Analyzing significance of Task Scheduling Algorithms in Cloud environment

Geetu[1] Dr.Anand Sharma[2]
[1]Research Scholar (PhD Comp. Applications), Guru Kashi University, Talwandi Sabo, Punjab, India.
[2]Assistant Professor, UCCA, Guru Kashi University, Talwandi Sabo, Punjab, India.

Abstract – Cloud computing has found its widespread use because of the several provisions it has enabled and made easier in technology sector. The research paper is intended to analyze and implement the three prominent task scheduling algorithms (FCFS, SJF, RR) which plays critical role in cloud computing. The more efficiently the processes are organized for execution on remote servers, the more optimal gets the performance of cloud setup. The paper elaborated the performance of three task scheduling algorithms under different scenarios and an effective conclusion has been reached.

Keywords – Cloud computing, FCFS, RR, SJF, processes.

I. INTRODUCTION

Cloud computing comprises of software's, servers, storage, services, databases, networking and more. The organizations offering cloud computing services are known as Cloud Service Provider (CSP). Google apps provided by Google is an example of CSP. The cloud server can be located anywhere with any configuration. Internet is the central medium via which these cloud services are provided. So it can be stated that cloud computing is a model intended to permit appropriate, on-demand network access to a common pool of configurable computing resources that can used with minimal interaction with the service provider. Cloud computing services are driven by enormous data centers comprised of number of virtualized server instances, networks, high bandwidths, cooling and power supply, and several supporting systems. Some uses of cloud computing are-create new services and apps, store, back up and recover data, host websites and blogs, deliver software on demand, analysis data [1, 2].

II. CHARACTERISTICS OF CLOUD COMPUTING

There are some natural characteristics associated with cloud computing that sustain IT from the environment or energy efficiency and economy perspective:

- **Centralization :** Centralization refers to shifting of applications, storage, and infrastructure to cloud where all computing relevant software sans applications are shifted to central server in order to minimize cost and make efficient use of resources [3, 4].
- **Virtualization:** Virtualization refers to virtualizing any components of IT including network, servers, routers, firewalls, and storage devices.
- **Automation:** It is the use of IT to reduce the human interaction in producing things, e.g. provisioning the resources. Automation reduces the cost, improves quality.
- **Broad Network Access:** Users can access Cloud services as soon as they have a device with capability to connect to the Cloud such as laptops, PDAs, mobile phones. Cloud services can be accessed from anywhere and at any time [5].
- **Internet:** Cloud use internet as a main infrastructure to connect customers to it that is widely used.
- **On demand self-service:** Users can access the cloud services on demand without interference of IT organization. One can logon to a website at any time and use them.
- **Pay per use:** Users can access the cloud services only when they use it and cloud just charge them for that specific service [6, 7].
- **Simplification:** Running many applications inside one world make it simply understandable for users.

- **Dynamic Movement of Resources:** It moves virtual machines and storage inside data center and across them as well due to more suitable conditions such as lower cost, daytime, power and consumption and maintenance concerns.
- **Standardization:** In order to eliminate the complexity from Cloud, one vendor equipment's should be used inside Cloud like unique vendor switches and routers or all the operating systems belong to one company [8].
- **Technology Convergence:** It is capable to unify all computing technologies such as storage, network, virtualization and servers in one platform to lower the cost and enhance the scaling of data center deployment [9].
- **Federation:** It is about bundling disparate Cloud computing data centers together via connecting their infrastructures to enable resource sharing.
- **Multi-tenancy (Shared):** Multiple customers use the shared infrastructure. Resources are allocated to users on demand, they are not aware of location of services and whom the resources are shared with.
- **Dynamic Provisioning (Elasticity):** Cloud responds rapidly to customer demand flexibly. This feature regards to dynamically adjusting the capacity and scaling up and down the resources such as network, storage and processing depending on customer demand requirements avoiding inessential energy and resource usage. The resources being used by customers at any given point of time are automatically monitored [10].
- **On Demand:** As opposed to ordinary computing that resources are inside IT infrastructure, in case of Cloud Computing we have access to any resources residing in the Cloud without having any dedicated ones to use internal services.

