Efficient Docker Container Scheduling Using ABC Optimization Technique

Ram Krishna Verma, Ruchika Gupta

Abstract: ***In recent years the usage of virtualized technology is increasing rapidly. This makes enhancement in the performance efficiency leads to the need of the virtualization machine. This study is developed to enhance the performance level of the docker containers in cloud computing. The work presented in the paper considers the major parameters like availability, load, location, and energy of virtual machines to increase the system performance. The major objective of the work is to analyze and distribute the load of machines equally. The ABC (Artificial or Counterfeit Bee Colony) algorithm is used. For this purpose the ABC algorithm replaces the traditional ACO approach because of its various features such as simplicity, flexibility, and robustness. The output of the proposed work is evaluated in the terms of energy consumption and job completion. The observed values corresponding to these factors prove the proficiency of the suggested ABC algorithm based technique over traditional ACO algorithm based technique.***

Keywords— Cloud Computing, Container, Docker, Energy Consumption, job completion.

I. INTRODUCTION

Cloud computing has a prominent growth in the area of information technology. Due to its characteristics such as distributed computing, rapid development of virtualization and proliferation of access to the high-speed Internet, cloud technologies have become more popular. The National Institute of Technology and Standards (NITS) defines the distributed computing as a model to enable beneficial, ubiquitous on-request to network access to a shared pool of processing resources that can be configured (e.g., servers, storage, applications, networks, and services) [1]. These resources can be hastily provisioned and discharged with less board-management efforts or interaction to service provider. On the basis of different deployment models, cloud is divided into four types; namely, public clouds, private clouds, community clouds, and hybrid clouds. Enterprise software and distributed computing experiences novel languages, technologies, and platforms however after some time, many of them drop by wayside in cloud computing, Docker has brought a revolutionary change due to its different features [2, 4]. The resources are delivered as services in cloud figuring. By and large there are three kinds of service delivery models describe as, Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS) [3].

A large number of researches has been worked extensively in the domain of docker management and scheduling however only a few of them support the concept of optimization for effective job scheduling. After having a review to the traditional work, the present work aims to implement the Artificial Bee Colony (ABC) optimization algorithm(refer ABC Algorithm). ABC algorithm is motivated by the smart behaviour of Bee. ABC as an advancement kind tool gives a populations based searching system in which individual food positions are altered by the artificial bee with time and the bee's point is to find the spots of food (flower juice) sources with high Nectars (it is sweet liquid produced by flowers, which bees collect it and make them into honey) amount.

Fig. 1 illustrates that it is about behaviour of bees when they move in search of food. General Explanation: Employed bee (E_1) will start searching for food and on same time Onlooker bee has to wait for employed bee, though the food which will bring out by employed bee (E_1) from food resources that will be put away in onlooker bee for monitoring purpose. Whenever for the first time, when the employed bee (E_1) bring the food and come, at that time onlooker bee isn't having any food reference in specific order to judge the food, which comes for the first time as shown in Fig.1.

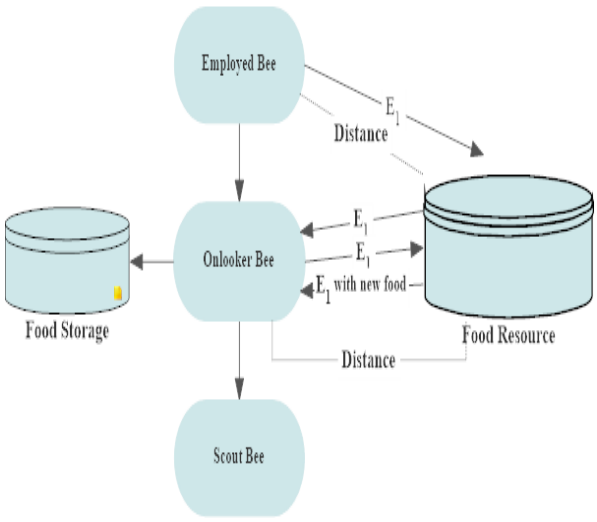


Fig.1. Architecture of Artificial Bee Colony

Revised Manuscript Received on July 10, 2019.

Ram Krishna Verma, Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India.

Ruchika Gupta, Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India.

