Journal of Engineering Science and Technology
Vol. 11, No. 9 (2016) 1282 - 1295

ADAPTATION OF JOHNSON SEQUENCING ALGORITHM FOR JOB SCHEDULING TO MINIMISE THE AVERAGE WAITING TIME IN CLOUD COMPUTING ENVIRONMENT

SOUVIK PAL*, PRASANT KUMAR PATTNAIK

School of Computer Engineering, KIIT University, Campus 15, Bhubaneswar, India
*Corresponding Author: souvikpal22@gmail.com

Abstract

Cloud computing is an emerging paradigm of Internet-centric business computing where Cloud Service Providers (CSPs) are providing services to the customer according to their needs. The key perception behind cloud computing is on-demand sharing of resources available in the resource pool provided by CSP, which implies new emerging business model. The resources are provisioned when jobs arrive. The job scheduling and minimization of waiting time are the challenging issue in cloud computing. When a large number of jobs are requested, they have to wait for getting allocated to the servers which in turn may increase the queue length and also waiting time. This paper includes system design for implementation which is concerned with Johnson Scheduling Algorithm that provides the optimal sequence. With that sequence, service times can be obtained. The waiting time and queue length can be reduced using queuing model with multi-server and finite capacity which improves the job scheduling model.

Keywords: Cloud broker, Cloud computing, Queuing model, Job scheduling.

1. Introduction

Cloud computing is a service-oriented model, which is associated with academic research and IT Industry. In cloud computing environment, computing machines are to be built from physically distributed components such as processing elements, data storage and software resources [1]. A cloud infrastructure can be structured into services in agreement with the requirement of the client, which can grow or shrink in real-time scenario [2, 3].

The end users use the computing and physical resources in utility manner which

Nomenclatures

c	The number of servers
$E[\tau]$	Average or mean Inter-arrival time, ns
$E(S)$	Average or mean service time, ns
K	Maximum capacity of the system
Lq	Average number of customers in the queue
Ls	Average number of customers in the system
$P(x)$	Probability of x arrivals; x=0,1,2,.....
Wq	Average waiting time in the queue, ns
Ws	Average waiting time in the system, ns
x	Number of arrivals per unit of time

Greek Symbols

λ_n	Average arrival rate, n=0, 1, 2,…, K-1, $\lambda_n = \lambda$; $n \geq K$, $\lambda_n = 0$.
μ	Average service rate
ρ	ρ is denoted as $\rho = \lambda/c_\mu$
$\overline{\lambda}$	$\overline{\lambda}$ is denoted as $\overline{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \lambda(1-P_K)$
τ	Inter-arrival time, ns

Abbreviations

CSP	Cloud Service Provider
SC	Service Centre
SLA	Service Level Agreement

describes a business framework for delivering the services and computing power on-demand. After getting the services, cloud users have to pay the service providers based on their usage. This situation leads to a business relationship through Cloud Brokerage Services which act as mediator who facilitates the users to choose the best resources.

Cloud broker enforces easy access to cloud services from the service providers. Through the Cloud Broker, the clients can easily get the services and deploy the applications onto cloud platform. Cloud Broker provides a platform whereby he collects the information from the user, analyse the data, send the data to the CSP on behalf of the user and also provides the billing services. Cloud Broker provides data integration services across all the components of the cloud services. Cloud brokers are there to assist the users to keep the track of all the activities such as execution time of each request, specific data centre used, numbers of data centres, calculation of waiting time of each request. The user-requests can be scheduled using Johnson Scheduling algorithm and the waiting time can be reduced by using Queuing theory. Cloud Brokers are responsible for implementing these algorithms which may facilitates the users as well as the CSPs. This paper emphasizes on solving the issue of job scheduling in cloud computing environment by Johnson algorithm. Moreover, a system design has been modelled to fit Johnson sequencing algorithm and to minimize the waiting time queuing theory has been used. The paper is organized as follows:

In Section 2, we have discussed modelling on service job scheduling in cloud computing environment which illustrates the system design, state diagram of the

design, system flow, and queuing model. In Section 3, we have presented the numerical analysis and comparison study. Section 4 concludes the work and lastly an Appendix A has been appended to this article.

1.1. Literature survey of the related work

Job Scheduling problem is a core research issue in the field of cloud computing [4]. It is concerned with minimization of the waiting time after using the scheduling algorithms. Job scheduling enhances the functioning of cloud to gain the maximum profit. The aim of using different scheduling algorithms is to find a proper list in which the tasks are scheduled to execute and reduce the total job execution time. There are diverse types of scheduling algorithms presented in the cloud environment that varies from the traditional scheduling algorithms which may not apply to the cloud systems since cloud is a distributed environment that comprises of heterogeneous systems.