III. TASK SCHEDULING ALGORITHMS

The three prominent task scheduling algorithms are mentioned as under [11, 12].

- **First Come First Serve scheduling algorithm (FCFS)**

FCFS is a non-preemptive scheduling algorithm. FCFS, allocate the CPU to the processers in which they come in the queue. FCFS uses FIFO (First-In-First-Out) strategy. Only one process at a time can run. Processes are served according to their arrival. The process that requests for the CPU first, is allocated first to the CPU and remaining processes has to wait in ready queue until the CPU gets free. The length or the duration of the processes does not matter. Process can't be interrupted until it finishes. However, it is average performance wise, it has high average waiting time and high average turnaround time and varies every time according to burst time which makes it less capable [12, 13, 16].

- **Round Robin scheduling algorithm (RR)**

It is a preemptive scheduling algorithm. Round Robin is algorithm in which equal time slot is allocated to all the processes initially in the queue. Time slice is defined for each process by CPU. Every process is considered as equal. The working of round robin is based on time sharing. There is time limit for processing each process and after time slot comes to end, process is postponed and added back to the ready queue. If a process burst time is less than the quantum time, then CPU is immediately assigned to next process in the queue. The average time in round robin is long [12, 14, 17].

- **Shortest Job First algorithm (SJF)**

Shortest Job First is a non-preemptive scheduling algorithm in which the processes are executed on the basis of the time required for different processes to complete. The job which requires the minimum time period are executed before than the jobs requiring longer time period for execution. Shortest Job First algorithm minimize the waiting time. SJF is considered as a best algorithm because of its simple nature. SJF is most favorable as it gives minimum average time [11, 15, 18].

IV. IMPLEMENTATION AND CONTRIBUTION

Three different scenarios have been analyzed and implemented using three task scheduling algorithms discussed in section III above.

Case 1

No. of elements under study	-	07							
Process Time schedule	-	25	5	15	45	35	15	25	

- *First Come First Serve Scheduling Algorithm*

Table 1: Illustrates the processing of different processes in Case 1 as per FCFS

Process	Burst Time	Waiting Time	Turnaround Time
P2	25	0	25
P3	5	25	30
P4	15	30	45
P5	45	45	90
P6	35	90	125
P7	15	125	140
P8	25	140	165

Average waiting time (milliseconds) - 65

Average Turnaround time (milliseconds) - 88.57

Throughput - 0.079033

- *Shortest Job First*

Table 2: Illustrates the processing of different processes in Case 1 as per SJF

Process ID	Process Time	Waiting Time	Turnaround Time
2	5	0	5
3	15	5	20
6	15	20	35
1	25	35	60
7	25	60	85
5	35	85	120
4	45	120	165

Average waiting time (milliseconds) - 46.42

Average Turnaround time (milliseconds) - 70

Throughput - 0.1

Round Robin Scheduling

Table 3: Illustrates the processing of different processes in Case 1 as per RR

Process	Burst Time	Waiting Time	Turnaround Time
P2	25	107	132
P3	5	26	31
P4	15	76	91
P5	45	117	162
P6	35	118	153
P7	15	80	95
P8	25	108	133

Average waiting time (milliseconds) - 90.28

Average turnaround time (milliseconds) - 113.85

Throughput - 0.0614805519

Comparative Table

Table 4: Comparative table shown readings of three parameters under discussion

Scheduling Algorithm	Average Waiting Time (milliseconds)	Average Turnaround Time (milliseconds)	Throughput (No. of processes/ Avg. Turnaround Time
FCFS	65.000000	88.571429	0.0790322576
SJF	46.428571	70.000000	0.1
RR	90.285714	113.857143	0.0614805519

Comparative visualization of Average Waiting Time of task scheduling algorithm is shown in Fig. 1.