Therefore, the employed bee will put the food as it is in food bucket (Onlooker bee). Then next, again the same employed bee will go for searching of food and again employed bee will return to the onlooker bee with new food. Now during second time, when employed bees coming to onlooker bee with new food. Now there is some quantity of food left in the food repository (from first food arrival) in order to judge the quality of food (refer Fig. 1).

Existing Algorithm 1: Artificial Bee Colony Algorithm

Begin

1. Locate Training Samples
2. Develop the initial population as
3. $x_i = 1,2,3, \ldots \ldots n$

   // with respect to the job counts in cloud
4. Evaluate the initial fitness (f_i) for initial population
5. Set j=1
6. Repeat
7. for each employed bee in the population
8. generate new solution (v_i) by
9. $x_m = l_i + rand(0,1) * (u_i - l_i)$

   // l_i and u_i is the lower and upper limit of the solution
10. Evaluate f_i(Load, Availability, Location, Energy)
11. Calculate the probability values of P_i corresponding to the solution x_i
12. $v_{mi} = x_{mi} + \emptyset_{mi}(x_{mi} - x_{ki})$
13. For each onlooker bee
14. Select a solution x_i depending on P_i
15. Produce new solution u_i
16. Calculate the value of f_i
17. Apply greedy selection process
18. If there is an abandoned solution for the scout
19. Then replace it with the new solution produced randomly
20. Store the best solution
21. j++
22. Until maximum count of j is reached

End

In the proposed work, the initial population is generated with respect to the job numbers for each iteration. The fitness function is evaluated with respect to the load, availability, location, and energy of the virtual machines. During each and every iteration, the combination of the host machine and available jobs is considered for the evaluation of fitness function. The evaluated fitness value is stored from further process. Then the combination of host machine and jobs with the maximum fitness value is passed to the docker for processing. For the rest of the remaining jobs the process repeats itself. In this manner the proposed work manages the docker container scheduling efficiently.

The rest of this paper is organized as follows. Section II reviews existing container and virtual-machine technology, their job scheduling history, and ABC algorithm for the resource related job adjusting problems. Section III exhibits the journey of docker in cloud computing, engine beyond Docker orchestration and cloud computing, and its adjustment to introduce an ABC algorithm based system. Section IV formalizes Docker scheduling problem with the help of essential preliminaries which represents all ABC-related drawbacks utilized in our scheduling algorithm (calculation) including planning of the program. Section V proposes ABC algorithm for better optimization while Section VI represents the experimental setup, about result analysis, finally section VII concludes the paper.

II. RELATED WORK

In recent years, numerous containers orchestration frameworks have emerged. An orchestration framework is essentially a container clustering's framework which enables one deal with a group of containers hosts. The orchestration dependably incorporates a sort of containers schedulers. For instance, the stand-alone Docker Swarms [2] and Google 'Kubernetes' [5] ships with their own unchanging schedulers. Scheduling algorithm for cluster's may have several purposes. For instance, some algorithms have been intended to use the clusters resource efficiently. While many others, have been intended for maximizing the application's execution. Google talked three main fundamental scheduler designs in [6]. First, the unchanging (monolithic) algorithm is schedulers with the single framework system to deal with all task placements request. The independent Docker Swarm [2] just as SwarmKit actualizes unchanging (monolithic) schedulers [7]. The second design is the level-two schedulers. In this design, it enables the clusters to be partitioned into many sub-clusters. Each sub-clusters may have its own specific schedulers. The incorporate resource supervisor acts as level-two scheduler to offer a lot of resource to every individual schedulers. Bernstein, David actualizes this design (architecture) [8]. The third one is the shared-state scheduling design architecture. A shared-state scheduler is allows to access the clusters and let the scheduler contend together freely for all. It is completely appropriated schedulers and there are neither incorporated assest (resource) allocators nor the incorporate policy implementation mechanism. Every schedulers are deciding in similar way of which the level-two scheduling algorithm system does. Google Omega is advantageous of these schedulers [6].

