International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 6, Issue 2, 2019, PP 1-11
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
DOI: http://dx.doi.org/10.20431/2349-4859.0602001
www.arcjournals.org

Implementing and Evaluating Impact of Task Scheduling in Cloud Environment

Dr. Gagandeep Jagdev

Dept. of Computer Science, Punjabi University Guru Kashi College, Damdama Sahib, Punjab, India.

Corresponding Author** ***Dr. Gagandeep Jagdev, *Dept. of Computer Science, Punjabi University Guru Kashi College, Damdama Sahib, 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*

1. 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].

2. 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.

3. 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].

4. 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 - 10

Process Time schedule - 3 1 3 2 4 2 5 7 4 5

- **First Come First Serve Scheduling Algorithm**

Table1. *Illustrates the processing of different processes in Case 1 as per FCFS*

Process	Burst Time	Waiting Time	Turn Around Time
P2	3	0	3
P3	1	3	4
P4	3	4	7
P5	2	7	9
P6	4	9	13
P7	2	13	15
P8	5	15	20
P9	7	20	27
P10	4	27	31
P11	5	31	36

Total Waiting Time (milliseconds) - 129

Average waiting time (milliseconds) - 12.9

Total Turnaround Time (milliseconds) - 165

Average Turnaround time (milliseconds) - 16.5

Throughput (No. of processes / Average Turnaround Time) - 0.6060

- **Shortest Job First**

Table2. *Illustrates the processing of different processes in Case 1 as per SJF*

Process ID	Burst Time	Waiting Time	Turnaround Time
2	1	0	1
4	2	1	3
6	2	3	5
1	3	5	8
3	3	8	11
5	4	11	15
9	4	15	19
7	5	19	24
10	5	24	29
8	7	29	36

Total Waiting Time (milliseconds) - 115

Average waiting time (milliseconds) - 11.500000

Total Turnaround Time (milliseconds) - 151

Average Turnaround time (milliseconds) - 15.1

Throughput (No. of processes / Average Turnaround Time) - 0.662251

- **Round Robin Scheduling**

Table3. *Illustrates the processing of different processes in Case 1 as per RR*

Process	Burst Time	Waiting Time	Turnaround Time
P2	3	17	20
P3	1	2	3
P4	3	16	19
P5	2	4	6
P6	4	16	20
P7	2	8	10

P8	5	24	29
P9	7	25	32
P10	4	22	26
P11	5	26	31

Total Waiting Time (milliseconds) - 160

Average waiting time (milliseconds) - 16.000000

Total Turnaround Time (milliseconds) - 196

Average Turnaround time (milliseconds) - 19.600000

Throughput (No. of processes / Average Turnaround Time) - 0.51020

- **Comparative Table**

Table4. *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	12.9	16.5	0.6060
SJF	11.5	15.1	0.662251
RR	16	19.6	0.51020

Comparative visualization of three parameters shown in Table 4 considering 10 processes is shown in Fig. 1.

