Process Optimization of Big-Data Cloud Centre Using Nature Inspired Firefly Algorithm and K-Means Clustering

Jayaraj T, J. Abdul Samath

Abstract: **During the last decade, the growth of big data is immeasurable in information technology. Big data has the potential to take all the decisions necessary for a company or business. But it has many challenges as well. As its size and volume are immeasurably ample it is a very challenging task to store, process and mines it. At the same time as a boon to it cloud computing has a large capacity to store this big data and provides tremendous processing power. It is a challenging task to process large amount of data frequently in the big-data cloud center through the thousands of interconnected servers. Due to the day by day growth of the big-data, big-data cloud center is forced to improve its Quality of Service (QoS) metrics like throughput, latency and response time. Hence, to develop an optimal data processing optimization method is a current research problem that has to be solved. The major intention of this paper is to develop an application that provides maximum throughput, minimum latency and reduce the response time. Toward this, we have developed an optimization technique using nature-inspired firefly optimization algorithm and k-means clustering (FA-KMeans). The developed optimization method has been evaluated with state of art algorithms. Its experimental result elucidates that our proposed method provides good throughput, reduces latency and response time.**

***Keywords*: *Cloud computing, Multi Cloud, Firefly Optimization, Big-data,* Big-Data Cloud Centre.**

I. INTRODUCTION

Big data is a structured, unstructured and semi-structured huge voluminous data [1]. It goes on growing unaccountably. Its data size is growing from exabytes to zettabytes. For example, Facebook manages photos of 40 billion users a day [2]. The Walmart process one million customer transaction every hour. Moreover, all these are saved in a cloud memory. Its size is almost 2.5 petabytes. Also, 65000 billion photos are uploaded to Instagram every minutes, likewise per minute videos which cloud be played for 500 hours are uploaded to YouTube. In Google one billion searches are made and 294 billion emails are send every day. By 2020 the number of smartphone users will reach 6.1 billion likewise Internet of Things (connected devices) will be around 26 billion. Its data cannot be stored in the architecture of the current database management system. So, this big data with growing nature produces new challenges day by day. These challenges are reflected in store, processing and data visualization. Meanwhile, the cloud computing provides infrastructure storage and processing power needed for the big data. The architecture of cloud computing has been so designed as to suit big data.

The real scenario of cloud computing is an architecture composed of a group of powerful interconnected servers. These servers are located in different locations. Moreover, these locations are at different ranges. This architecture as a virtual resource is used to store and process different data. Its architecture is shown in fig 1.

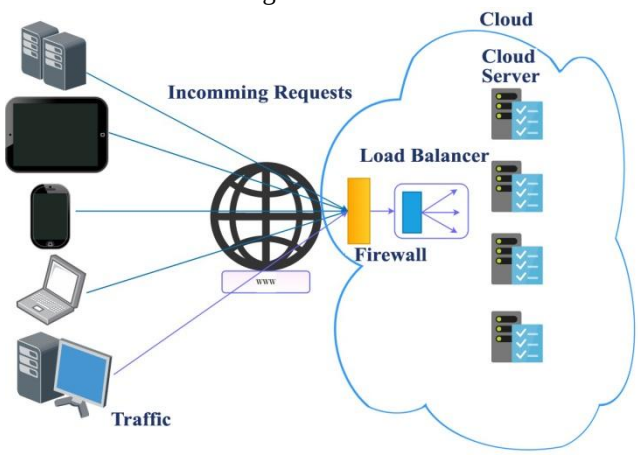

Fig1 Architecture of multi cloud environment.

There are three services in cloud computing [3][4][5]. They are infrastructure as a service (IAAS), platform as a service (PAAS) and software as a service (SAAS).

IAAS is a service model. It delivers infrastructure as a service to the end-user on demand. Generally, IAAS provides data Centre space, storage and hardware services.

SAAS is a software delivery and software licensing model. This software is provided to the cloud service user on a subscription basis on demand.

PAAS is a service provided as per the user demand. It includes Apache, Windows azure, beanstalk etc.

All those hardware and software services are virtualized and provided to the end-users as services. Its service and applications of service are described in fig 2. Still due to the enormous growth of the big data this cloud computing services face a variety of challenges.