There are number of work effort has been assume through ABC for 'Job balancing and load scheduling' within distributed technologies. The actual algorithm which is informal in the paper is in the form of ABC, the ABC (artificial bee Colony) algorithm systems. In 2012, Anan Banharnsakun et. al. suggested to manipulate the use of ABC

techniques to optimize JSSP to determine as it is one of the big difficult Job scheduling problems. First rated by creating the potential decision of honey bee swarms within nature environment [9]. Hence their paper proposed an powerful scheduling techniques, which is Best so far ABC (artificial or counterfeit bee colony) for solving the JSSP. In 2014, Mayssa Koubaa et al, [10] used ABC algorithm method to implement load scheduling to a traditional guideline between a sea-faring recourses scheduling which is already decoded by GRASP technique by [11] and the ABC algorithm applied to this paper. Though, earlier it has been demonstrated that the ABC has ability to give the better results than any other algorithm procedure. Finally this resulted to perform better in ABC techniques using GRASP in solving the seafaring recourses scheduling problems.

In 2015, Zhao Ming et. al utilize the optimization techniques of control emergency afford scheduling problem is an approach to improve ABC by assembling together with reverse study of initialization [12]. While DABC (discrete or individual artificial bee colony algorithms). In 2016, evaluating the value of performance within DABC [13] algorithm in which by solving the combination flow shop scheduling upon minimum length criterion, which can normally produce better enduring results, which is so-far better than any other scheduling algorithms. For more simple solution to illuminate JSP with ABC [14] with an Advantage perspective against ABC algorithms system and genetic algorithms system, by combining both ABC algorithm and genetic (hereditary) algorithm, improves global-search and local-search potential of both the algorithm, Finally, the complementing algorithm upgrade the efficiencies of the algorithms effectively. Re-creation the search part which includes employ bees for food search, onlooker bees for food storage, and scout bees for hunting, As per specific know ledged problem [15]. A complete computational drive based on the 720 benchmark is an example decisive of accomplishment for the proposed 'DABC algorithms to decode the DPFSP' with the all out-stream time measure. Author gives out a DABC (discrete or individual artificial bee colony) for taking care of such an advantageous issue. Their analysis demonstrates that they introduced such algorithms which performed obviously better than the scattered searching algorithms.

III. DOCKER IN CLOUD COMPUTING

In the beginning of 2013, Docker was transferred to open source. Docker was becoming popular progressively and its popularity exploded in 2014. More than 300 million container downloads were made in amid of 2015. Since the era of the mainframe, container technology has been with it but this technology has been resurged in the last few years because of the increased virtualization, reduced emphasis on operating systems in the cloud, and establishment of Linux [16].

In docker, a facility of automating the applications is offered only when applications are deployed into Containers. An additional layer of deployment engine was added by docker on top of container (where applications are virtualized and executed) environment. Docker is designed to offer a rapid [17] environment in order to run the code effectively and furthermore, it also offers an additional facility of capable work process in order to take the code from the PC for testing reason before its generation. Figure 2 [18] depicts the major components of Docker System:

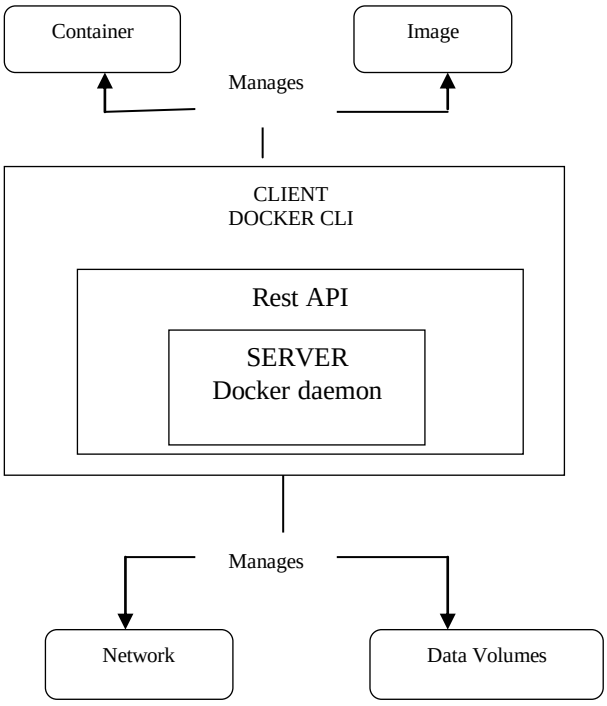


Fig.2. Generalized Docker Systems

Docker has four major internal components that also include the following:

a) Docker Images- Image can be build by using two methods. In the first method, a read-only template is used to build the image and second method is used for creating a docker file.
b) Docker Client and Server- Docker can be described as an application based on client and server. Request is sent from the docker client to the docker server and it is processed accordingly. Docker dispatches the complete RESTful (Representational state transfer) API and a command line client binary [8]. Same machine can be used to run docker daemon/server and client or a local client of docker can be associated with a remote-server (running into another machine).
c) Docker Containers- Docker container is created by docker image. Containers carry the complete kit that is necessary for an application [20], thus the application can be run in isolation. To solve the problems by using cloud computing core techniques, the major role is played by Virtualized resources.
d) Docker Registries- Docker registries are use or informal to store docker images. These registries work correspondingly to source code archives in which pictures

can be pushed or pulled from a solitary source. Public and private are two types of registries.

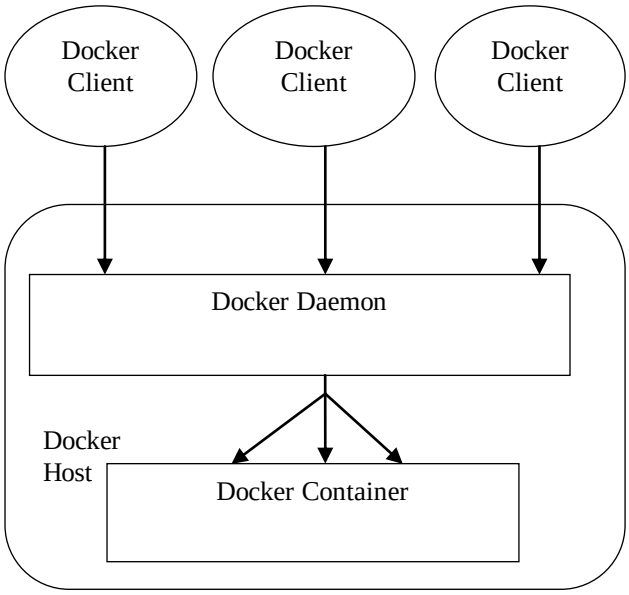


Fig.3. Architecture of Docker Container

Internal security of docker: We scrutinize the internal security of Docker on the basis of security requirements and the system and attacker model. The authors declared that an OS-level virtualization solution should meet the following requirements to stumble upon the attacks:

a) Process isolation- Preventing compromised containers from using process management interfaces for interfering with other containers is the main objective of process isolation. In Docker, processes are isolated by limiting the permissions of processes running into a container and visibility to processes running in the other containers and the underlying host and wrapping the them into namespaces.
b) File system isolation- isolation of file system is attained by protecting the file systems of the host and containers from illegal access and modification. Mount namespaces also known as the file system namespaces are used by Docker in order to isolate the file system hierarchy related to different containers [21].
c) IPC isolation- IPC is an abbreviation for Inter-Process Communication, is a set of objects used to exchange data amongst processes, such as semaphores, message queues, and shared memory segments.
d) Network isolation- Isolation in network is essential so as to prevent network-based attacks, such as ARP spoofing and Man-in-the-Middle (MitM).
e) Device isolation- **C**groups has a feature of Device White list Controller that offer a way to limit the set of devices that Docker allows a container to access.
f) Limiting of resources- In order to deal with concern of docker attack, the main components utilized by docker are **C**groups. **C**groups helps in controlling the amount of resources, such as memory, CPU, and disk I/O that can be used by any Docker container and it ensures that each container obtains its fair share of the resources. It also prevents consumption of all resources by a single container.

V. PROBLEM FORMULATION

Docker containers hold the absolute environment is needed for an application to run/compile the service container. After reviews to the traditional work in this domain, this study has finds out some of the drawbacks in expiry work. In traditional work [1], it is stated that the use of meta-heuristics algorithms, like ACO (Ant Colony Optimization techniques), is workable to upgrade the schedulers' optimally. Thus, the author had utilized the ACO algorithm system which spread the application' containers over Dockers host services for better balancing of overall reservoir employ and hence to lead in a different class of performance of system application criteria. The drawbacks of traditional system are as follows:

- Less number of parameters had been considered for evaluation and processing of the system.
- The use of ACO leads to the less efficiency in system's performance. Since the ACO is a classic optimization algorithm that is not preferred nowadays over advanced optimization algorithms.