Cloud computing, as market-oriented service utilities begun with task scheduling concept accordingly. Some of the basic scheduling algorithms can be used for scheduling in cloud computing, such as First Come First Serve (FCFS) Algorithm (in the queue the job comes first, is served first) [4, 5]. As well as Round Robin (RR) Algorithm (the jobs are being looped using a specific time slice or time quantum until they completes their execution) [6 - 8]. There are some other algorithms [9] like Resource-Aware-Scheduling Algorithm (RASA) [10], Reliable Scheduling Distributed in Cloud Computing (RSDC) [11], Priority-based Service Scheduling Policy [12] and Extended Max-Min Scheduling [13], as well as Optimistic Differentiated Job Scheduling algorithm [14]. In the next section we will present some of the related work on cloud computing scheduling methods.

Cepek et al. [15] have discussed non-pre emptive flow shop scheduling whereby a set of jobs are processed through a set of service machines with some distinct order. This is used for minimizing the makespan which in turn speed up the performance of the system. However, Johnson Sequencing Rule (Flow Shop Algorithm) with N Jobs and 2 Machines was proposed to find the optimal sequence. Johnson [16] has been applied in various aspects of operational research [17], Industry and computer applications [18, 19]. This rule finds an optimum schedule with minimized makespan.

Buyya et al. [20] have presented cloud computing as delivering IT services which facilitates the user with the computing utilities. Their paper deals with user-driven service control and computational capabilities. The resources are allocated in accordance with Service Level Agreement (SLA). Moreover, the paper also deals with how Virtual Machines (VMs) are working according to the tasks requested by the customers.

Jiang and Ni [21] have presented FCFS algorithm combined with backfilling and priority strategy for task scheduling in grid computing environment. This concept has been used for reducing the response time and also for improving the system resource utilization. They have also considered the concept of resource recycling after completion of all tasks.

Li [22] has focused on the resources utilization to gain the highest job scheduling system performance. For that he has considered M/G/1 queuing model

with non-pre-emptive priority. Their paper shows the system cost function to get the estimated optimistic value of service designed for each job, which guarantees Quality of Service (QoS) conditions of customer jobs and the optimal profits for the service providers.

Sowjanya et al. [23] have discussed the queue length and waiting time. They have applied M/M/s queuing model to reduce the mean queue length and waiting time by varying the number of servers.

Khazaei et al. [24] have described novel estimated methodical model which deals with performance evaluation of the servers. Their paper is meant to get the approximation of the total probability distribution of the response time of the requests. Their model allows the cloud providers to establish the relationship between input buffer size and the number of Service Centres (SCs). They have analyzed immediate service probability, blocking probability, and the average number of tasks in the system.

Pal and Pattnaik [25] have discussed on the classification of virtualization environment in cloud. The virtualization in cloud computing includes automatic resources provisioning, scheduling of user request, accounting of renewal request and so on. Virtualization can be implemented through cloud broker. Cloud broker [26] acts as an interface to facilitate the IT user to choose the appropriate data centre capable of providing adequate resources according to the requirement of the customer. It is also responsible for scheduling of the tasks requested by the customer. They have discussed on the design aspect of the cloud broker and work flow strategy using sequence diagram. They have also shown the procedures how the scheduling can be enabled in cloud broker.

Spicuglia et al. [27] have shown the procedures of collecting data from different data centres in heterogeneous systems. They have discussed about how to join the best queue and plug and play workload controller which tries to minimize the variance and upper percentile of response times.

Guo et al. [28] have described dynamic performance optimisation in cloud environments using M/M/S Queuing system. They have proposed the function, strategy and synthesis optimisation mode using the queuing model. As well as they have compared and analyzed the Shortest Service Time First and FCFS methods which shows optimised results of average queue length, average waiting time and the number of customers.

1.2. Objective of the study

In the previous section, we have discussed the FCFS algorithm, Johnson Sequencing algorithm, queuing model with multi-server and finite capacity in the system. In this paper, a system model has been designed where Johnson Algorithm and queuing system has been implemented to minimize the service time in cloud computing environment. Considering a batch comprising of certain number of jobs, this is easy to find service time using Gantt chart. So that, the service times for each job can be obtained using that Gantt chart. After that, using the M/M/c/K queuing system it can be reduced the average number of customers in the queue and in the system and as well as the average waiting time in the system and in the queue.

2. Modelling On Job Scheduling in Cloud Computing Environment

In cloud computing environment, different types of user-specific jobs are requested. Those jobs are required to be scheduled to get the optimal sequence which can be used to reduce the waiting time using existing queuing model.