Revised Manuscript Received on October 05, 2019.

Jayaraj T, Research Scholar, Research and development Centre, Bharathiar University, Coimbatore, India. yoursjayan@gmail.com.

Dr. J. Abdul Samath, Assistant Professor, Chikkana Government Arts and Science College, Tiruppur, India, abdul_samath@yahoo.com

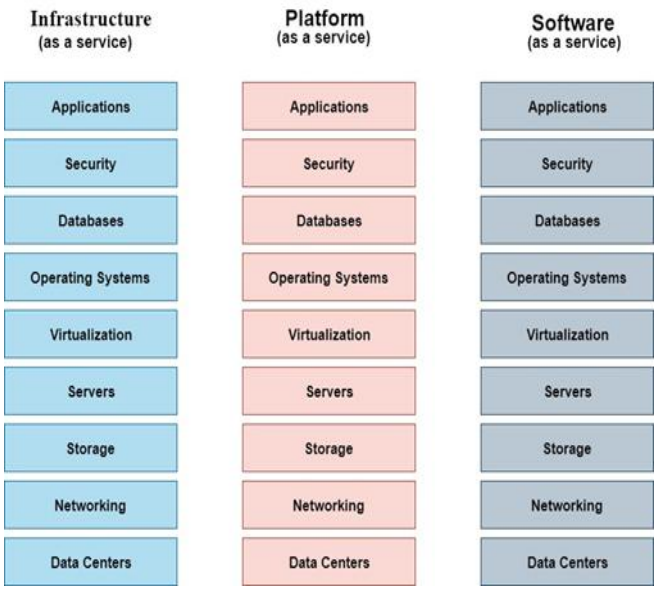

Fig2 Architecture of cloud services and applications.

When this cloud system is used for big data huge processing power is required for data processing. For the same, we unite groups of notes, process it and obtain enough processing power. Big data cloud centers consist of the following characteristics as huge transmission volume, high transmission frequency and hard transmission deadline. Hence, process scheduling is an indispensable issue in cloud computing. In order to avoid these nodes with less load and those at the shortest distance are identified and thereby the Quality of Service(QoS) metrics [6][7]: throughput and latency can be improved and decreased respectively. In the literature review, these research-related different algorithms have been reviewed. But with regard to the big data, it is a mandatory research problem that the performance of cloud computing should be improved. For the same, we have applied the nature-inspired firefly algorithms in the paper.

In this paper are dealt with in the following section is noted hereunder. Through the literature review, we have explained the already developed research papers in section II. In section III we have explained the proposed method and newly developed optimization algorithm. In section IV we have evaluated the experimental framework and implementation result. Finally, this paper has been concluded.

II. LITERATURE REVIEW

To improve cloud computing efficiency and give a better QOS many researchers have already proposed various scheduling algorithms. All these already proposed algorithms are meant only for performing the load balancing of cloud computing. In the existing methods, nothing has been notified about the balancing and optimization of big data cloud centre.

In 2015 Kumar Nishant et al[8], developed a modified Ant colony optimization for cloud and grid computing. In this approach instead of the own result updated by the ant, it is updated continuously as a single result set. This algorithm has been implemented using Java programming language. In 2015 a cloud load balancing strategy was developed by Wei-Tao [9] Wen et al using ant colony optimization algorithm for load balancing strategy. This is an optimization algorithm based on a new distributed virtual machine migration strategy. In this, the local migration agent automatically monitors resource utilization. During this monitoring section previous and current unnecessary migration. Moreover, it adopts two different types of traversing strategies for ants. It is used to find out the near-optimal virtual machine. To perform this experiment cloud SIM has been utilized.

In 2015 Jeng-Shyang Pan et al [10] proposed a load balancing system for cloud computing based on interaction artificial bee colony optimization algorithm. To simulate this, cloud SIM tools have been used. Using cloud SIM tools a various number of virtual machines have been created and experiments conducted. In 2017 Jing Yao et al [11] developed a modified artificial bee colony optimization method for load balancing in cloud computing. This improved method optimizes the amount of nectar to reach maximum throughput.