Due to the above defined factors, there is a requirement to develop such a system that could outperform the traditional system with respect to the defined issues.

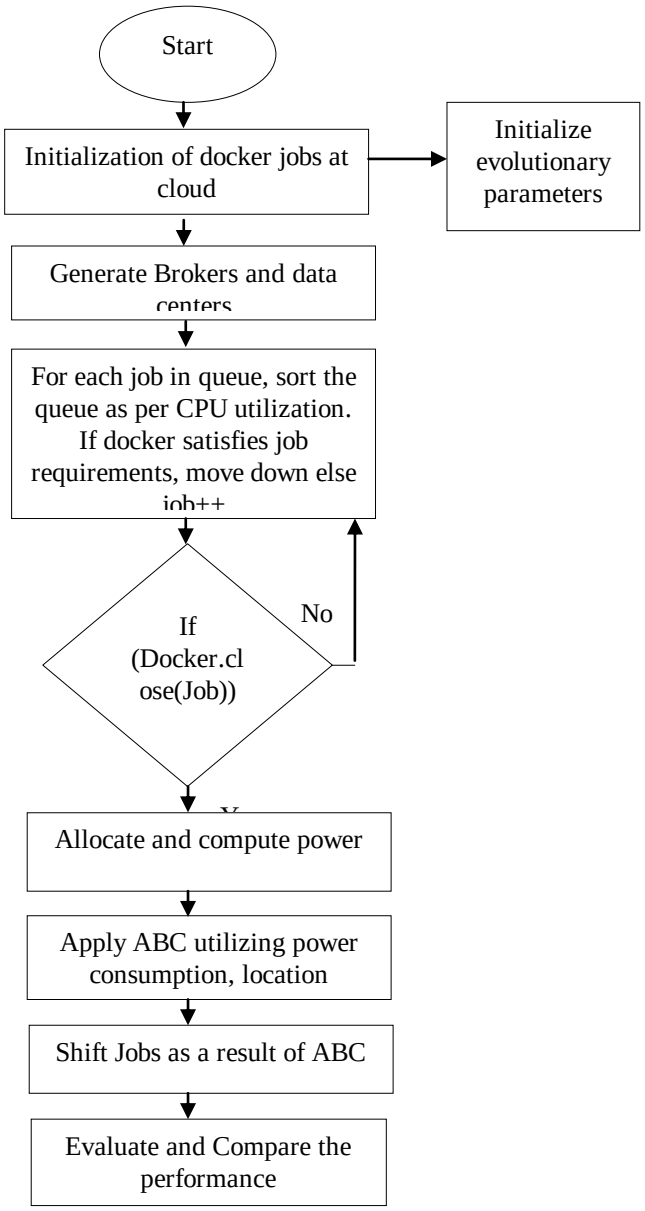
Algorithm 2: Pseudo Code

Step 1: Initialize the cloudsim package
Step 2: Initialize cloudsim library
  (i) Initialization of docker jobs at cloud
  (ii) Initialize evolutionary parameters
Step 3: Generate Brokers and data centers
Step 4: Initializing the docker properties and job properties
Step 5: Allocation of power and cloudlets generation
Step 6: Sending the job into descending order
Step7: Sort the require queue as per CPU utilization according to jobs
  (i) If docker satisfies job requirements move down else job++
Step 8: If docker close (job) goto step 5 else goto to step 7.
Step 9: Apply ABC utilizing power consumption, location
Step 10: Shift Jobs as a result of ABC
Step 11: Finally evaluating and comparing the performances

VI. PROPOSED WORK

The augmentation to utilize virtualization advancements helps the direction for a virtualization arrangement, in which can give adaptable, secure client condition, and thick. A colossal number of virtualization arrangements have developed inside the market. They can be all around characterized into two noteworthy classes: compartment (container) based virtualization strategies and hypervisor-situated in virtualization. Among these two classes, compartment (container) based virtualization strategies is skilful to give an all the more light-weight and effective virtual foundation. For the most part it enables multiple times progressively virtual conditions to destroy on a physical server, which is contrasted with hypervisor situated in virtualization. However, compartment (container) based virtualization strategies also derived with various issues as discussed in above section of this study.

