - 2010 2nd Int. Work. Intell. Syst. Appl. ISA 2010, pp. 2–6, 2010.
[2] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud Migration Research: A Systematic Review," IEEE Trans. Cloud Comput., vol. 1, no. 2, pp. 142–157, 2013.
[3] S. Sharma and A. Chugh, "Survey Paper on Cloud Storage," vol. 1, no. 2, pp. 208–213, 2013.
[4] A. O. Joseph, J. W. Kathrine, and R. Vijayan, "Cloud security mechanisms for data protection: A survey," Int. J. Multimed. Ubiquitous Eng., vol. 9, no. 9, pp. 81–90, 2014.
[5] N. S. Dhande, "Fog Computing : Review of Privacy and Security Issues," vol. 3, no. 2, pp. 864–868, 2015.
[6] A. A. Nasr, N. A. El-Bahnasawy, and A. El-Sayed, "Performance Enhancement of Scheduling Algorithm in Heterogeneous Distributed Computing Systems," Int. J. Adv. Comput. Sci. Appl., vol. 6, no. 5, pp. 88–96, 2015.
[7] K. P. Saharan and A. Kumar, "Fog in Comparison to Cloud: A Survey," Int. J. Comput. Appl., vol. 122, no. 3, pp. 975–8887, 2015.
[8] S. Agarwal, S. Yadav, and A. K. Yadav, "An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing," Int. J. Inf. Eng. Electron. Bus., vol. 8, no. 1, pp. 48–61, 2016.
[9] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," IEEE Internet Things J., vol. 3, no. 6, pp. 854–864, 2016.
[10] S. N. K. Dr., "A Review-Fog Computing and Its Role in the Internet of Things," Proc. first Ed. MCC Work. Mob. cloud Comput., no. November, pp. 13–16, 2016.
[11] N. K. Giang, V. C. M. Leung, and R. Lea, "On Developing Smart Transportation Applications in Fog Computing Paradigm," Proc. 6th ACM Symp. Dev. Anal. Intell. Veh. Networks Appl. - DIVANet '16, pp. 91–98, 2016.
[12] M. Niranjanamurthy, P. B. Kavitha, P. Kasana, and S. N. Vishnu, "Research Study on Fog Computing for Secure Data Security," Int. J. Sci. Technol. Manag., vol. 5, no. 1, pp. 221–228, 2016.
[13] F. Y. Okay and S. Ozdemir, "A fog computing based smart grid model," 2016 Int. Symp. Networks, Comput. Commun. ISNCC 2016, 2016.
[14] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint Optimization of Task Scheduling and Image Placement in Fog Computing Supported Software-Defined Embedded System," IEEE Trans. Comput., vol. 65, no. 12, pp. 3702–3712, 2016.
[15] S. Chakraborty, S. Bhowmick, P. Talaga, and D. P. Agrawal, "Fog Networks in Healthcare Application," Proc. - 2016 IEEE 13th Int. Conf. Mob. Ad Hoc Sens. Syst. MASS 2016, pp. 386–387, 2017.
[16] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," GIoTS 2017 - Glob. Internet Things Summit, Proc., 2017.
[17] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: a review of current applications and security solutions," J. Cloud Comput., vol. 6, no. 1, 2017.
[18] M. M. Lopes, M. A. M. Capretz, and L. F. Bittencourt, "MyiFogSim : A Simulator for Virtual Machine Migration in Fog Computing," Proc. 10th Int. Conf. Util. Cloud Comput., no. Vm, pp. 47–52, 2017.
[19] H. A. M. Name, F. O. Oladipo, and E. Ariwa, "User mobility and resource scheduling and management in fog computing to support IoT devices," 7th Int. Conf. Innov. Comput. Technol. INTECH 2017, pp. 191–196, 2017.
[20] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K. K. R. Choo, and M. Dlodlo, "From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework," IEEE Access, vol. 5, no. c, pp. 8284–8300, 2017.
[21] T. Pandikumar and T. Belissa, "Distributed Denial of Service (DDOS) Attack Detection in Software Defined Networking with Cloud Computing," Int. J. Eng. Sci. Comput., vol. 7, no. 6, pp. 12685–12690, 2017
[22] S. Waghmare, S. Ahire, H. Fegade, and P. Darekar, "Securing Cloud using Fog Computing with Hadoop Framework," vol. 5, no. 3, pp. 34–38, 2017.
[23] V. P. Lalitha, M. Y. Sagar, S. Sharanappa, S. Hanji, and R. Swarup, "Data security in cloud," 2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017, vol. 2, no. 4, pp. 3604–3608, 2018.
[24] L. Ni, J. Zhang, and J. Yu, "Priced timed petri nets based resource allocation strategy for fog computing," Proc. - 2016 Int. Conf. Identification, Inf. Knowl. Internet Things, IIKI 2016, vol. 2018–Janua, no. c, pp. 39–44, 2018.