In 2012 Zhanghui Liu et al [12] proposed PSO based task scheduling for load balancing in cloud computing. In this algorithm, the standard PSO has been improved. Moreover to classify fitness value a single mutation mechanism and a self-adapting intra weight method have been introduced. To simulate this optimization process Matlab has been used and to evaluate the experiment already developed PSO algorithm has been applied. In 2016 Jigna Acharya et al [13] proposed a particle swarm optimization based load balancing method for cloud computing. This proposed algorithm is to minimize the makespan. Cloudsim 3.0.2 Tool kit is used to develop the experimental framework.

A. *Limitations in the already existing methods:*

- The authors concentrate only on improving the response time without taking any step towards reducing the processing cost.
- ACO Algorithm allocates the pre-defined tasks. Hence when the tasks higher than those are allocated are formed problem arises.
- The major limitation of the active monitoring load balancing algorithm arises when the hardware configuration changes.

III. PROPOSED METHOD

A. Firefly Algorithm

This firefly algorithm was developed by Xin-She Yang in 2007 and 2008 at Cambridge University [14][15]. This algorithm is based on behavior and flashing patterns of the firefly. In algorithm 1 its pseudo-code has been explained.

B. Characteristics of the firefly algorithm

- Firefly 'X' is attracted by the other firefly's. The firefly with less brightness is attracted by the one with more brightness. As the distance between these two fireflies increases their brightness level goes on decreasing.
- If a firefly is not brighter its movement will be random.
- The brightest firefly has random movement capability. The brightness of the firefly denotes the fitness function.
- The firefly with high brightness denotes high fitness function.

Algorithm 1 The pseudo-code of Firefly algorithm.

```
BEGIN
1. Initialize algorithm parameters;
2. MaxGen: the maximum number of generations;
3. Objective function of f(x), where x = (x1,……,xd);
4. Generate initial population of fireflies or xi(i = 1,2,….,n);
5. Define light intensity of Ii at Xi via f (xi);
        While (t < MaxGen)
                For i = 1 to n (all n fireflies)
                        For j = 1 to n (all n fireflies)
                                If (Ij > Ii)
                                        move firefly I
                                towards j;
                                        Attractiveness
                                varies with distance r via
                                Exp [- γ r2];
                                        Evaluate new
                                solution and update light
                                intensity;
                                End if
                        End for j
                End for i
        Rank the fireflies and find the current best;
        End while
6. Post process result and visualization;
End procedure;
```

C. Proposed Method

In cloud computing environment *n* number of a virtual machine are there. We denote the as V={V1,V2,…….V*n*). These virtual machines are powerful computers. Its graphical view is explained in fig3. These virtual machines are interconnected with one another. The processing cost of the virtual machine is calculated by using the following formula. We denote it as PC.

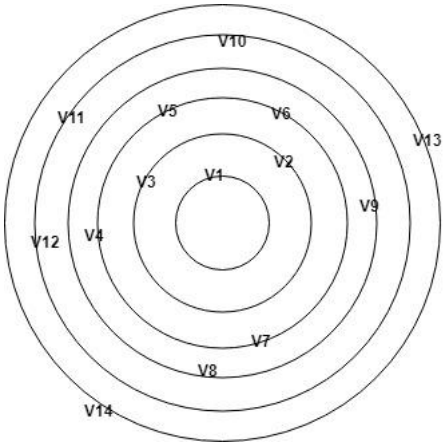

Fig3 graphical view of VM in Big Data Cloud Architecture.

$$PC=V_{CPU}+V_{MEM}+V_{BW} \quad (1)$$

- V_{CPU} =Denotes the CPU utilization of VM.
- V_{MEM} =Denotes the memory utilization of VM.
- V_{BW} = Denotes the bandwidth utilization of VM.

Random generations of Fireflies: Random location is created for all the virtual machines. This location denotes the location of actual virtual machine.

Calculating the fitness function: The fitness function calculated by using the formula 1. The brightness of the firefly is assigned based on the calculated processing cost.