The proposed work aims to eliminate the drawbacks of traditional work by developing a novel approach. In this the list of factors are enhanced by using the following parameters:



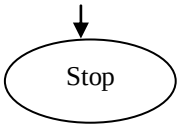


Fig.4. Structure of proposed work

a. Load
b. Availability,
c. Location, and
d. Energy

Along with this the traditional ACO is replaced by ABC optimization technique. As ACO is a meta-heuristic based calculation which is broadly separated embraced for enhancements and combinatory issues definition. There are the same numbers of works utilizing ACO to take care of couple of real issues for job scheduling virtual machines. But it is less advantageous over ABC optimization algorithm. Because ACO is a heuristic algorithm, ACO has disadvantages of defects of searching local optimization and slow convergence speed. Other than this, the ACO also lacks at:

- Theoretical analysis is very difficult.
- Sequences of random conclusion (is dependent).
- Probability administration changes by iterations.
- Experimental research is good rather than theoretical.
- Time to connect convergence undetermined.

Whereas, the advantages of ABC algorithm are as follows:

- Robustness, flexibility simplicity
- It has potential to range over local solutions.
- It also has potential to take care of objective cost.
- It has ease of implementation feature.
- Broad applicability, complex functions.

The methodology of proposed work is represented by figure 4.

VII. RESULTS ANALYSIS

In this work, the novel approach for docker management in cloud computing has been developed by replacing the traditional heuristic algorithm with ABC optimization technique. The objective of the study is to reduce the energy consumption and to enhance the count for job completion. The graph in figure 5 depicts the energy consumed by normal docker system and docker system with ABC algorithm. The evaluation for energy consumption is done on the basis of various numbers of jobs executed by the system.

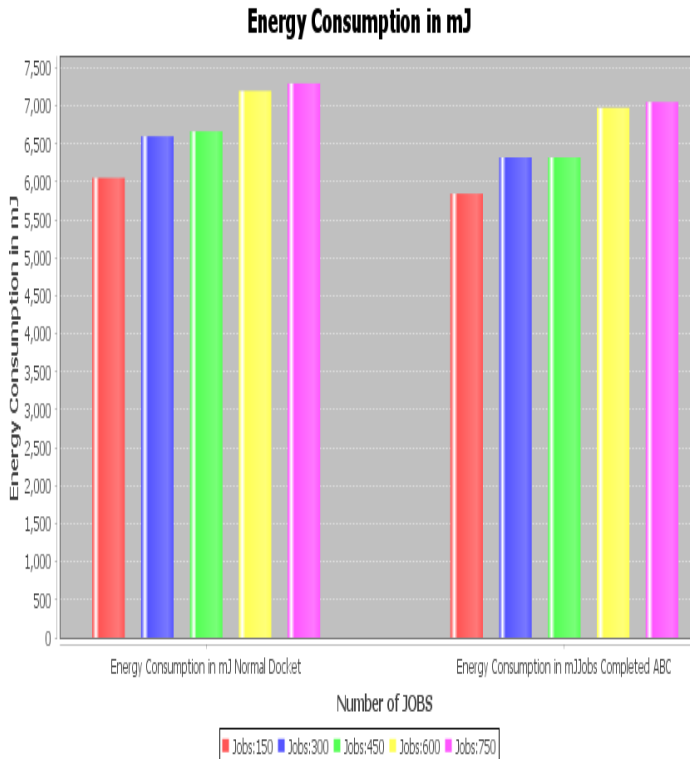


Fig.5. Comparison Analysis of energy consumption for traditional and proposed docker system