2.1. System design

This section illustrates the aspects of system design using a schematic diagram which deals with the scheduling phase and the queuing model. In scheduling phase, Johnson Sequencing Algorithm has been considered to provide an optimised sequence of jobs. As a queuing system, M/M/c/K model has been taken to find different waiting times.

In our design as shown in Fig. 1, there are *n* numbers of customers who make the request to the cloud broker. The requests can be Resource-based, Infrastructure-based, Platform-based, Software-based or Storage-based. Cloud broker, as an intermediation service, does identity and access management capabilities. After authorization of the customer-access, in accordance with the SLA service, all the requirements and user data are reported to the service provider. The Monitor module gathers all the requests or jobs and resource information from the user for a particular time span. The Analyser module determines the available resources. If the requested resources are available, the resources are provisioned according to the SLA service terms and conditions. After getting the resource confirmation, Scheduler module schedules the jobs according to Johnson Algorithm, which finds optimum sequence and minimize makespan which in turn reduce the waiting time of the customers. These scheduled jobs are passed through M/M/c/K Queuing System that leads to finding different waiting lines which will be discussed later. Efficient usage of servers can maximize the sharing of systems and computational resources beside minimizing the cost complexity, and reducing the waiting time.

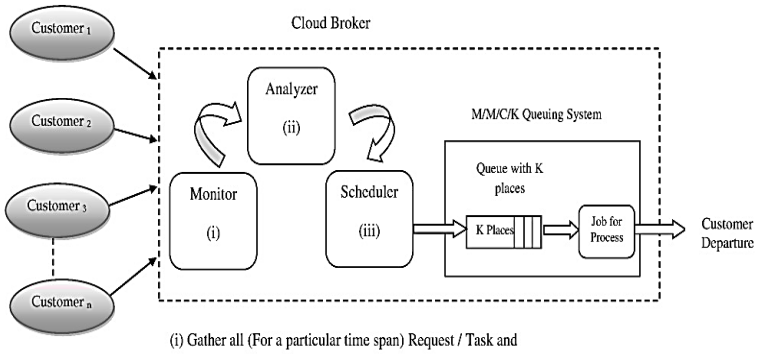

Fig. 1. System design.

2.2. State diagram

This section deals with the state diagram of the system design as shown in Fig. 2. Cloud user first interacts with the cloud broker and sends User_Request ( ) in

order to get access to the cloud infrastructure. All the information and requests are sent using Send_Info( ) and send_request ( ).

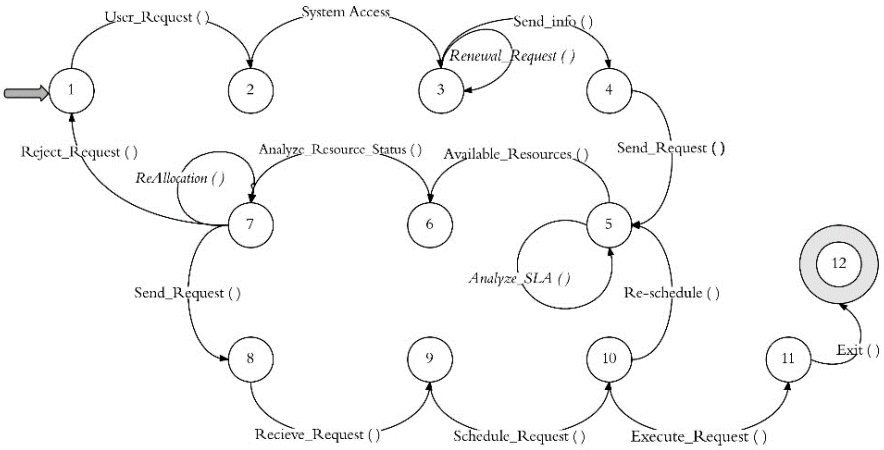

Fig. 2. State diagram.

At that stage, for the existing customers, Renewal_Request ( ) can be executed, for which an existing service can be renewed in accordance with the requirements of the user. All the requests should meet the SLA. The service requested by the user can be processed only after analyzing the available resources using Available_Resources ( ) and Analyze_Resource_Status ( ). The resources can be provisioned according to user's needs. If the requested resources are not available, the request can be thrown out by Reject_Request ( ). At that stage, the resources can be reallocated for the existing customers by ReAllocation ( ). Then the requests are in ready state and prepared to be executed and the requests are to be scheduled using Schedule_Request ( ). After scheduling, they are in a queue and exit the system after making Execute_Request ( ) and Exit ( ) respectively. In that design, the scheduling concept and queuing concept are mainly focused to minimize the waiting time in the system as well as in the queue.