Updating the location of the firefly: Based on the brightness level of the firefly each firefly is compared. Two fireflies with less light intensity are by the one with bright intensity. These three processes are repeated as many iterations.

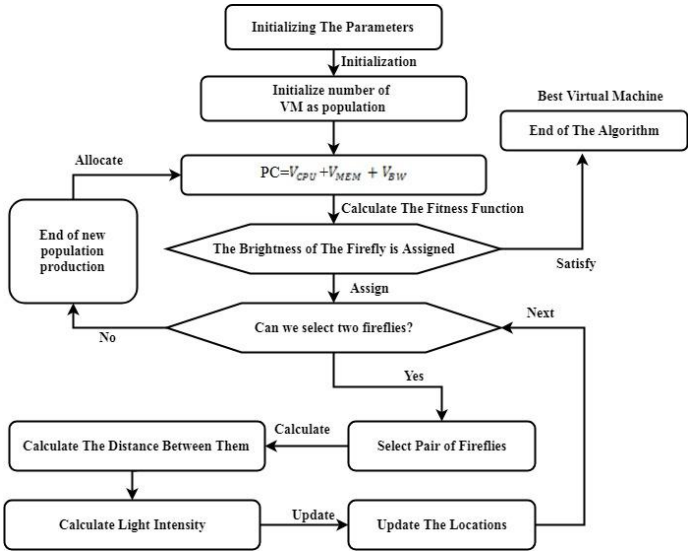

Fig 4 Process flow of proposed method

By this, its location goes on changing the number of times. At the end of every cycle, the firefly with the best light intensity (Best virtual machine) is selected and updated. At the end of each iteration each virtual machine information is stored in a hash table. There will be two values in the stored information: Its index value and its processing cast. This will be finally classified by using k-means crusting. K means clustering will classify the virtual machine in to different group based on processing cost. From this group of virtual machine can be easily found out for big-data data processing. Fig 4 show the process flow of the proposed method.

IV. EXPERIMENTAL ANALYSIS

In this section, the experiment of our proposed algorithm has been explained in detail. Moreover, the proposed method has been evaluated with state of art algorithms and their results have been compared. The proposed process scheduling method is implemented through cloud SIM and JAVA. Cloud SIM has the capability to create resources allocation and virtualized environment for developers.

The QOS is an important factor of cloud service provider. When cloud service providers provide a service to the cloud users the service level agreement is done. When this agreement is violated the cloud service providers lose their reliability and incur penalty. Hence, QOS management is an important factor of cloud management. In this experiment we have used three important QoS metrics: response time, throughput and latency. We include every metrics as six stages for the experiment. First, we provide 20 tasks to the virtual machines and evaluate its response time, throughput and latency. Next, we go on increasing the task as 40, 60, 80, 100, 120 and 140 and evaluated their response time, throughput and latency and tabulate them.

Response time: It denotes the times that the cloud service provider responds to cloud service user. It is measured as millisecond.

Throughput: It denotes how much of time is required to process a data. It is measured in MBPS.

Latency: It denotes the time required to submit a packet and its reaches the destination.

We have evaluated the proposed method with the three state of art algorithms, reviewed in this literature survey. They are Particle Swarm Optimization Based Load Balancing (PSO), Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO). In this performance analysis we have used three important metrics of QOS.

Table1 performance comparison by response times

Number of tasks	FA-KMeans	PSO[8]	ABC[11]	ACO[12]
20	21.8	30.2	32.1	34
40	22.1	31.3	33	35
80	23.4	32.5	33.7	37.2
100	23.9	33.9	33.9	39.2
120	24.2	34	34.1	41.2
140	25.1	34.2	34.8	42

