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

Meta-heuristics Approach for container scheduling: A Review

Niral Jadav[1], Madhuri Patel[2]

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

ABSTRACT- Container Scheduling problem is an emerging problem in the field of cloud computing. Scheduling a Container can be performed using many techniques, in which meta-heuristic is one of them. In this paper we provide literature review regarding meta-heuristics approach used to solve container scheduling problems in cloud environment.

Keywords: Virtualization, Container scheduling, Meta-heuristic approach.

1. INTRODUCTION

Cloud computing play a significant role in the era of information technology. There are two important words 'cloud' and 'computing' combine with each other provides on demand services as per consumer need. Virtualization in cloud computing is one of the important technology. Virtualization makes cloud computing becomes promising technology and enables users to access services, application from anywhere anytime just providing on-demand availability of resources. With the help of cloud virtualization technology one can achieve elasticity of large scale shared resources. Virtual machine provide hardware level virtualization and enables users to install guest operating system separately on each virtual machine and to run application and services on it.

However virtualization provides virtual environment, containerization is an advance form of virtualization technology. It provides running of multiple containers running separately on host operating system and enables operating system virtualization. Containers are nothing but light-weight virtual machine and can run fast as compare to virtual machine. Container contains packages of software and having all necessary information that are require to run in any environment.

A Docker consist layered file system similar to linux virtualization stake as shown in fig 1 (right). To add a writable file system on top of the read-only file system Docker uses a union mount. This allows multiple read-only file systems to be stacked on top of each other. To create new images by building on top of base images this property can be used. Only the top layer is writable, which is the container itself.[1] Containerization facilitates single applications in containers to clusters of container hosts that can run containerized applications across cluster hosts. As shown in Fig.1(left) individual containers are grouped together to form a group of cluster. Number of cluster consist several host. There are link associated with two or more container to connect them with each other. Now a days Docker becomes the most popular container providing solution in the containerization environment.

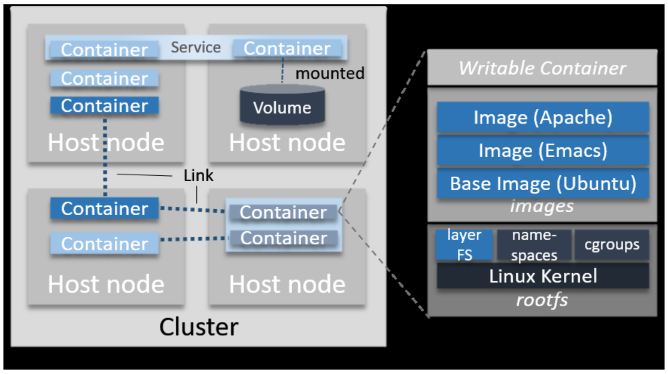

Fig 1. Container Cluster Architecture.[1]

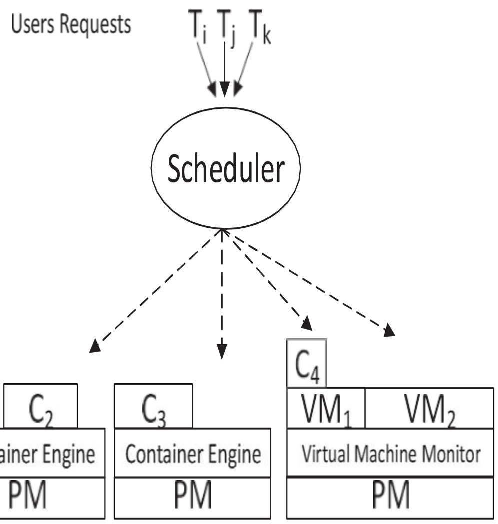

Fig 2. Generalized block diagram of container scheduling.[2]

In order to understand container scheduling better way here we can understand how containers and virtual machine differs.

Containers V.S. Virtual Machines: [3]

The two virtualization technologies of virtual machine (VM) and docker container each have their own specializations and benefits. Understanding their features and limitations can help us improve the performance of applications running in these virtualization environments. The recent advancement of docker is different from traditional VMs. First, the VM hypervisor manages resources among different virtual machines in a distributed mode, where each VM is assigned the maximum limit of resources it can use. Unlike VM, containerized virtualization performs centralized resource management among different containers.

Depending on the underlying technology Container scheduling can take on different forms. For example, Fig 2. a generalized block diagram of containers scheduling is shown. Depending on implementation details, the incoming tasks (Ti) from users can be scheduled directly on container (Cj) running on physical machine (PM) or on virtual machine (VM) running on the physical hardware.[2] As per requirement it is the responsibility of the scheduler to identify the best placement of the incoming tasks and to find the best possible schedule by taking various performance metrics like utilization, makespan, power, etc.

Resources like processing unit, memory and page cache are shared among all containers. No prior resource allocations are done and all containers compete for the shared resources at runtime. As a result, resource contention and performance interference become more severe when using Docker containers than using VMs. Thus, it is more urgent and necessary to have an effective scheduling algorithm to launch Docker containers in order to mitigate such performance interference and improve resource utilization in the Docker container environment. Second, each VM has their own VM image that consists of guest OS and libraries. This type of virtualization results in better security and isolation, but causes higher startup overhead. Thus, a casual practice is to launch all VMs at the beginning. On the other side, as Docker containers are lightweight without guest OS, they can be launched when needed. This thus gives us a good opportunity to dynamically schedule and launch Docker containers at runtime.

