International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 07 | July - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Container Scheduling Based On Cuckoo Search Algorithm

Niral Jadav[1], Bakul Panchal [2], Madhuri Patel [3]

[1, 2, 3] Department of Information Technology, L.D College of Engineering, Ahmedabad, Gujarat, India

Abstract - Nowadays, Virtualization plays a significant role in the field of cloud computing. Similar to virtual machine scheduling, containers scheduling is also one of the emerging problems in the cloud environment. While container scheduling problem is an NP-hard problem, there is no polynomial time complexity to find optimal schedule for such a large size problems. Meta-heuristics are a popular class of population based optimization algorithms inspired by intelligent process and behavior arising in nature. These are widely used to solve such kind of optimization problems. Aiming at to perform container scheduling inspired by the parasitic behavior of cuckoo bird. In this paper we have proposed container scheduling with the help of cuckoo search algorithm.

Keywords - Container Scheduling, Meta-heuristics algorithms, cuckoo search algorithm

1. Introduction

Cloud computing has become one of the attractive technology in the field of both ICT (Information and Communication Technology) trade and academic research. With the help of cloud computing users can access provision, process, store and network important computer resources, operating system, virtual desktops, web services, development platform and database. It also uses specific application as a service offered by cloud computing providers such as a "utility" on "pay as you go". [1] There are many technologies used in cloud environment one of them is virtualization technology. Virtualization allows multiple applications to run separately on same physical hardware. Thought virtualization has its own advantages, it faces some limitations like excessive time utilization, long start up and long shut down, complicated scheduling migration process and so on. These problems leads to new the emergence of a new process called containerization that further lead to virtualization at OS level, virtualization brings. Where in case of container centric environment, there is physical infrastructure over host operating system is running and container engine creates multiple containers and running it without installing any guest operating system separately. This is because containers are packages of software that contains all the necessary information that are required to run. Its startup time is fast as compare to virtual machine and provides quick provisioning.

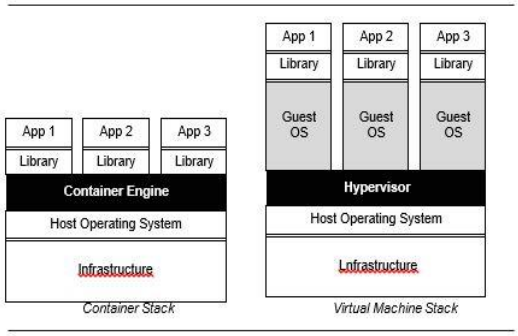

Fig. 1. Containerization v.s. Virtualization environment.[5]

Scheduling container just like virtual machine scheduling can perform different form depending on the underlying technology. For example depending on the implementation details the incoming tasks(T_i) from users can be scheduled directly on container (C_j) running on physical machine (PM) or on virtual machine (VM) running on the physical hardware. And it is responsibility of scheduler to choose scheduling algorithms by taking various performance metrics such as utilization, makespan, power etc…[6] In the following Fig.2 shows generalized block diagram of container scheduling.

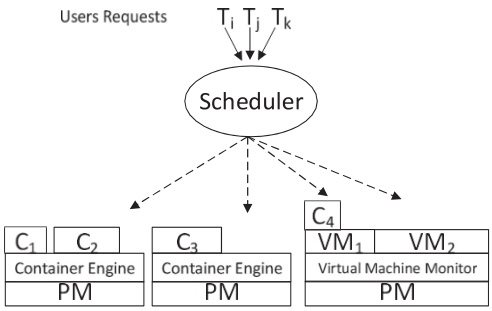

Fig. 2 Scheduler block diagram[6]

Scheduling container in the field of cloud computing is an NP Hard problem so that there are no polynomial complexity algorithm to find optimal schedule for such a large size problem. Meta Heuristics are popular class of population based optimization algorithms there are widely used to solve the problem of optimization in many discipline. We used one of the meta-heuristics approach inspired by cuckoo search algorithm to solve the problem of container scheduling.