2.3. System flow

This section deals with the system flow that is concerned with Johnson Sequencing Algorithm followed by queuing system with finite capacity and multiple SCs. For implementing Johnson Algorithm it has been considered the followings [16]:

Consideration 1: N jobs or requests will be executed on two SCs ($SC_1$and SC_2) arranged in the sequential manner $(SC_1 \rightarrow SC_2)$.

Consideration 2: No SCs can process more than one job at a time.

Consideration 3: Each job, once started, must be performed till completion.

Consideration 4: All the jobs are in ready state, so that any one of them can be picked up for processing.

Consideration 5: Time for transfer of a job from one SC to another is negligible.

In our system design, 5 numbers of jobs and the processing time of each job are to be put in a matrix shown in Table 1. Here $T_{[i][j]}$ is the dimension of the

processing time matrix, where i and j are positive integer numbers and for Table 1, i=5 and j=2.

Table 1. Processing time matrix.

Task	Processing time on SC_1	Processing time on SC_2
Job 1	T_{11}	T_{12}
Job 2	T_{21}	T_{22}
Job 3	T_{31}	T_{32}
Job 4	T_{41}	T_{42}
Job 5	T_{51}	T_{52}

After getting this matrix, Johnson Sequencing Algorithm is implied to get the optimised sequence and there after M/M/c/K queuing model is applied to get the waiting lines as shown in Fig. 3.

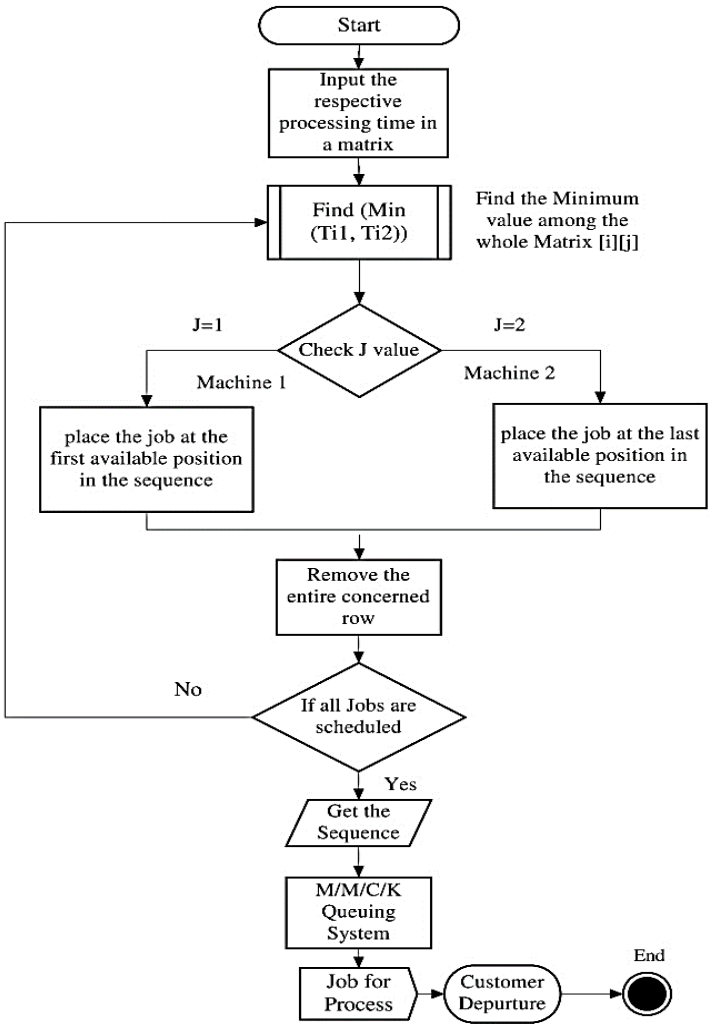

Fig. 3. Flow Chart of the system design.

2.4. Queuing model for cloud computing environment

Queuing theory is the learning of the phenomenon of the waiting line. The basic queuing process is completely described by specifying [29, 30] arrival process, service process, no of servers and the places of maximum capacity. Assuming that user requests come to the server at a certain rate with a Poisson distribution, whereas the process time for each job is supposed to be taken as exponential distribution. Considering two SCs and five places of waiting positions as capacity, it can be constructed an M/M/c/K queuing model with non-pre emptive systems. In this paper, we have merged job shop scheduling [31] with the queuing model. As per Kendal's notation [32], In case of Arrival Distribution (M), Inter-arrival times are Independent, Identically Distributed (IID) random variables with exponential distribution. In Service Distribution (M), Service times are IID and exponentially distributed.