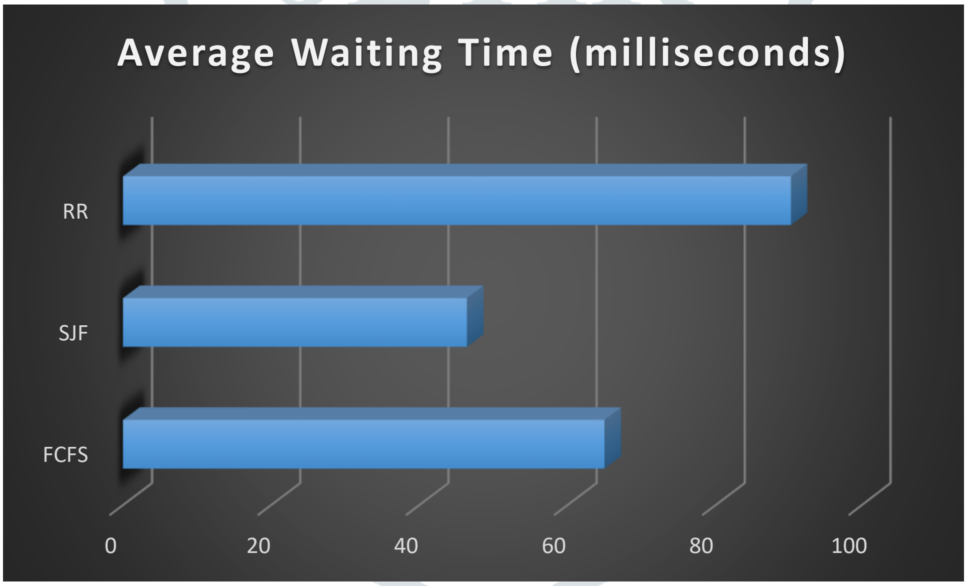

Fig. 1 Comparative visualization of Average Waiting Time of task scheduling algorithm

Comparative visualization of Average Turnaround Time of task scheduling algorithm is shown in Fig. 2

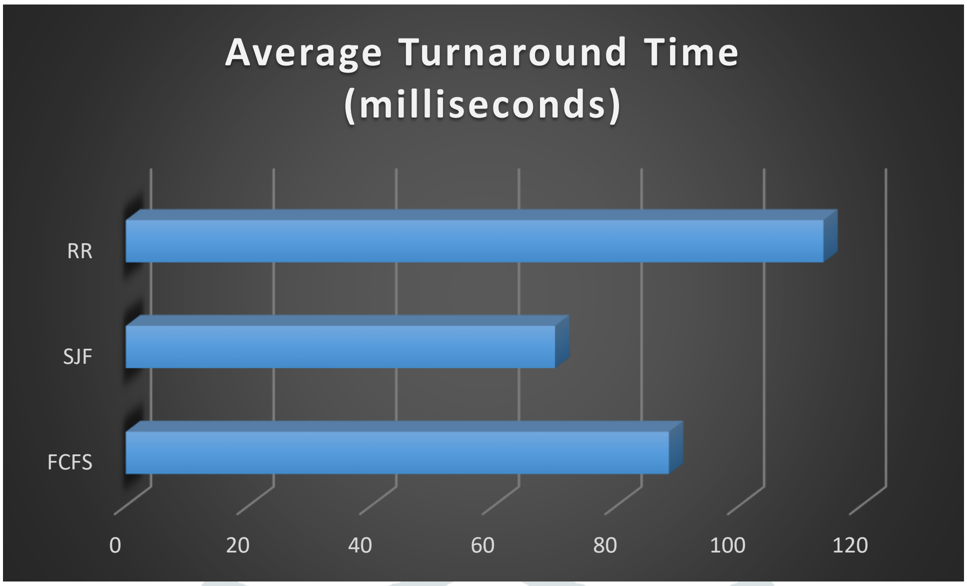

Fig. 2 Comparative visualization of Average Turnaround Time of task scheduling algorithm

Comparative visualization of Throughput of task scheduling algorithm is shown in Fig. 3.