In the field of cloud computing, optimization problems are mostly considered to be single objective, but in reality, these are typically multi-objective in nature. For instance, every cloud user wishes to get cloud services with minimum cost and high performance. On the other hand, cloud providers want to provide cloud services with maximum utilization of resources, high revenue and profit.[4] This is a way that the algorithms need to resolve the multi-objective problematic issues in a different way from algorithms for single-objective optimizations. Meta-heuristic optimization algorithm shows their advantages and benefits in dealing with multiple-objective functions.

Scheduling is one of the important challenges in cloud computing because there are number of virtual machine (VM) requests and the pool of physical resources. Job scheduling is considered as an NP- hard problem that uses heuristic algorithms to approximate solution. Scheduling in the cloud can occur in a static mode, where all jobs are submitted at once and resource allocation is pre-computed before system execution. In a dynamic mode, the arrival of jobs is unpredictable and resource allocation is completed in real-time.[5] Job prioritization might complicate at the scheduling process as some jobs can wait, while others cannot. However, High-priority jobs pre-empt low-priority jobs, which might delay completion of the job.

The meta-heuristic optimization algorithms use two basic strategies while searching for the global optimum; exploration and exploitation. While the exploration process succeeds in enabling the algorithm to reach the best local solutions within the search space, the exploitation process expresses the ability to reach the global optimum solution which is likely to exist around the local solutions obtained. A meta-heuristic algorithm must be able to rapidly converge to the global optimum solution of the related objective function.[6] Furthermore, the run-time required by a meta-heuristic algorithm to reach to global minimizer and the total calculation amount should be at acceptable levels for practical applications. The algorithmic structure of a meta-heuristic algorithm is desired to be simple so that it allows an easy adaptation to different problems. Also, it is desired that the meta-heuristic algorithm has no algorithmic control parameters or very few algorithmic control parameters excluding the general ones (i.e., size of population, total number of iterations, problem dimension) of the population based optimization algorithms. If a meta-heuristic algorithm has algorithmic control parameters, the related algorithm must not highly dependent on the initial values of the mentioned algorithmic control parameters.

In this paper we are proving literature review regarding the meta-heuristics approaches used to solve container scheduling problems in cloud environment. The remainder of this paper is structured as follows: Section 2 is related work regarding literature review and summary, Section 3 is conclusion.

2. Related work

Docker container scheduling becomes gradually researcher’s hot topic. In which meta-heuristics is one of the approach used in case of container scheduling. In this paper we have discuss scheduling of containers with the help of meta-heuristics approach. This is first kind of review work provided in case of meta-heuristics approach used for scheduling container in cloud computing.

Literature Review

Reference	Objective of paper	Methodology/Technology	Limitation
[7]	The first ant colony algorithm of container scheduling for docker with the goal to enhance resource utilization and provide load balancing	Ant colony algorithm, Docker	Use fixed set of meta-heuristic parameter
[8]	The 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.	Ant colony algorithm with Best fit function	Requires lots of computation and takes long time to run
[9]	A new container scheduling algorithm based on multi objective optimization (multiopt) with a goal to overcome the default scheduling of docker swarm.	Multiopt, Docker	Fault tolerance of containers not considered.
[10]	A multi objective container placement ant colony optimization algorithm that modify existing ant colony algorithm and used additional objective network usage and cost	MOCP-ACO	MOCP-ACO performs better for all tested cases, except run time.
[11]	A particle swarm optimization based container scheduling for docker platform which have solved the problem of insufficient resource utilization and load balancing problem.	PSO, Docker	Used Fixed Parameters
[12]	The placement of containers under a nonlinear energy consumption model.	IGA	Only considers a single optimization objective
[13]	A Two-stage Multi-type Particle Swarm Optimization approach, named TMPSO, to energy-aware container consolidation in Cloud data centers.	TMPSO	Could work on multi objective optimization problem.
[14]	A genetic algorithm approach, using the Non-dominated Sorting Genetic Algorithm-II (NSGA-II), to optimize container allocation and elasticity management.	NSGA-ll	Should perform in real container cluster.
[15]	A new energy-efficient container-based scheduling (EECS) strategy for processing various types of IoT and non-IoT based tasks with quick succession.	APSO	Parameter tuning should be performed.
[16]	The task scheduling method in the cloud computing environment, and combines the improved ant colony algorithm and the adaptive particle swarm algorithm.	ACO , APSO	Use few optimization criteria

3. Conclusion

Unlike traditional virtual machine, the usage of containers has been increasing in recent years. Hence container scheduling becomes important problem in the field of cloud computing. There are different approaches to perform scheduling in containers. In this paper we have discuss meta-heuristics techniques used for scheduling containers and summarized to understand objectives, methodology and limitation of existing work.

References

[1] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, "Cloud container technologies: A state-of-the-art review," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 677–692, 2019, doi: 10.1109/TCC.2017.2702586.

[2] 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.

[3] J. Bhimani *et al.*, "Docker container scheduler for I/O intensive applications running on NVMe SSDs," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 4, no. 3, pp. 313–326, 2018, doi: 10.1109/TMSCS.2018.2801281.

[4] S. H. H. Madni, M. S. A. Latiff, J. Ali, and S. M. Abdulhamid, "Multi-objective-Oriented Cuckoo Search Optimization-Based Resource Scheduling Algorithm for Clouds," *Arab. J. Sci. Eng.*, vol. 44, no. 4, pp. 3585–3602, 2019, doi: 10.1007/s13369-018-3602-7.

[5] 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.

[6] P. Civicioglu and E. Besdok, *A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms*, vol. 39, no. 4. 2013.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.