Poisson fashion is considered as arrival pattern since arrivals of customers are based on a massive numbers of independent sources. It has been taken into consideration that page hit occurs at a certain time point zero. For modelling this distribution we need an approximate value of λ (λ= the rate parameter). Assuming that τ is the time between two successive arrivals. So that we assume τ = Inter-arrival time. We can denote $E[\tau]$ as the average or mean Inter-arrival time.

Average arrival rate $\lambda = \frac{1}{E[\tau]}$.

An exponential distribution with the rate parameter λ has density $a(t) = \lambda e^{-\lambda t}$ (t = time of customer arrival). For any given arrival time, a Poisson distribution can be established by using this formula:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \text{for } x = 0, 1, 2, \ldots$$

where $P(x)$ = Probability of x arrivals, x = number of arrivals per unit of time, and λ = Average arrival rate.

Service time is the time elapsed between the starting of the service to its completion. In case of service process, we assumed that Service times are IID and exponentially distributed. We assume S_i be the service time of i^{th} customer. So average or mean service time, denoted by $E(i)$, will be

$E(S) = \frac{\sum_{i=0}^{n} Si}{n}$ (n = number of jobs). So that service rate will be $\mu = \frac{1}{E(S)}$.

In order to enable system stability, the system will be in an equilibrium condition provided that the utilization factor $\rho = \frac{\lambda}{\mu} \leq 1$.

3. Numerical Analysis

This section deals with the numerical analysis and results. Assuming that there are 5 jobs and the respective processing time of each job has been taken in a matrix as follows:

According to the service times in Table 2, the Gantt chart (FCFS basis) has been made in accordance with the considerations shown in Fig. 4.

Table 2. Initial processing values.

Task	Processing time on SC_1	Processing time on SC_2
Job 1	0.05	0.02
Job 2	0.01	0.06
Job 3	0.09	0.07
Job 4	0.03	0.08
Job 5	0.10	0.04

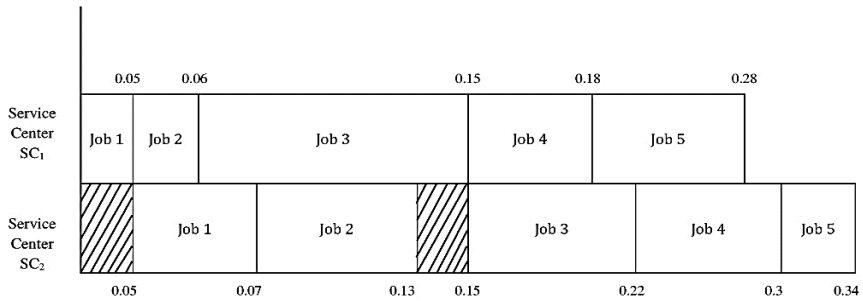

Fig. 4. Gantt chart using FCFS algorithm.

Now the service time of each process can be found using Gantt chart, and the mean service time and the average service rate may easily be calculated.

For Job 1: service time is (0.07-0) = 0.07,

Job 2: (0.13-0.05) = 0.08,

Job 3: (0.22-0.06) = 0.16,

Job 4: (0.3-0.15) = 0.15,

Job 5: (0.34-0.18) = 0.16.

Mean service time is $\frac{0.62}{5}$ = 0.124. and Service rate will be $\mu = \frac{1}{E(S)}$ = 8.0645.

While considering the Johnson Sequencing algorithm, the average service rate can be easily calculated accordingly as shown in Fig. 5.

For Job 1: service time is (0.3-0.23) = 0.07,

Job 2: (0.07-0.0) = 0.07,

Job 3: (0.22-0.04) = 0.18,

Job 4: (0.15-0.1) = 0.14,

Job 5: (0.27-0.13) = 0.14.

Mean service time is $\frac{0.60}{5}$ = 0.12 and Service rate will be $\mu = \frac{1}{E(S)}$ = 8.3333.

The results are shown in Tables 3 and 4 and the respective formulae have been appended in Appendix A. These results show that service time and average waiting time can be minimized by implementing Johnson Sequencing Algorithm and queuing system in comparison to existing FCFS Algorithm in cloud computing environment.

According to the numerical results we have discussed the comparison study regarding *Lq*, *Ls*, *Wq*, and *Ws*. Figures 6 to 9 show that the average number of customers and the average waiting time in the queue and in the system can be minimized using Johnson sequencing algorithm rather than FCFS algorithm. These comparisons produce better outcomes in case of average number of customers and the average waiting time in case of Johnson Sequencing Algorithm. This study helps the CSPs to provide better quality of service as waiting time is less and leads to customer satisfaction.