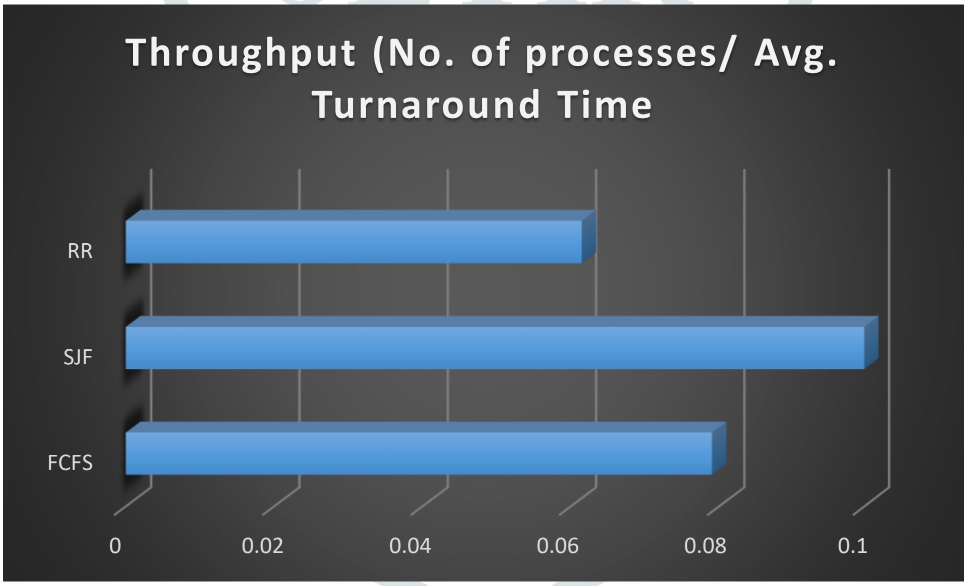

Fig. 3 Comparative visualization of Throughput of task scheduling algorithm

Case 2

No. of elements under study - 06

Process Time schedule - 4 2 1 3 2 1

- *First Come First Serve Scheduling Algorithm*

Table 5: Illustrates the processing of different processes in Case 2 as per FCFS

Process	Burst Time	Waiting Time	Turnaround Time
P2	4	0	4
P3	2	4	6
P4	1	6	7
P5	3	7	10

P6	2	10	12
P7	1	12	13

Average Waiting Time (milliseconds) - 6.500000

Average Turnaround Time (milliseconds) - 8.666667

Throughput - 0.6923076656

- *Shortest Job First*

Table 6: Illustrates the processing of different processes in Case 2 as per SJF

Process ID	Process Time	Waiting Time	Turnaround Time
3	1	0	1
6	1	1	2
5	2	2	4
2	2	4	6
4	3	6	9
1	4	9	13

Average waiting time (milliseconds) - 3.666667

Average Turnaround Time (milliseconds) - 5.833333

Throughput - 1.0285714873

- *Round Robin scheduling*

Table 7: Illustrates the processing of different processes in Case 2 as per RR

Process	Burst Time	Waiting time	Turnaround Time
P2	4	8	12
P2	2	2	4
P4	1	4	5
P5	3	9	12
P6	2	6	8
P7	1	8	9

Average Waiting Time (milliseconds) - 6.166667

Average Turnaround Time (milliseconds) - 8.333333

Throughput - 0.7200000288

Comparative Table

Table 8: Comparative table shown readings of three parameters under discussion

Scheduling Algorithm	Average Waiting time (milliseconds)	Average Turnaround Time (milliseconds)	Throughput (No. of processes/ Avg. Turnaround Time
FCFS	6.500000	8.666667	0.6923076656
SJF	3.666667	5.833333	1.0285714873
RR	6.166667	8.333333	0.7200000288

Comparative visualization of Average Waiting Time of task scheduling algorithm is shown in Fig. 4.