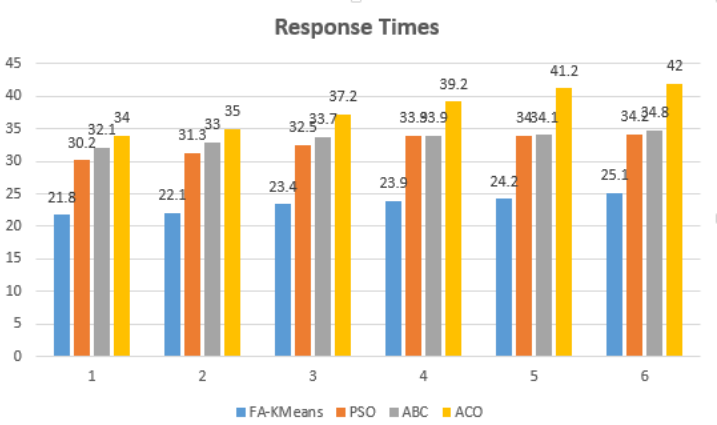

Fig 5 Response Time chart

In table.1 has been tabulated the different response times of our proposed method and of the already existing methods. Fig.5 has explained the response time variation of the proposed method and the already existing methods. From this it is elucidated that our proposed FA-KMeans method is faster than the already existing methods.

Table2 performance comparison by Throughput

Number of tasks	FA-KMeans	PSO[8]	ABC[11]	ACO[12]
20	390	240	290	220
40	382	232	280	217
80	381	236	277	213
100	375	231	262	209
120	371	231	255	205
140	362	228	250	201

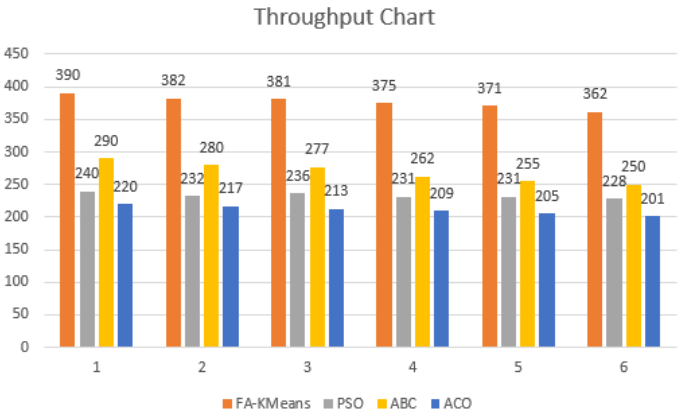

Fig 6 Throughput chart

Table.2 and fig.6 explain the throughput details of the already existing method and our proposed method. Our proposed method provides better throughput than the already existing methods.

Table3 performance comparison by latency

Number of tasks	FA-KMeans	PSO[8]	ABC[11]	ACO[12]
20	31.3	52	70	44.2
40	32.8	57	73	44.6
80	33.1	62	78	44.9
100	33.5	64	84	45.6
120	34.5	67	85	46.6
140	35.7	72	90	47.1

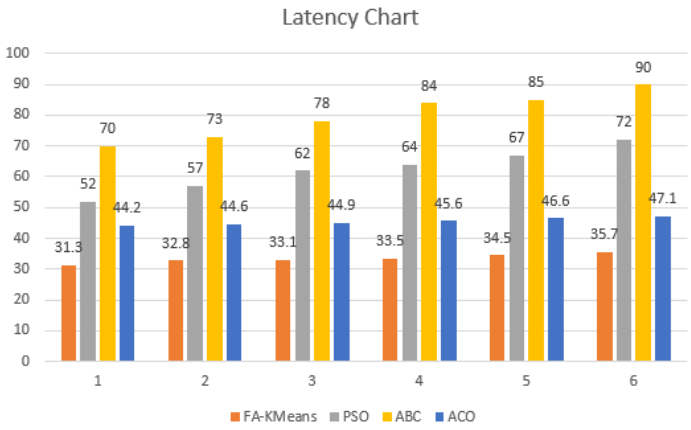

Fig7 Latency chart

Table.3 and Fig.7 shows the average latency of our proposed method and the already existing methods. The experimental results shows that our FA-KMeans based optimization method provides better latency than the already existing methods.