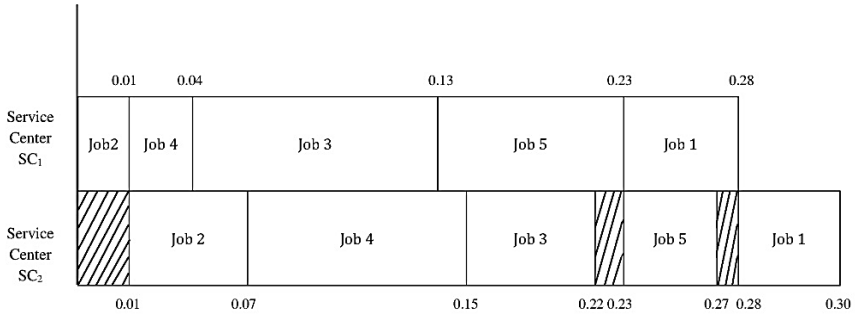

Fig. 5. Gantt chart using Johnson sequencing algorithm.

Table 3. Results with FCFS algorithm.

Johnson Algorithm				
	Lq	***Ls***	***Wq***	***Ws***
$\lambda = 2$	0.0035	0.2435	0.0017	0.1217
$\lambda = 4$	0.0280	0.5075	0.0070	0.1270
$\lambda = 7$	0.1445	0.9755	0.0209	0.1409

Table 4. Results with Johnson sequencing algorithm.

FCFS Algorithm				
	Lq	***Ls***	***Wq***	***Ws***
$\lambda = 2$	0.0038	0.2518	0.0019	0.1259
$\lambda = 4$	0.03090	0.5263	0.0077	0.1317
$\lambda = 7$	0.1585	1.0158	0.0229	0.1469

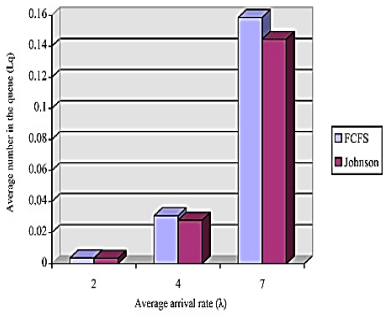

Fig. 6. Analysis of the Average number of customer in the queue (*Lq*).

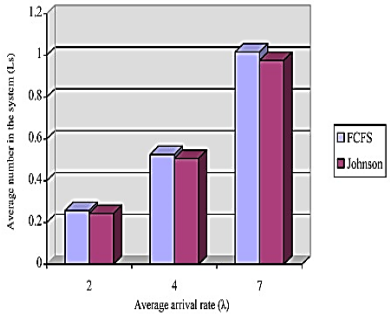

Fig. 7. Analysis of the average number of customer in the system (*Ls*).

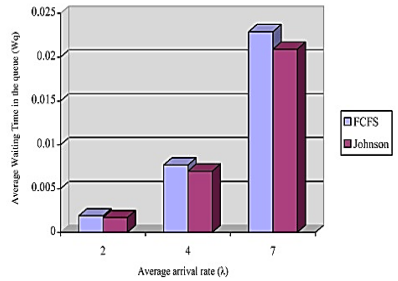

Fig. 8. Analysis of the average waiting time in the queue (*Wq*).

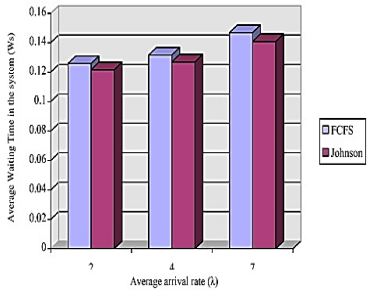

Fig. 9. Analysis of the average waiting time in the system (*Ws*).

4. Conclusion

In recent days, cloud computing is a very popular word in academia and in research. Cloud broker uses the virtualized computing resources and allocate them according to the requirement of the user based on SLA policies. In this paper, we have discussed the scheduling algorithms and queuing model with multi-server and finite capacity. We have presented that the Johnson Algorithm and queuing model can easily be used in suitable environment; so that it will produce better outcomes in waiting times in comparison to FCFS with same queuing model. We have shown in this article that using Johnson Sequencing Algorithm, an optimal sequence can be obtained and also using M/M/c/K queuing model, the waiting time and queue length can be reduced. We have also shown the comparison study. At the end of this work, the related cost per service, waiting time due to increased number of servers has been kept as the future work.