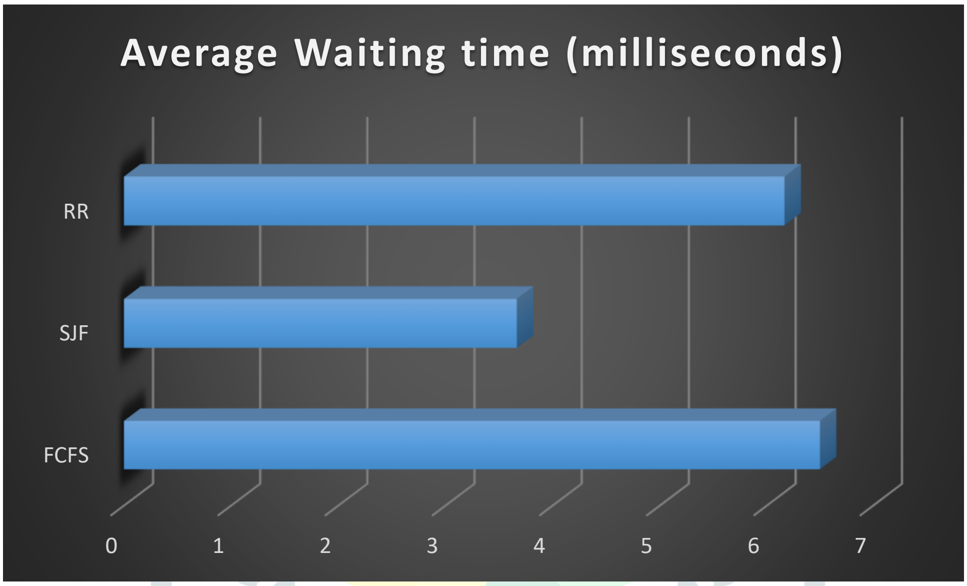

Fig. 4 Comparative visualization of Average Waiting Time of task scheduling algorithm

Comparative visualization of Average Turnaround Time of task scheduling algorithm is shown in Fig. 5.

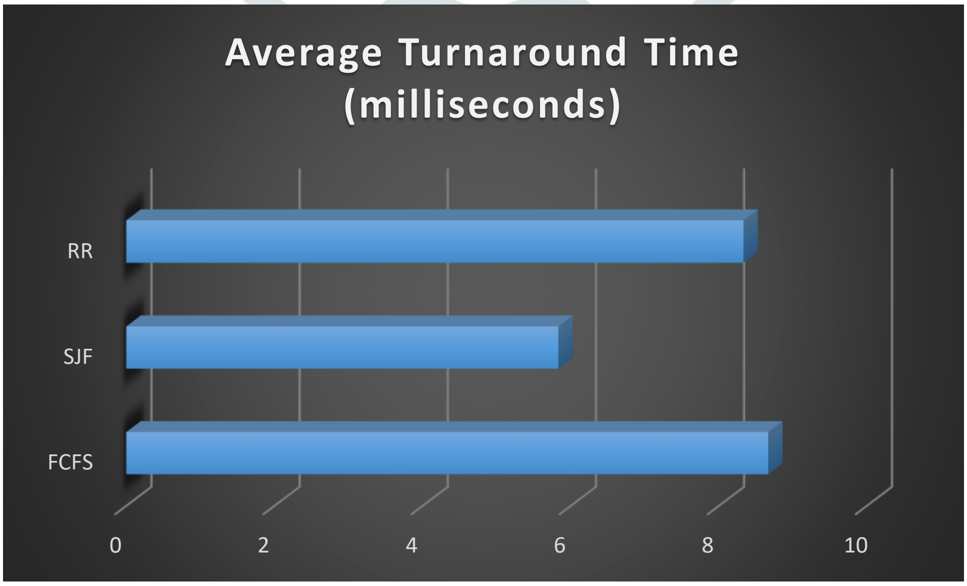

Fig. 5 Comparative visualization of Average Turnaround Time of task scheduling algorithm

Comparative visualization of throughput of task scheduling algorithm is shown in Fig. 6.