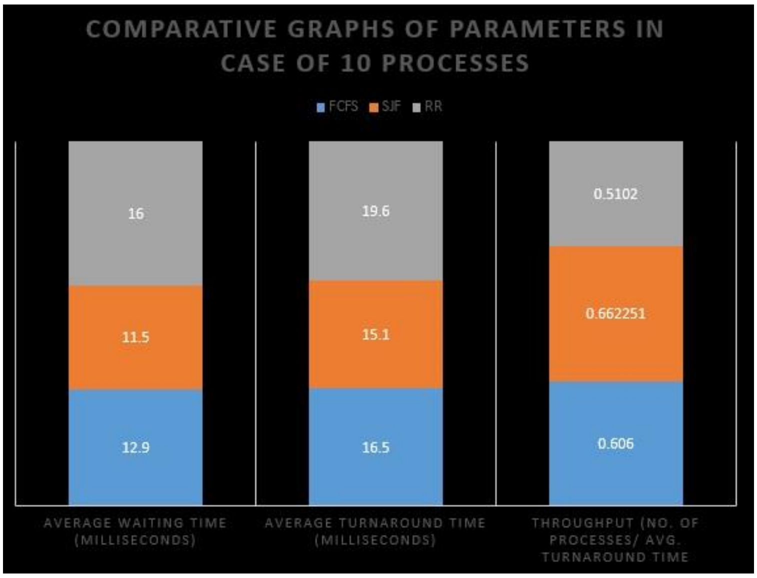

Fig1. *Comparative visualization of three parameters as per Table 4*

Case 2

No. of elements under study - 20

Process Time schedule - 3 6 7 2 5 9 6 4 1 3 5 9 7 4 2 5 4 7 1 2

- **First Come First Serve Scheduling Algorithm**

Table5. *Illustrates the processing of different processes in Case 2 as per FCFS*

Process	Burst Time	Waiting Time	Turnaround Time
P2	3	0	3
P3	6	3	9
P4	7	9	16
P5	2	16	18
P6	5	18	23
P7	9	23	32
P8	6	32	38

P9	4	38	42
P10	1	42	43
P11	3	43	46
P12	5	46	51
P13	9	51	60
P14	7	60	67
P15	4	67	71
P16	2	71	73
P17	5	73	78
P18	4	78	82
P19	7	82	89
P20	1	89	90
P21	2	90	92

Total Waiting Time (milliseconds) - 931

Average waiting time (milliseconds) - 46.55

Total Turnaround Time (milliseconds) - 1023

Average Turnaround time (milliseconds) - 51.15

Throughput (No. of processes / Average Turnaround Time) - 0.391006

- **Shortest Job First**

Table6. *Illustrates the processing of different processes in Case 2 as per SJF*

Process ID	Process Time	Waiting Time	Turnaround Time
9	1	0	1
19	1	1	2
15	2	2	4
4	2	4	6
20	2	6	8
10	3	8	11
1	3	11	14
17	4	14	18
14	4	18	22
8	4	22	26
5	5	26	31
11	5	31	36
16	5	36	41
2	6	41	47
7	6	47	53
18	7	60	67
13	7	67	74
6	9	74	83
12	9	83	92

Total Waiting Time (milliseconds) - 604

Average waiting time (milliseconds) - 30.2

Total Turnaround Time (milliseconds) - 696

Average Turnaround time (milliseconds) - 34.800000

Throughput (No. of processes / Average Turnaround Time) - 0.574712

- **Round Robin scheduling**

Table7. *Illustrates the processing of different processes in Case 2 as per RR*

Process	Burst Time	Waiting time	Turnaround Time
P2	3	36	39
P3	6	61	67
P4	7	76	83

P5	2	6	8
P6	5	65	70
P7	9	79	88
P8	6	67	73
P9	4	46	50
P10	1	16	17
P11	3	47	50
P12	5	67	72
P13	9	76	85
P14	7	76	83
P15	4	53	57
P16	2	26	28
P17	5	71	76
P18	4	57	61
P19	7	75	82
P20	1	34	35
P21	2	34	36

Total Waiting Time (milliseconds) - 1068

Average waiting time (milliseconds) - 53.4

Total Turnaround Time (milliseconds) - 1160

Average Turnaround time (milliseconds) - 58

Throughput (No. of processes / Average Turnaround Time) - 0.344827

- **Comparative Table**

Table8. *Comparative table shown readings of three parameters under discussion as per case 2*

Scheduling Algorithm	Average Waiting time (milliseconds)	Average Turnaround Time (milliseconds)	Throughput (No. of processes/ Avg. Turnaround Time
FCFS	46.55	51.15	0.391006
SJF	30.2	34.8	0.574712
RR	53.4	58	0.344827

Comparative visualization of three parameters shown in Table 8 considering 20 processes is shown in Fig. 4.