References

1. Foster, I. (2003). The grid: Computing without bounds. *Scientific American*, 288 (4), 78-85.
2. Sarathy, V.; Narayan, P.; and Mikkilineni, R. (2010). Next generation cloud computing architecture -enabling real-time dynamism for shared distributed physical infrastructure. *19th IEEE International Workshops on Enabling*

Technologies: Infrastructures for Collaborative Enterprises (WETICE'10), Larissa, Greece, *Proceedings IEEE*, 48-53.

3. Pal, S.; and Pattnaik, P.K. (2012). Efficient architectural Framework of Cloud Computing. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, 1(2), 66-73.
4. Zhao, W.; and Stankovic, J.A. (1989). Performance analysis of FCFS and improved FCFS scheduling algorithms for dynamic real-time computer systems. *Real Time Systems Symposium, Proceedings IEEE*, 156-165.
5. Frijns, R.M.W.; Adyanthaya, S.; Stuijk, S.; Voeten, J.P.M.; Geilen, M.C.W.; Schiffelers, R.R.H.; and Corporaal, H. (2014). Timing analysis of first-come first-served scheduled interval-timed directed acyclic graphs. *Design, Automation and Test in Europe Conference and Exhibition (DATE), Proceedings IEEE*, 1-6.
6. Mohapatra, S.; Mohanty, S.; and Rekha, K.S. (2013). Analysis of different variants in round Robin algorithms for load balancing in cloud computing. *International Journal of Computer Applications (IJCA)*, 69(22), 17-21.
7. Yassein, M.O.B.; Khamayseh Y.M.; and Hatamleh, A.M. (2013). Intelligent randomize round Robin for cloud computing. *International Journal of Cloud Application and Computing, ACM*, 3(1), 27-33.
8. Mishra, R.K.; Kumar, S.; and Naik, S. B. (2014). Priority based round-Robin service broker algorithm for cloud-analyst. *International Advance Computing Conference* (*IACC*), *Proceedings IEEE*, 878-881.
9. Salot, P. (2013). A survey of various scheduling algorithm in cloud computing environments. *International Journal of Research in Engineering and Technology*, 2(2), 131-135.
10. Parsa, S.; and Entezari-Maleki, R. (2009). RASA: A new task scheduling algorithm in grid environment. *World Applied Sciences Journal* (*Special Issue of Computer and IT*), 152-160.
11. Delavar, A.G.; Javanmard, M.; Shabestari, M. B.; and Talebi, M.K. (2012). RSDC (Reliable scheduling distributed in cloud computing). *International Journal of Computer Science, Engineering and Applications* (*IJCSEA*), 2(3), 1-16.
12. Dakshayini, M.: and Guruprasad. H.S. (2011). An optimal model for priority based service scheduling policy for cloud computing environment. *International Journal of Computer Applications* (*IJCA*), 32(9), 23-29.
13. El-Sayed, T.E.; El-Desoky, A.I.; Al-Rahamawy, M.F. (2012). Extended max-min scheduling using Petri net and load balancing. *International Journal of Soft Computing and Engineering* (*IJSCE*), 2(4), 198-203.
14. Ambike, S.; Bhansali, D.; Kshirsagar, J.; and Bansiwal, J. (2012). An optimistic differentiated job scheduling system for cloud computing. *International Journal of Engineering Research and Applications* (*IJERA*), 2(2), 1212-1214.
15. Cepek, O.; Okada, M.; and Vlach, M. (2002). Nonpreemptive flowshop scheduling with machine dominance. *European Journal of Operational Research*, 139(2), 245-261.
16. Johnson, S.M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1), 61-68.