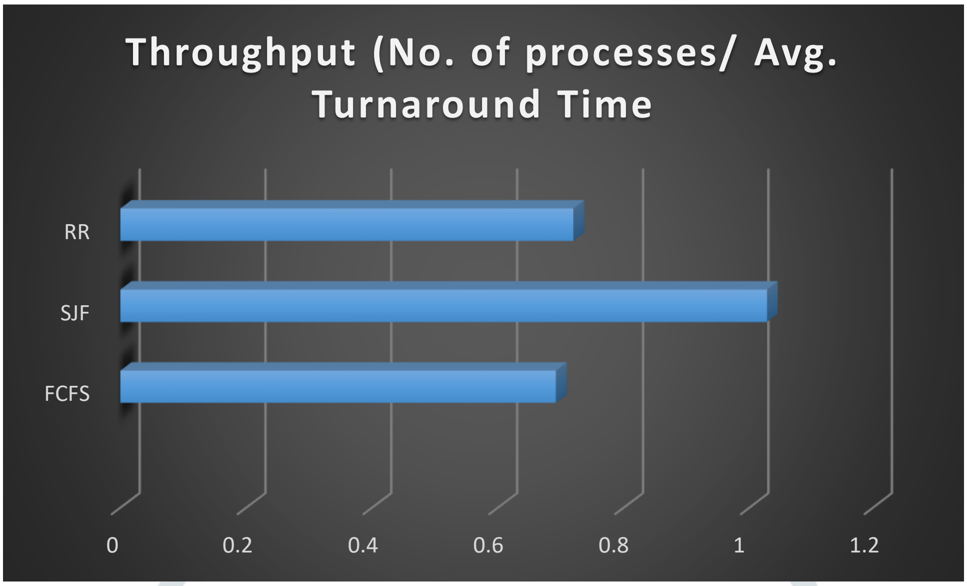

Fig. 6 Comparative visualization of throughput of task scheduling algorithm

V. CONCLUSION

Cloud computing is widely used in the world but it lacks with many issues mainly service reliability. The performance of cloud services is always analyzed upon the performance of user tasks submitted to the system. Task scheduling plays a significant role in enhancing the performance of the cloud services. The research work conducted in the paper emphasized on effective and efficient scheduling of tasks/jobs intended to be performed in cloud environment. The more the effectiveness in handling the numerous jobs in cloud computing, more would be the efficiency at the cost of minimum energy consumption.

REFERENCES

1. N. Bacanin, M. Tuba, in *IEEE Congress on Evolutionary Computation, CEC 2015*. Fireworks algorithm applied to constrained portfolio optimization problem (IEEENew York, 2015), pp. 1242–1249. https://doi.org/10.1109/CEC.2015.7257031.
2. deOliveira L.L., Freitas A.A., Tinós R., Multi-objective genetic algorithms in the study of the genetic code's adaptability. Inf. Sci.425 **:**, 48–61 (2018). https://doi.org/10.1016/j.ins.2017.10.022
3. A.S. Pillai, K. Singh, V. Saravanan, A. Anpalagan, I. Woungang, L. Barolli, A genetic algorithm-based method for optimizing the energy consumption and performance of multiprocessor systems. Soft Comput.22(10), 3271–3285 (2018). https://doi.org/10.1007/s00500-017-2789-y
4. Abusfian Elgelany, Nadar Nada "Energy Efficiency for Data Centre and Cloud Computing: A Literature Review" Sudan University, Khartoum, Sudan, Fatih University, Istanbul, Turkey.
5. Ankita Choudhary, Shilpa Rana, K.J.Matahai, "A Critical Analysis of Energy Efficient Virtual Machine Placement Techniques and its Optimization in Cloud Computing" Department of CEA, NITTTR, Bhopal, India.
6. Gagandeep Jagdev et al., "Analyzing Commercial Aspects and Security Concerns Involved in Energy Efficient Cloud Computing" in International Journal of Scientific and Technical Advancements (IJSTA), ISSN-2454-1532, 2016.
7. Z. Zheng, N. Saxena, K.K. Mishra, A.K. Sangaiah, Guided dynamic particle swarm optimization for optimizing digital image watermarking in industry applications. Futur. Gener. Comput. Syst. (2018). https://doi.org/10.1016/j.future.2018.05.027