The remainder of this paper organized as follows: The related work discusses literature review in Section 2. The cuckoo inspired container scheduling discusses in Section 3. Section 4 discusses experimental evaluation of proposed methodology. Discussion is available in Section 5. Then Section 6 concludes the paper.

2. Related work

A. Literature Review

[5]proposed the first ant colony algorithm of container scheduling for docke r with the goal to enhance resource utilization and provide load balancing. Docker swarm is used for this algorithm in scheduling containers. They used fixed set of meta-heuristics parameters, the result of adjusting is to be improved.[7]proposed placement architecture for container as a service in cloud environment with a goal to improve resource utilization in terms of number of CPU cores and memory size for both VMs and PMs, and to minimize the number of instantiated VMs and active PMs in a cloud environment. They use Ant Colony Optimization based on Best Fit (ACO-BF) with the proposed fitness function, the proposed ACO-BF placement algorithm outperforms the BF and MF heuristics and the resource utilization of both VMs and PMs provides significant improvement. To achieve resource utilization the algorithm requires lots of computation and its solution take a long time.[8]proposed a new container scheduling algorithm based on multi objective optimization (multiopt) with a goal to overcome the default scheduling of docker swarm. Multiopt increases the maximum TPS by 7% and reduces the average response time per request by 7.5% while consuming roughly same allocation time. However, fault tolerance of containers not considered.[9]proposed multi objective container placement ant colony optimization algorithm that modify existing ant colony algorithm and used additional objective network usage and cost. MOCP-ACO performed better in reducing VM cost but is only slightly better in reducing network traffic.[10]proposed particle swarm optimization based container scheduling for docker platform which have solved the problem of insufficient resource utilization and load balancing problem. The performance of the PSO is improved 20% and 19% higher than the spread and improved random respectively under the same host configuration.[11]proposed the placement of containers under a nonlinear energy consumption model. They propose an improved genetic algorithm called IGA for efficiently searching the optimal CP solution by introducing two different exchange mutation operations and constructing a function as the control parameter that is selectively control the usage of the two operations. However they only considers a single optimization objective, i.e., energy efficiency.[12]proposed a Two-stage Multi-type Particle Swarm Optimization approach, named TMPSO, to energy-aware container consolidation in Cloud data centers. The experimental results show that proposed PSO based algorithm is able to provide solutions in a more efficient and effective way, in particular processing the requirement of massive applications.[13]proposed to optimize container allocation and elasticity management, a genetic algorithm approach using the Non-dominated Sorting Genetic Algorithm-II (NSGA-II). Their optimization algorithm enhances system provisioning, system performance, system failure, and network overhead.[14]proposed a new energy-efficient container-based scheduling (EECS) strategy for processing various types of IoT and non-IoT based tasks with quick succession. for finding a suitable container for each task with minimum delay the proposed method use accelerated particle swarm optimization (APSO) technique.[15] To simulate the CloudSim scheduling framework and simulate the kube-scheduler model scheduling framework combines the improved ant colony algorithm and the adaptive particle

swarm algorithm in the meta heuristic algorithm. The algorithm is used to improve the allocation scheme of the kubescheduler model, and reduce the resource cost and the maximum load.

B. Cuckoo Search Algorithm

Preparing a nest, provisioning food, and defending offspring are natural behavior of birds. There are many kinds of bird utilize parasitic brooding by exploiting resources allocated by other birds. Generally cuckoo birds are brood parasites; for their own eggs they exploit the nests of other birds. When the female cuckoo is ready to lay its egg, it searches for a different host nest for its egg. This strategy is used as some host species may discover the cuckoo egg in their nest and either abandon their nest or throw this egg away. Moreover, the cuckoo needs its hatchlings to get all the resources of the other bird. By ensuring that at least some of its egg avoid detecting by the host bird, the cuckoo lays its eggs in nests of birds whose eggs closely resemble those of the cuckoo.[16]