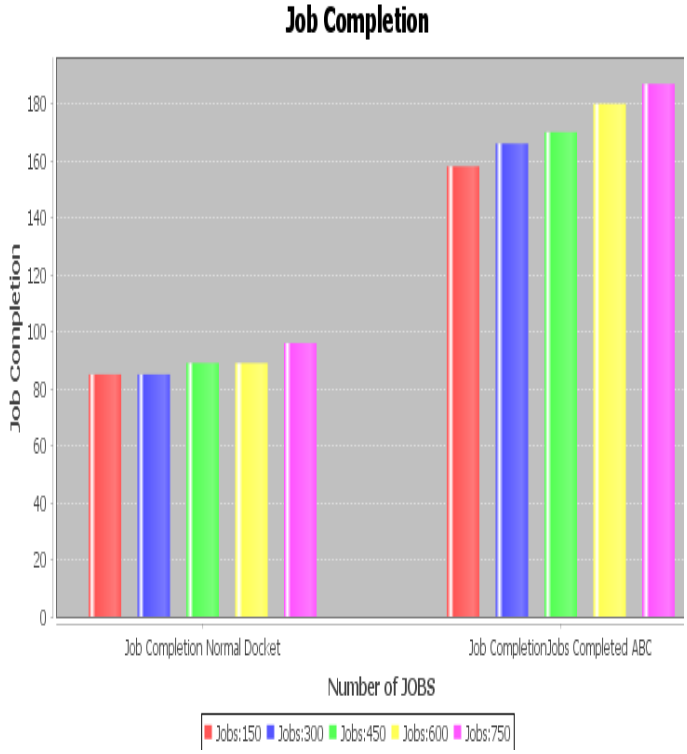


Fig.6. Comparison Analysis of Jobs completion for traditional and proposed docker system

The count of jobs in this work is shown from 150 to 750. The bar for energy level is plotted on y axis and it ranges from 0 mj to 7500 mJ with the difference of 500 mJ. The graph depicts that the energy consumption for 150 jobs in normal docker system is 6051.5768, whereas in proposed work energy consumption for the execution of similar jobs is 5845.612432. Likewise, the energy consumption for other jobs in case of proposed work is lower in comparison to the traditional docker system (refer Fig 5). The corresponding values are shown in Table 1

TABLE 1. ANALYSIS FOR ENERGY CONSUMPTION

Job Count	Energy Consumption (mJ)	
	Traditional Docker	Docker with Proposed ABC
150	6051.5768	5845.612432
300	6599.1313	6322.457248
450	6665.365	6322.972812
600	7199.6683	6974.107025
750	7295.083	7049.412628

The count for job completion by proposed and traditional docker system is shown in figure 6. The count of completed jobs is evaluated by implementing 5 iterations. The graph depicts that the traditional work is capable enough to complete the 85 jobs and in proposed work, total 158 jobs are processed out of 150 jobs with respect to the 5 iterations. The rest of the job count is shown in Table 2.

TABLE 2. ANALYSIS FOR JOBS COMPLETION

Job Count	Jobs completion	
	Traditional Docker	Docker with Proposed ABC
150	85	158
300	85	166
450	89	170
600	89	180
750	96	187

The facts shown in Table 2 are observed from graph 3. The table proves that the count for job completion in case of proposed docker system is higher and efficient than the traditional docker system. Similarly, the performance analysis of various job counts is performed and it is proved that the proposed work outperforms the traditional work.

VIII. CONCLUSION

This study develops a novel approach for managing the energy consumption and job completion in docker system. The novelty of the work is that the ABC algorithm is implemented to enhance the performance. Along with this the dependency parameters are also enhanced and improved by adding load, availability, location and energy factor. The performance analysis is done for various job counts as shown in above section. On the basis of the basis of the observed results, it is concluded that the proposed work outperforms

the traditional docker system in terms of energy consumption and job completion. The results depicts that the energy consumption of proposed work is balanced as per the count of jobs processes in 5 iterations.

In future more amendments could be done to reduce the complexity, processing time and speed of the proposed work.

ACKNOWLEDGMENT

Special thanks to Dr. Ruchika Gupta, Associate Professor, Computer Science Engineering Department, Chandigarh University for providing the research equipment's and internal foundation. This work is supported by Department of Computer Science Engineering, Chandigarh University.

REFERENCES

[1] Kaewkasi, Chanwit, and Kornrathak Chuenmuneewong. "Improvement of container scheduling for docker using ant colony optimization." *2017 9th international conference on knowledge and smart technology (KST)*. IEEE, 2017.

[2] Liu Lijuan,"Research and implementation of Docker performance service in distributed platform", *International Journal Of Engineering And Computer Science*, vol 6, issue 11, pp 23072-23076, 2017.

[3] Liu, Di, and Libin Zhao. "The research and implementation of cloud computing platform based on docker." *2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2014.

[4] Zhang, Q., Liu, L., Pu, C., Dou, Q., Wu, L. and Zhou, W., 2018. A comparative study of containers and virtual machines in big data environment. *arXiv preprint arXiv:1807.01842*.