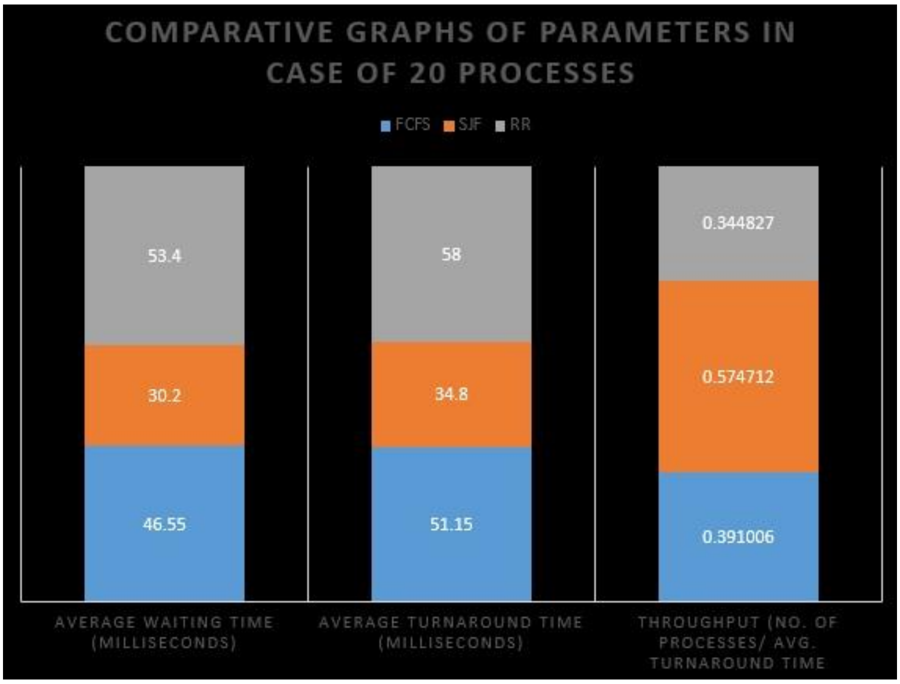

Fig4. *Comparative visualization of three parameters as per Table 8*

Case 3

No. of elements under study - 30

Process Time schedule -3 6 7 2 5 9 6 4 1 3 5 9 7 4 2 5 4 7 1 2 2 4 6 8 1 3 5 7 9 5

- **First Come First Serve Scheduling Algorithm**

Table9. *Illustrates the processing of different processes in Case 3 as per FCFS*

Process	Burst Time	Waiting Time	Turnaround Time
P2	3	0	3
P3	6	3	9
P4	7	9	16
P5	2	16	18
P6	5	18	23
P7	9	23	32
P8	6	32	38
P9	4	38	42
P10	1	42	43
P11	3	43	46
P12	5	46	51
P13	9	51	60
P14	7	60	67
P15	4	67	71
P16	2	71	73
P17	5	73	78
P18	4	78	82
P19	7	82	89
P20	1	89	90
P21	2	90	92
P22	2	92	94
P23	4	94	98
P24	6	98	104
P25	8	104	112
P26	1	112	113
P27	3	113	116
P28	5	116	121
P29	7	121	128
P30	9	128	137
P31	5	137	142

Total Waiting Time (milliseconds) - 2046

Average waiting time (milliseconds) - 68.2

Total Turnaround Time (milliseconds) - 2188

Average Turnaround time (milliseconds) - 72.933

Throughput (No. of processes / Average Turnaround Time) - 0.411336

- **Shortest Job First**

Table10. *Illustrates the processing of different processes in Case 3 as per SJF*

Process	Burst Time	Waiting Time	Turnaround Time
9	1	0	1
19	1	1	2
25	1	2	3
4	2	3	5
20	2	5	7
21	2	7	9
15	2	9	11
10	3	11	14