8. Gagandeep Jagdev et al., "Implementation of DES and AES Cryptographic Algorithms in Accordance with Cloud Computing", International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), ISSN: 2349-4840 (P), ISSN: 2349-4859(O), DOI: http://dx.doi.org/10.20431/2349-4859.0404001**,** Volume 4, Issue 4, 2017, pp – 1 - 14.
9. N. Zhang, X. Yang, M. Zhang, Y. Sun, K. Long, A genetic algorithm-based task scheduling for cloud resource crowd-funding model. Int. J. Commun. Syst.**31**(1) (2018). https://doi.org/10.1002/dac.3394
10. Awada Uchechukwu, keqiu Li, Yanming Shen, "Improving Cloud Computing Energy Efficiency" School of Computer Science and Technology, Dalian University of technology, Dalian Shen.
11. Gagandeep Jagdev et al., "Analyzing Working of DES and AES Algorithms in Cloud Security", International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), ISSN: 2349-4840 (P), ISSN: 2349-4859(O), DOI: http://dx.doi.org/10.20431/2349-4859.0403001**,** Volume 4, Issue 3, 2017, pp – 1 - 9.
12. A. Gerasoulis, T. Yang, A comparison of clustering heuristics for scheduling directed acyclic graphs on multiprocessors. J. Parallel Distrib. Comput. **16:**, 276–291 (1992). Elsevier. https://doi.org/10.1016/0743-7315(92)90012-C
13. B. Zhang, M. Zhang, Y. Zheng, in *Advances in Swarm Intelligence - 5th International Conference, ICSI 2014, October 17-20, 2014, Proceedings, Part I*. Improving enhanced fireworks algorithm with new Gaussian explosion and population selection strategies (SpringerCham, 2014), pp. 53–63. https://doi.org/10.1007/978-3-319-11857-4_7
14. Gagandeep Jagdev et al., "Elaborating Security Algorithms in Cloud Environment", International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), ISSN: 2349-4840 (P), ISSN: 2349-4859(O), Volume 5, Issue 1, 2018, pp.1-8, *DOI: http://dx.doi.org/10.20431/2349-4859.0501001*
15. J.Srinivas, K.Venkata subba Reddy.Dr. A. Moiz Qyser "cloud computing basics" assistant Professor, Dept. of CSE, M.J College of Engg & Tech, Hydrabad, India.
16. Kamyab Khajehei "Role of Virtualization in Cloud Computing" Dashtestan Branch Islamic Azad University, Borazjan, Iran.
17. Gagandeep Jagdev et al., "Comparative Analysis and Implementation of Cryptographic Algorithms in Cloud Computing", International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), ISSN: 2349-4840 (P), ISSN: 2349-4859(O), Volume 5, Issue 1, 2018, pp.1-9, *DOI: http://dx.doi.org/10.20431/2349-4859.0501003.*
18. S. Chen, Y. Liu, L. Wei, B. Guan, PS-FW: A hybrid algorithm based on particle swarm and fireworks for global optimization. Comp. Int. Neurosc.2018:, 6094685–1609468527 (2018). https://doi.org/10.1155/2018/6094685