[5] V. Vieux and et al., "Swarm: a Dcoker-native clustering system."[Online].Availabe:https://github.com/docker/swarm

[6] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek, and J. Wilkes, "Omega: flexible, scalable schedulers for large compute clusters," in *Proceedings of the 8th ACM European Conference on Computer Systems*. ACM, 2013, pp. 351–364

[7] SwarmKit contributors, "The SwamKit Project." [Online]. Available:https://github.com/docker/swarmkit.

[8] Bernstein, David. "Containers and cloud: From lxc to docker to kubernetes." *IEEE Cloud Computing* 1.3 (2014): 81-84.

[9] Banharnsakun, Anan, Booncharoen Sirinaovakul, and Tiranee Achalakul. "Job shop scheduling with the best-so-far ABC." *Engineering Applications of Artificial Intelligence* 25.3 (2012): 583-593.

[10] Koubaa, Mayssa, Sonda Elloumi, and Souhail Dhouib. "Artificial Bee Colony to solve seafaring staff scheduling problem: A real case." *2014 International Conference on Advanced Logistics and Transport (ICALT)*. IEEE, 2014.

[11] Ammar, Mohamed Haykal, Mounir Benaissa, and Habib Chabchoub. "GRASP for seafaring staff scheduling: Real case." *2013 International conference on advanced logistics and transport*. IEEE, 2013.

[12] Zhao Ming, Song Xiao- Yu and Gao Yi-Chen, "Emergency Scheduling Optimization Based on Improved Artificial Bee Colony" *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)* DOI:10.1109/ICSESS.2015.7339196. 2015 IEEE.

[13] Liu, Y., Ouyang, D., Gu, W. and Wang, L., 2016, December. A Discrete Artificial Bee Colony Algorithm for Permutation Flow Shop Scheduling. In *2016 9th International Symposium on Computational Intelligence and Design (ISCID)* (Vol. 2, pp. 161-164). IEEE.

[14] Lvshan, Y., Dongzhi, Y. and Weiyu, Y., 2017, November. Artificial bee colony algorithm with genetic algorithm for job shop scheduling problem. In *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (pp. 433-438). IEEE.

[15] Pan, Jia-Qi, Wen-Qiang Zou, and Jun-Hua Duan. "A Discrete Artificial Bee Colony for Distributed Permutation Flowshop Scheduling Problem with Total Flow Time Minimization." In *2018 37th Chinese Control Conference (CCC)*, pp. 8379-8383. IEEE, 2018.

[16] Rad, Babak Bashari, Harrison John Bhatti, and Mohammad Ahmadi. "An introduction to docker and analysis of its performance." *International Journal of Computer Science and Network Security (IJCSNS)* 17.3 (2017): 228.

[17] Wang, Zuocheng, Lixia Xue, and Yanli Luo. "Cloud Computing Platform Based on the Docker." *DEStech Transactions on Computer Science and Engineering* icmsie (2016).

[18] de Alfonso, Carlos, Amanda Calatrava, and Germán Moltó. "Container-based virtual elastic clusters." *Journal of Systems and Software* 127 (2017): 1-11.

[19] Blum, Christian. "Ant colony optimization: Introduction and recent trends." *Physics of Life reviews* 2.4 (2005): 353-373.

[20] Bui, Thanh. "Analysis of docker security." *arXiv preprint arXiv:1501.02967* (2015).

[21] Miao Liyao, Chen Lijun, "A cluster expansion method based on Docker container," *Computer Applications and Software*, pp. 34-39, 2017.

AUTHORS PROFILE

Ram Krishna Verma is a full-time research scholar at the Computer Science Engineering Department, Chandigarh University, Punjab, India, 2017-19. He received his B. Tech from the Computer Science Engineering Department, Jawaharlal Nehru Technological University, Hyderabad, India, 2013-17 His research interests include Cloud Computing, Machine Learning, Natural Language Processing and Light Weight Technology (Docker Engine).

Ruchika Gupta is an associate professor at the Computer Science Engineering Department, Chandigarh University, Punjab, India. She received her Ph.D. from the Computer Engineering Department, National Institute of Technology, Surat, India. Her research interests include Location Privacy, Spatial Anonymity, Data Security, Mobile Computing, and Machine Learning.

.