1	3	14	17
26	3	17	20
17	4	20	24
22	4	24	28
14	4	28	32
8	4	32	36
11	5	36	41
16	5	41	46
5	5	46	51
27	5	51	56
30	5	56	61
7	6	61	67
23	6	67	73
2	6	73	79
3	7	79	86
13	7	86	93
28	7	93	100
18	7	100	107
24	8	107	115
12	9	115	124
29	9	124	133
6	9	133	142

Total Waiting Time (milliseconds) - 1441

Average waiting time (milliseconds) - 48.0333

Total Turnaround Time (milliseconds) - 1583

Average Turnaround time (milliseconds) - 52.76667

Throughput (No. of processes / Average Turnaround Time) - 0.568540

- **Round Robin Scheduling**

Table11. *Illustrates the processing of different processes in Case 3 as per RR*

Process	Burst Time	Waiting Time	Turnaround Time
P2	3	55	58
P3	6	95	101
P4	7	120	127
P5	2	6	8
P6	5	99	104
P7	9	128	137
P8	6	101	107
P9	4	65	69
P10	1	16	17
P11	3	66	69
P12	5	101	106
P13	9	125	134
P14	7	120	127
P15	4	72	76
P16	2	26	28
P17	5	105	110
P18	4	76	80
P19	7	119	126
P20	1	34	35
P21	2	34	36
P22	2	36	38
P23	4	79	83
P24	6	106	112
P25	8	118	126
P26	1	44	45

P27	3	84	87
P28	5	108	113
P29	7	117	124
P30	9	117	126
P31	5	112	117

Total Waiting Time (milliseconds) - 2484

Average waiting time (milliseconds) - 82.8

Total Turnaround Time (milliseconds) - 2626

Average Turnaround time (milliseconds) - 87.53333

Throughput (No. of processes / Average Turnaround Time) - 0.3427267

- **Comparative Table**

Table12. *Comparative table shown readings of three parameters under discussion as per case 3*

Scheduling Algorithm	Average Waiting time (milliseconds)	Average Turnaround Time (milliseconds)	Throughput (No. of processes/ Avg. Turnaround Time
FCFS	68.2	72.933	0.411336
SJF	48.033	52.7666	0.568540
RR	82.8	87.533	0.3427267

Comparative visualization of three parameters shown in Table 12 considering 30 processes is shown in Fig. 5.