17. Yang, D.L. and Chern, M.S. (2000). Two-machine flowshop group scheduling problem. *Computers and Operations Research*, 27(10), 975-985.
18. Cheng, T.C.E. (1993). Efficient implementation of Johnson's rule for the n/2/F/Fmax scheduling problem. *Computers and Industrial Engineering*, 22, 495-499.
19. Yoshida, T.; and Hitomi, K. (1979). Optimal two-stage production scheduling with setup times separated. *AIIE Transactions*, 11(1), 261-273.
20. Buyya, R.; Yeo C.S.; and Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. *The 10th IEEE International Conference on High Performance Computing and Communications, Proceedings IEEE*, 5-13.
21. Jiang, H; and Ni, T. (2009). PB-FCFS-a task scheduling algorithm based on FCFS and backfilling strategy for grid computing. *Joint Conferences on Pervasive Computing* (*JCPC 2009*), *Proceedings IEEE*, 507-510.
22. Li, L. (2009). An optimistic differentiated service job scheduling system for cloud computing service users and providers. *Third International Conference on multimedia and Ubiquitous Engineering*, *Proceedings IEEE*, 295-299.
23. Sowjanya, T.S.; Praveen, D.; Satish, K.; and Rahiman, A. (2011). The queuing theory in cloud computing to reduce the waiting time. *International Journal of Computer Science and Engineering Technology* (*IJCSET*), 1(3), 110-112.
24. Khazaei, H.; Misic, J.; and Misic, V.B. (2012). Performance analysis of cloud computing centers using M/G/m/m+r queuing systems. *IEEE Transactions on Parallel and Distributed Systems*, 23(5), 936-943.
25. Pal, S.; and Pattnaik, P.K. (2013). Classification of virtualization environment for cloud computing. *Indian Journal of Science and Technology*, 6(1), 3965-3971.
26. Pal, S.; and Pattnaik, P.K. (2015). Designing aspect and functionality issues of cloud brokering service in cloud computing environment. *Journal of Theoretical and Applied Information Technology*, 81(2),. 389-398.
27. Spicuglia, S.; Chen, L.Y.; Binder, W. (2013). Join the best queue: Reducing performance variability in heterogeneous systems. *Sixth International Conference on Cloud Computing (CLOUD 2013), Proceedings IEEE*, 139-146.
28. Guo, L.; Yan, T.; Zhao, S.; and Jiang, C. (2014). Dynamic performance optimisation for cloud computing using M/M/m queuing system. *Journal of Applied Mathematics*, 2014, 1-8.
29. Tadj, L. (1996). Waiting in line. *Potential, IEEE*, 14(5), 11-13.
30. Cheng, C; Li, J; and Wang, Y (2015). An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. *Tsinghua Science and Technology*, 20(1), 28-39.
31. Kuo, R.J; and Cheng, C. (2013). Hybrid meta-heuristic algorithm for job shop scheduling with due date time window and release time. *The International Journal of Advanced Manufacturing Technology*, 67(4), 59-71.
32. Kendall, D.G. (1951). Some problems in theory of queues. *Journal of the Royal Statistical Society* (*B*), 13(2), 151-185.

Appendix A

Waiting Lines Formulae using M/M/c/K Queuing Model

In this article, a queuing system has been used with multiple servers and maximum capacity denoted as M/M/c/K Model. c indentifies the number of servers. Sometimes the systems have a finite capacity of queue. In the system model, only maximum of K number of customers are permitted. So K is maximum capacity of the system. Therefore, (K-c) is the queue capacity.

If the queuing system is "full", the customers that arrive to the system are declined to enter into the system. That means at that time the average arrival time becomes zero. If n denotes the number of arriving customers. So we can write:

For n=0, 1, 2,…, K-1, $\lambda_n = \lambda$; and for $n \geq K$, $\lambda_n = 0$;

For steady-state probabilities are $P_n = X_n P_0$; where

For $n = 1, 2, \ldots, c$; $X_n = \dfrac{(\lambda/\mu)^n}{n!}$;

For $n = c, c+1, \ldots . K$, $X_n = \dfrac{(\lambda/\mu)^n}{c!c^{n-c}}$;

For $n > K$; $X_n = 0$.

So that we can write,

For $n = 1,2,\ldots,c$; $P_n = \dfrac{(\lambda/\mu)^n}{n!} P_0$;

For $n = c, c+1, \ldots . K$; $P_n = \dfrac{(\lambda/\mu)^n}{c!c^{n-c}} P_0$;

For $n > K$; $P_n = 0$.

where $P_0 = \left[\sum_{n=0}^{c} \dfrac{(\lambda/\mu)^n}{n!} + \dfrac{(\lambda/\mu)^c}{c!} \sum_{n=c+1}^{K} \left(\dfrac{\lambda}{c\mu} \right)^{n-c} \right]^{-1}$.

It has been denoted average number of customers in the system, average number of customers in the queue, average waiting time in the system, and average waiting time in the queue as *Ls*, *Lq*, *Ws*, and *Wq* respectively.

$$Lq = \frac{P_0 (\lambda/\mu)^c \rho}{c!(1-\rho)^2} \left[1 - \rho^{K-c} - (K-c)\rho^{K-c}(1-\rho) \right]$$

$$Ls = \sum_{n=0}^{c-1} nP_n + Lq + c\left(1 - \sum_{N=0}^{c-1} P_n \right), \text{ where } \rho = \lambda / (c\mu).$$

$$Ws = Ls / \lambda, \text{ where } \bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \lambda(1 - P_K).$$

$$Wq = Lq / \lambda.$$