V. CONCLUSION

Nowadays various companies depend on big data to improve their business. Due to the characters of big data: Huge volume and heterogeneity it is a complicated task to process and get the necessary information. Cloud computing gives the entire infrastructure and the processing power to big data. Still, it is not able to give the quality of service to cloud users. In this paper using nature-inspired firefly algorithm and k-means clustering best virtual machines are selected form the cloud architecture to process big data. This proposed method reduces cloud QoS violations multiple times. It is evaluated with the state of art methods. For this experiment, three QoS metrics were used: throughput, latency and response time. The response time of FA-KMeans is in the range of 23.42 milliseconds when this value is compared with the values of the already existing methods it is very low. Moreover, average throughput is about 376.8 MBPS and latency is around 33.5 milliseconds. Experimental results show that the proposed method reduces latency, response time and improve throughput multiple times.

REFERENCES

1. K. Sekar et al, "Comparative study of encryption algorithm over big data in cloud systems", 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom).
2. Uthayasankar Sivarajah et al, "Critical analysis of Big Data challenges and analytical methods", Journal of Business Research Volume 70, January 2017, Pages 263-286.
3. Sonia Shahzadi et al, "Infrastructure as a service (IaaS): A comparative performance analysis of open-source cloud platforms", 2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), ISSN: 2378-4873.
4. Zeng et al, "The Improvement of PaaS Platform", 2010 First International Conference on Networking and Distributed Computing.
5. Aniruddha S, "Cloud computing: Software as a service", 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT).
6. Mariem Jelassi, "A survey on quality of service in cloud computing", 2017 3rd International Conference on Frontiers of Signal Processing (ICFSP).
7. Rouven Krebs et al, Metrics and techniques for quantifying performance isolation in cloud environments, Science of Computer Programming Volume 90, Part B, 15 September 2014, Pages 116-134.
8. Kumar Nishant et al, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization", 2012 UKSim 14th International Conference on Computer Modelling and Simulation.
9. Wei-Tao Wen et al, "An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment", 2015 Ninth International Conference on Frontier of Computer Science and Technology.
10. Jeng-Shyang Pan et al, "Interaction Artificial Bee Colony Based Load Balance Method in Cloud Computing", Genetic and Evolutionary Computing pp 49-57.
11. Jing Yao et al, "Load Balancing Strategy of Cloud Computing based on Artificial Bee Algorithm", 2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT).
12. Zhanghui Liu et al, "A PSO-Based Algorithm for Load Balancing in Virtual Machines of Cloud Computing Environment", International Conference in Swarm Intelligence, ICSI 2012: Advances in Swarm Intelligence pp 142-147.
13. Jigna Acharya et al, "Particle Swarm Optimization Based Load Balancing in Cloud Computing", 2016 International Conference on Communication and Electronics Systems (ICCES).
14. Xin-She Yang et al, "Firefly Algorithms for Multimodal Optimization", International Symposium on Stochastic Algorithms SAGA 2009: Stochastic Algorithms: Foundations and Applications pp 169-178.
15. T. Brenda Chandrawati et al, "A Review of Firefly Algorithms for Path Planning, Vehicle Routing and Traveling Salesman Problems", 2018 2nd International Conference on Electrical Engineering and Informatics (ICon EEI).

AUTHORS PROFILE

First Author T. Jayaraj is the Research Scholar in Bharathiar University Coimbatore. He got his PG degree M.S. Information Technology & E-commerce from Manonmaniam Sundaranar University, Thirunelveli in the year 2002. He completed M.Phil in Computer Science in 2006 at Manonmaniam Sundaranar University Thirunelveli. He has 15 years of teaching experience, presently working as assistant professor in VTM College of arts and science Arumanai, Kanyakumari. He guided many project at under graduation and post-graduation level. He participated several seminars and conferences. His area of Interest is Cloud Computing, Big Data, Data Mining and Networking.

Second Author Dr. J. Abdul Samath obtained his MCA degree from Alamelu Angappan College, Komarapalayam in the year 1999. He got his PhD degree from Gandigram Rural University Gandigram in the year 2008. He got his M.E.(CSE) from Anna University of Technology, Coimbatore in the year 2010. He has 19 years of teaching and research experience. He published 40 research papers in National and International journals and conferences and also he published 3 books. Under his guidance 6 research scholars completed PhD degree. He has the Membership in ISTE, CSI, ISCA, LIA, INNS, IAENG, etc. His area of Interest is Neural Networks, Simulation, Image Processing and Cloud Computing.