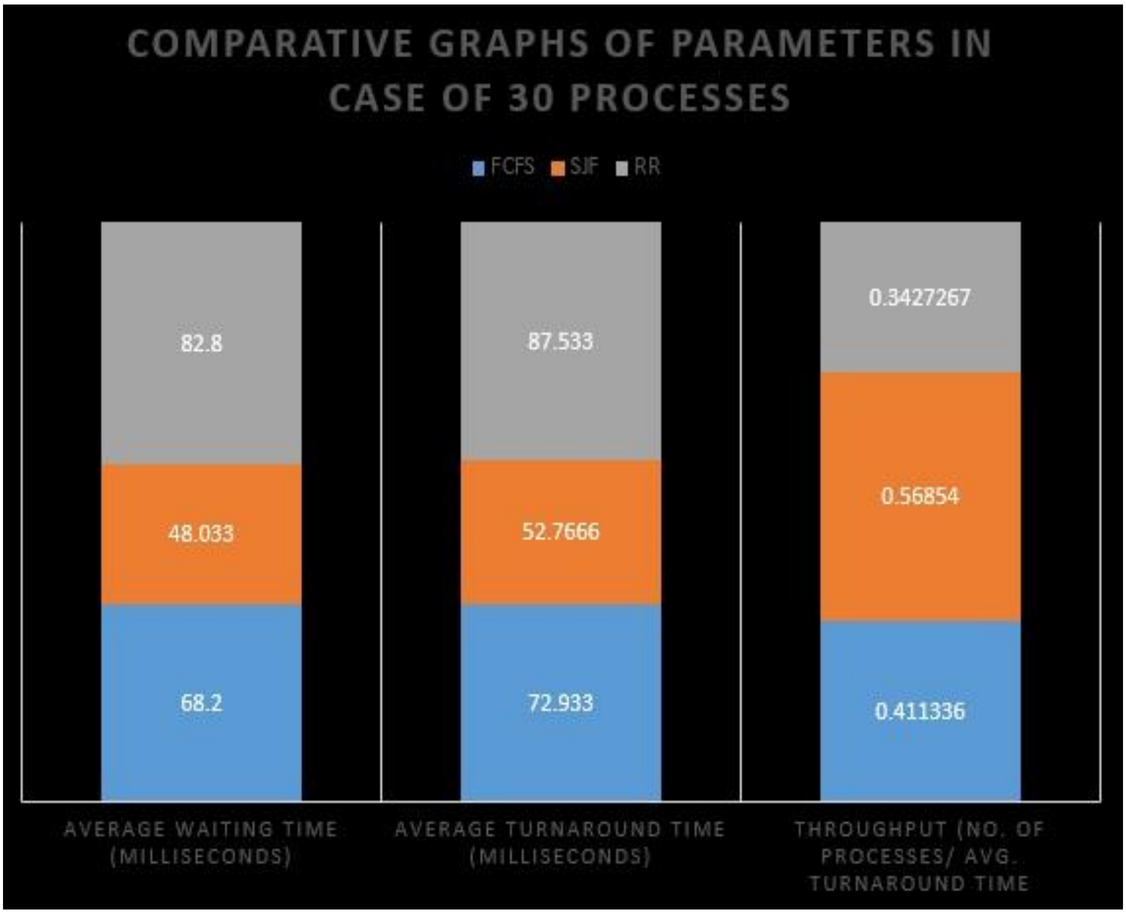

Fig5. *Figure shows comparative visualization of three parameters shown in Table 12*

5. 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. On the basis of the three cases studied and implemented in the research paper, it can be concluded that SJF is the best among three algorithms under study in terms of throughput.

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] Gurpreet Kaur & Dr. Gagandeep Jagdev (2017). Implementation of DES and AES Cryptographic Algorithms in Accordance with Cloud Computing, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 4(4), pp.1-14, DOI: http://dx.doi.org/10.20431/2349-4859.0404001

[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] Shaffy Bansal & Dr. Gagandeep Jagdev (2017). Analyzing Working of DES and AES Algorithms in Cloud Security, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 4(3), pp.1-9, DOI: http://dx.doi.org/10.20431/2349-4859.0403001

[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] Shaffy Bansal & Dr. Gagandeep Jagdev (2018). Elaborating Security Algorithms in Cloud Environment, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 5(1), 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] Shaffy Bansal & Dr. Gagandeep Jagdev (2018). Comparative Analysis and Implementation of Cryptographic Algorithms in Cloud Computing, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 5(1), pp.17-25, 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/1 0.1155/20 18/6094685

AUTHOR'S BIOGRAPHY

Dr. Gagandeep Jagdev, is a faculty member in Dept. of Computer Science, Punjabi University Guru Kashi College, Damdama Sahib (PB). His total teaching experience is more than 12 years and has 134 international and national publications in reputed journals and conferences to his credit. He is also a member of editorial board of several international peer-reviewed journals and has been active Technical Program Committee member of several international and national conferences conducted by renowned universities and academic institutions. His field of expertise is Big Data, Data Mining, Image processing in Diabetic Retinopathy, ANN, Biometrics, RFID, Cloud Computing, Cloud security, Cryptography, and VANETS.

Citation: *Gagandeep, j. (2019). Implementing and Evaluating Impact of Task Scheduling in Cloud Environment. International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 6(2), pp.1-11. http://dx.doi.org/10.20431/2349-4859.0602001*