C. Levy Flights Mechanism

Generally animals search for finding food in a random or quasi-random manner in the nature. The foraging path of an animal is a random walk because the next move is based on both the current location/state and the transition probability to the next location. The chosen directions probability modelled mathematically. Various studies had shown that the flight behaviour of many animals and insects demonstrates the typical characteristics of Lévy flights. A Lévy flight is a random walk in which the step-lengths are calculated according to a heavy-tailed probability distribution. The distance from the origin of the random walk tends to a stable distribution after a large number of steps.[17]

1. Cuckoo Inspired Container Scheduling

Cuckoo Search algorithm is a meta-heuristic approach inspired by brood parasitic behavior of cuckoo bird. In the scheduling of containers we used this approach by initializing cuckoo parameters and generating initial population. The algorithm starts by calculating initial best position and initial best fitness of container from available container list. To generate a new solution from available container list, it takes particular container let say i^{th} container and call cuckoo algorithm to calculate functionality of it and after that sorting is performed to keep the best solution. If the particular container is less fitted then in this case this one is abandon and again calculates the fitness value of container and sort accordingly. The solution is given by

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus Lévy\ (u) \quad (1)$$

and

$$\text{Lévy } u = t^{-\lambda}, \quad 1 < \lambda \leq 3 \quad (2)$$

Where α is a step size, λ indicates parameter for levy flight. In the following Fig.3 shows the main idea of proposed work.

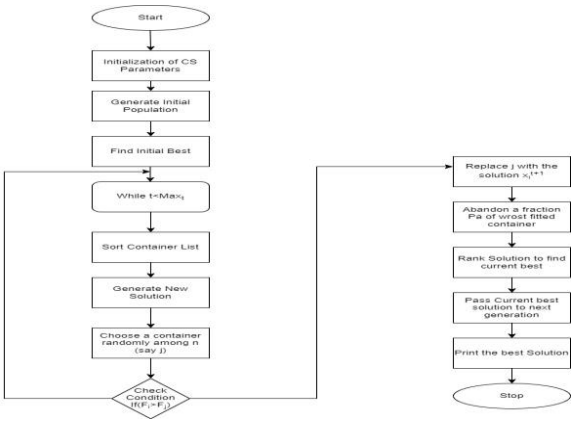

Fig.3 Flow Chart of Proposed Work

Proposed Algorithm

1. Set the initial value of the container size n, probability $P_a \in (0,1)$ and maximum number of iteration Max_t.
2. Set t: = 0. {counter initialization}
3. For (i=1 : i≤ n) do
4. Generate initial population of n container x_i^{t+1}.
5. Evaluate fitness function $f(x_i^t)$.
6. End For.
7. Generate a new solution x_i^{t+1} randomly by Levy Flight.
8. Evaluate fitness function x_i^{t+1} i.e., $f(x_i^{t+1})$.
9. Choose a container x_j among n solution randomly.
10. If ($f(x_i^{t+1}) > f(x_j^t)$) then
11. Replace the solution x_j with the solution x_i^{t+1}.
12. End if.
13. Abandon a fraction Pa of worst container.

14. Place container at new location using Levy flight a fraction Pa of worse container.
15. Keep the best solution.
16. Rank the solution and find current best solution.
17. Set t = t+1;
18. Until ($t \geq Max_t$)
19. Produce the best solution.

3. Experiment

This proposed method is implemented using python. Our experiment run on 64 bit operating system. The following table shows parameters of cuckoo based container scheduling.

Parameters	Value
Population Size	50
Max Domain	500
Min Domain	-500
Lambda	1.5
Pa	0.25
Step Size	0.01
Dimension	10
Trial	3
Iteration	30

Table 1 Parameters of proposed work.

The simplicity of the cuckoo algorithm is it requires less computation as compared to another meta-heuristic method and solves optimization problems in an efficient time. The following Fig.4 shows visual form of how containers are place using fitness values during the number of iteration when program is executed. We have performed it on three trails. In each trails there are thirty iterations on which it randomly takes container and calculates its fitness and scheduling of container is taken place if it is best fitted.

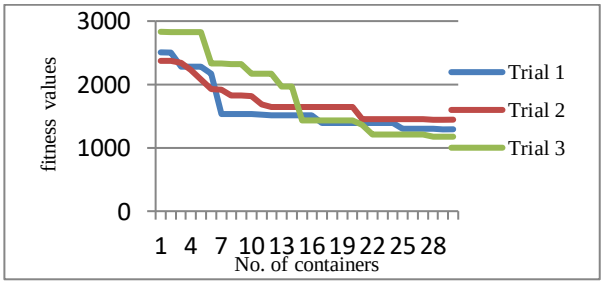

Fig.4 Container scheduling using cuckoo algorithm.

4. Discussion

In terms of scheduling a container we proposed with the help of cuckoo algorithm whereas scheduling using ant colony algorithm is available in case of container scheduling. Thought, Ant colony algorithm works well in container scheduling, it takes large amount of time to schedule particular container. We have performed our experiment in thirty containers taken it random and as per best fitness values it placement is performed. In the following figure 5 shows time required to schedule a container using our proposed approach. While, in figure 6 shows time cost using existing approach i.e using aco.

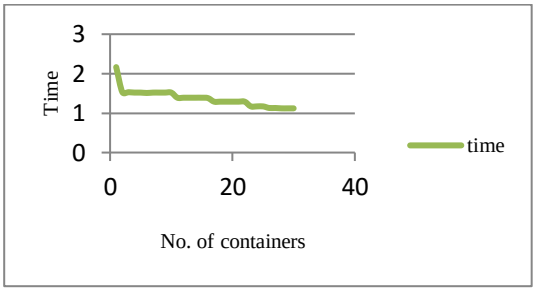

Fig. 5 time cost of proposed approach

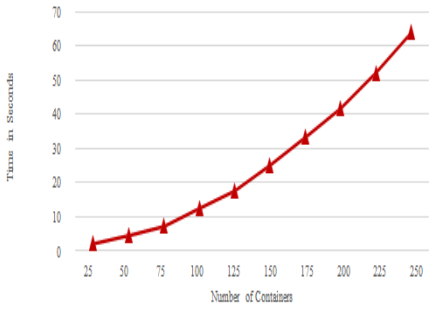

Fig. 6 time cost of existing approach(aco)[7]

After analyzing it is found that our proposed approach gives approximately 1.368 average response time in terms of scheduling.

5. Conclusion

Container scheduling nowadays is one of the hot topic of researcher's interest in the field of cloud computing. Scheduling of containers is possible by many techniques in which meta-heuristics approach is one of them. Meta-heuristics approaches are inspired by nature and it can be used to solve NP hard problems like scheduling of container. As per literature study it is found that cuckoo search algorithm requires minimal time to execute the tasks in terms of job scheduling. Hence it becomes our research direction to test the performance of scheduling using cuckoo search algorithm of meta-heuristics approach. We have performed cuckoo inspired container scheduling and the simplicity of this approach is that it requires less computation as compare to ant colony algorithm. The scheduling of containers is performed as per the best fitness value. Our proposed approach gives approximately 1.368 average response time in terms of scheduling.

References

[1] S. H. H. Madni, M. S. Abd Latiff, M. Abdullahi, S. M. Abdulhamid, and M. J. Usman, "Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment," *PLoS One*, vol. 12, no. 5, pp. 1–26, 2017, doi: 10.1371/journal.pone.0176321.

[2] A. M. Potdar, D. G. Narayan, S. Kengond, and M. M. Mulla, "Performance Evaluation of Docker Container and Virtual Machine," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1419–1428, 2020, doi: 10.1016/j.procs.2020.04.152.

[3] N. Zhou *et al.*, "Container orchestration on HPC systems through Kubernetes," *J. Cloud Comput.*, vol. 10, no. 1, 2021, doi: 10.1186/s13677-021-00231-z.

[4] J. Watada, A. Roy, R. Kadikar, H. Pham, and B. Xu, "Emerging Trends, Techniques and Open Issues of Containerization: A Review," *IEEE Access*, vol. 7, pp. 152443–152472, 2019, doi: 10.1109/ACCESS.2019.2945930.

[5] C. Kaewkasi and K. Chuenmuneewong, "Improvement of container scheduling for Docker using Ant Colony Optimization," *2017 9th Int. Conf. Knowl. Smart Technol. Crunching Inf. Everything, KST 2017*, pp. 254–259, 2017, doi: 10.1109/KST.2017.7886112.

[6] I. Ahmad, M. G. AlFailakawi, A. AlMutawa, and L. Alsalman, "Container scheduling techniques: A Survey and assessment," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.03.002.

[7] M. K. Hussein, M. H. Mousa, and M. A. Alqarni, "A placement architecture for a container as a service (CaaS) in a cloud environment," *J. Cloud Comput.*, vol. 8, no. 1, pp. 1–15, 2019, doi: 10.1186/s13677-019-0131-1.

[8] B. Liu, P. Li, W. Lin, N. Shu, Y. Li, and V. Chang, "A new container scheduling algorithm based on multi-objective optimization," *Soft Comput.*, vol. 22, no. 23, pp. 7741–7752, 2018, doi: 10.1007/s00500-018-3403-7.

[9] B. Burvall, "Improvement of Container Placement Using Multi-Objective Ant Colony Optimization," 2019.

[10] L. Li, J. Chen, and W. Yan, "A particle swarm optimization-based container scheduling algorithm of docker platform," *ACM Int. Conf. Proceeding Ser.*, pp. 12–17, 2018, doi: 10.1145/3290420.3290432.

[11] R. Zhang, Y. Chen, B. Dong, F. Tian, and Q. Zheng, "A Genetic Algorithm-Based Energy-Efficient Container Placement Strategy in CaaS," *IEEE Access*, vol. 7, pp. 121360–121373, 2019, doi: 10.1109/ACCESS.2019.2937553.

[12] T. Shi, H. Ma, and G. Chen, "Energy-Aware Container Consolidation Based on PSO in Cloud Data Centers," *2018 IEEE Congr. Evol. Comput. CEC 2018 - Proc.*, pp. 1–8, 2018, doi: 10.1109/CEC.2018.8477708.

[13] C. Guerrero, I. Lera, and C. Juiz, "Genetic algorithm for multi-objective optimization of container allocation in cloud architecture," *J. Grid Comput.*, vol. 16, no. 1, pp. 113–135, 2017, doi: 10.1007/s10723-017-9419-x.

[14] M. Adhikari and S. N. Srirama, "Multi-objective accelerated particle swarm optimization with a container-based scheduling for Internet-of-Things in cloud environment," *J. Netw. Comput. Appl.*, vol. 137, no. March, pp. 35–61, 2019, doi: 10.1016/j.jnca.2019.04.003.

[15] Z. Wei-guo, M. Xi-lin, and Z. Jin-zhong, "Research on kubernetes' resource scheduling scheme," *ACM Int. Conf. Proceeding Ser.*, pp. 144–148, 2018, doi: 10.1145/3290480.3290507.

[16] E. Aloboud and H. Kurdi, "Cuckoo-inspired job scheduling algorithm for cloud computing," *Procedia Comput. Sci.*, vol. 151, no. 2018, pp. 1078–1083, 2019, doi: 10.1016/j.procs.2019.04.153.

[17] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo Search Optimization- A Review," *Mater. Today Proc.*, vol. 4, no. 8, pp. 7262–7269, 2017, doi: 10.1016/j.matpr.2017.07.055.