[25] T. Islam and M. M. A. Hashem, "Task Scheduling for Big Data Management in Fog Infrastructure," no. Cdc, pp. 1–6.2018
[26] X. Pham, "Towards task scheduling in a cloud-fog computing system," 2016.
[27] O. X. Rj, X. Dwlrq, and R. I. Uhvrxufhv, "' DDoS attack mitigation and Resource provisioning in cloud using Fog computing," pp. 308–313, 2017
[28] H. A. M. Name, F. O. Oladipo, and E. Ariwa, "User mobility and resource scheduling and management in fog computing to support IoT devices," 7th Int. Conf. Innov. Comput. Technol. INTECH 2017, pp. 191–196, 2017
[29] D. Rahbari, S. Kabirzadeh, and M. Nickray, "A Security Aware Scheduling in Fog Computing by Hyper Heuristic Algorithm," pp. 87–92, 2017
[30] L. Yin, J. Luo, and H. Luo, "Tasks Scheduling and Resource Allocation in Fog Computing Based on Containers for Smart Manufacturing," IEEE Trans. Ind. Informatics, vol. PP, no. c, p. 1, 2018
[31] J. Akram and A. Rafi, "Efficient Resource Utilization in Cloud-Fog Environment Integrated with Smart Grids" 2018 Int. Conf. Front. Inf. Technol., pp. 188–193, 2018.
[32] L. Ni, J. Zhang, and J. Yu, "Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets ," Proc. - 2016 Int. Conf. Identification, Inf. Knowl. Internet Things, IIKI 2016, vol. 2018–Janua, no. c, pp. 39–44, 2018.
[33] X. Xu, Q. Liu, L. Qi, Y. Yuan, W. Dou, and A. X. Liu, "A Heuristic Virtual Machine Scheduling Method for Load Balancing in Fog-Cloud Computing"IEEE Int. Conf. High Perform. Smart Comput. IEEE Int. Conf. Intell. Data Secur., pp. 83–88, 2018.
[34] Thanh Dat Dang ; Doan Hoang "A data protection model for fog computing " 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), Year: 2017, Page s: 32 – 38
[35] Hua-Jun Hong ":From Cloud Computing to Fog Computing: Unleash the Power of Edge and End Devices "2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Year: 2017 Page s: 331 – 334
[36] Huigui Rong ; Jianfang Li ; Bingguo Chang ; Fei Long "A Crowd Sourcing Service Model for Optimizing User-Desired Storage ResourceScheduling" 2015 IEEE 17th International Conference on High Performance Computing and Communications, Year: 2015, Page s: 920 – 920

[37] Zhao Xiao-qiang ; He Zhi-e "The multi-objective water resources optimization scheduling based on chaos genetic algorithm" The 27th Chinese Control and Decision Conference (2015 CCDC), Year: 2015, Page s: 4500 – 450
[38] Xuan-Qui Pham ; Eui-Nam Huh" Towards task scheduling in a cloud-fog computing system" 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Year: 2016, Page s: 1 – 4
[39] Yung-Chiao Chen ; Yao-Chung Chang ; Chang-Hsu Chen ; Yu-Shan Lin ; Jiann-Liang Chen ; Yu-Yao Chang "Cloud-fog computing for information-centric Internet-of-Things applications " 2017 International Conference on Applied System Innovation (ICASI),Year: 2017, Page s: 637 – 64
[40] Zhen Liu ; Jiawei Zhang ; Yanan Li ; Lin Bai ; Yuefeng Ji" Joint jobs scheduling and lightpath provisioning in fog computing micro datacenter networks " IEEE/OSA Journal of Optical Communications and Networking, Year: 2018 , Volume: 10 , Issue: 7, Page s: 152 – 163
[41] Haruna Abdu Manis Name ; Francisca O. Oladipo ; Ezendu Ariwa" User mobility and resource scheduling and management in fog computing to support IoT devices "2017 Seventh International Conference on Innovative Computing Technology (INTECH), Year: 2017,Page s: 191 – 19
[42] Doan Hoang ; Thanh Dat Dang" FBRC: Optimization of task Scheduling in Fog-Based Region and Cloud " 2017 IEEE Trustcom/BigDataSE/ICESS, Year: 2017, Page s: 1109 – 1114
[43] Luiz F. Bittencourt ; Javier Diaz-Montes ; Rajkumar Buyya ; Omer F. Rana ; Manish Parashar "Mobility-Aware Application Scheduling in Fog Computing", IEEE Cloud Computing",Year: 2017 , Volume: 4 , Issue: 2, Page s: 26 – 35
[44] B. Paharia and K. Bhushan, "Fog Computing as a Defensive Approach Against Distributed Denial of Service (DDoS): A Proposed Architecture," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018, pp. 1-7.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY