Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN : 2348-4470
p-ISSN : 2348-6406

International Journal of Advance Engineering and Research Development

Special Issue on Recent Trends in Data Engineering

Volume 4, Special Issue 5, Dec.-2017

Docker Containers as Dynamic resource allocator

Prof. P.S Gaikwad[1]

Margi Adesara[2],Mayuri Jadhav[2], KruthiKanukuntla[2] ,Chaitrali Mhaske[2]

[1]*Assistant Professor Computer Engineering, AISSMS IOIT*
[2] *Student Computer Engineering, AISSMS IOIT*
SavitribaiPhule Pune University, Pune, India

Abstract —*Containers are made up of container images , which are light-weight, executable packages of software. In recent times containers are in boom in the IT world. Although containers are still in a primitive stage of development, they are having increasing use in production environments this is because they promise a streamline, easy to deploy and secure method of implementing specific infrastructure requirements and they also per an alternative to virtual machine. Containerization gained prominence with the open source Docker tool, which developed a method to give containers better portability allowing them to be moved among any system that shares the host OS type without requiring code changes. The container technology has evolved over period of time to overcome the drawback of the earlier systems. The novel approach aims at scheduling and load balancing of the container. Our results show significant improvement over standard scheduling with Docker. Docker made it easy to build and run containers from the command line.*

Keywords- *Container, Docker, Cloud ,Container-as-a-service, Virtualization.*

I. INTRODUCTION

Project Idea

The designed system in the DevOps category which provides a platform for the developers and testers where they can get the necessary software from the cloud as per the requirement. Overcoming the drawback of tediousdownloads and excess of load on the system leading to loss of data and crash of the system, we are designing containers which will be time and cost efficient(time and cost) the system. Along with this we are implementingscheduling and load balancing for the containers using a novel approach.

Motivation of the Project

Docker is a mature and adopted development tool. It also runs great on windows, macos and linux. While k8s can run containers via the rkt runtime, its still pretty new and will probably introduce unnecessary headache and edge cases. Docker also has a head start ona vibrant ecosystem for base images. Docker made it easy to build andrun containers from the command line. This technology was released atthe right time and in the right place. And it got huge traction. Dockerused this traction to get a very large amount of VC capital. Thereforethey now need to show a very large return.

II. RELATED WORK

In 2010 A Cloud Sim Based Visual Modeller for Analysing Cloud computing Environments and Application paper was published which emphasized on distributing applications among cloud infrastructures. But this paper had

various drawbacks which could not handle large data at the same time it would take more turn aroundtime . So to overcome this later in 2012 Fuzzy Network algorithm was introduced which increasedthe performance by using the Cloud Sim tool and it reduced the turnaround time.Then over a span Fuzzy logicbased dynamic loading of resources was done. Later in 2016role of Docker in shaping cloud technology and containersin virtualization was given emphasis. But the system didn't have any resource allocation technique. The in 2017 DynamicResource Allocation Algorithm for Container based ServiceComputing was done.This paper was successful in overcomingall the drawbacks of previous papers hence using containersmore efficiently.In earlier times Cloud analyst helped in distributing applicationsamong cloud infrastructures this was proposed by thepaper "Cloud Analyst :A Cloud Sim Based Visual Modeller forAnalysing Cloud computing Environments and Application".But the drawback of the system was it could not handle largedata at the same time it would take more turn around time.

This paper emphasizes on problem of effectively managing CPU utilization when many containers share a single set of resources. The problem is exacerbated in environments where an application is designed to maximize performance by utilizing resources as efficiently as possible. For example, in high performance computing, applications are routinely optimized for cache access and locality. The performance of such applications can be destroyed in virtualization environments by removing the benefits of cache locality; moreover, stalls can result for synchronized processes or threads.

Later in 2016, Cloud computing and Containers use increased tremendously and was in a boom in IT sector improving many drawbacks of previous papers. This had a very efficient applications in IT companies. This emphasized upon Role of Docker in shaping cloud technology and containers in virtualization. But could not allocate resources dynamically over the systems

This given paper emphasizes on the problem of effectively managing CPU utilization when many containers share a single set of resources. The problem is exacerbated in environments where an application is designed to maximize performance by utilizing resources as efficiently as possible. For example, in high- performance computing, applications are routinely optimized for cache access and locality. The performance of such applications can be destroyed in virtualization environments by removing the bene_ts of cache locality; moreover, stalls can result for synchronized processes or threads.

III. PROPOSED ARCHITECTURE

In the respectiveCaaS (Container as a Service) framework, the container is the basic component that constitutes the business workflow. The physical node (mostly a server) is the main carrier to deploy and execute containers. A cluster basically consists of a large number of parallel workflows that execute independently. The workflow changes on calculation time, data volume, network latency, submitted or completed time and other aspects, but they all share the resources in the same cluster. Thus, it is difficult to accurately estimate the resource consumption for the entire workflow. The performance of the nodes is determined at the beginning of workflow execution.In this way, the workload is dynamically distributed among the nodes, in a more balanced manner, when it receives a request for a container deployment and execution As shown in the above figure two modules, load balancer and the container manager, coordinate to complete the resource allocation. Load Balancer identifies the least loaded physical node based on the current load status of each node in the cluster, and Container Manager is responsible for deploying a container

according to the corresponding configuration file generatedin the previous stage.Every node in a cluster is a complex combination of multiple types of resources, and the physical configurations of resources for each node may be heterogeneous as well. Similarly, to handle a complex system where a lot of uncertain parameters exist, we propose to apply fuzzy logic control again, instead of conventional modelling algorithms .

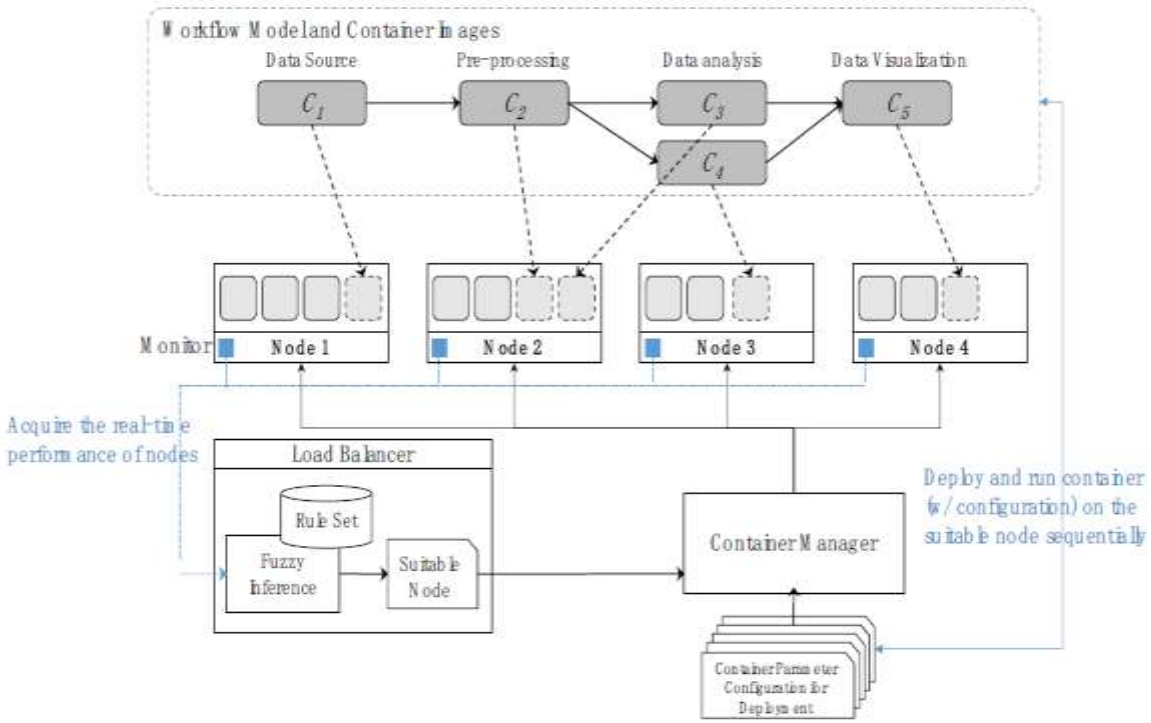

Fig. 1 Proposed Architecture of Scheduling of Containers.

III.ALGORITHM

Name of Algorithm- Container Scheduling Algorithm.

Description-The Container Selection algorithm selects the least busy node satisfying the minimum requirements of the current container and sends it to the Container Manager. As soon as the container has been deployed, Load Balancer prepares itself to schedule another container in the workflowbuild a queue for the current staging containers. It might be possible that returns null for a certain container, which means that there is no available node for deploying it. In this case, the corresponding workflow is suspended temporally, and the next sequential container(s) will be chosen to deploy, after being evaluated by the Container Selection algorithm.

Input constraints- Cl_table,Cids, currCount, minCount.

Output constraints- Scheduled containers with respective applications allocated with particular containers.

1. Initialize, *Cl_Table(*0..n-1*)* ← **0** At start all Coatiners's have zero allocation**.**, *K*← *m, C_id*←*-1* , *Cids()=-1,i*← *0*, currCount← 0, minCount← *Max_Value, TempCid*← **-1**;
2. Parses C_List() to LoadBalancer:
3. ***For i*← *0 to k*** *//SelectContainer randomly*
{
4. *TempCid*← *random* ***(****C_List()****)****.*
5. *C_id*← *TempCid*
6. ***If C_id Exist in Cl_Table(****C_id****) then***
7. currCount← *Cl_Table(C_id)*
8. ***Else***
9. currCount← ***0***
10. **C***ids()* ← *(C_id,* currCount*).*
11. ***End for***


```
}
12.  TempCid← -1
13.  currCount← 0
14. For i ← 1 to k
  {
15.   TempCid ← i
16.   currCount← Cids(TempCid)
17.  If currCount<minCount then
18.     minCount= currCount
19.   C_id← TempCid
20. End if
21. End for
  }
```

V. APPLICATIONS

[1] There is an advantage to running production code in containers.An online site for discounted luxury goods,Glit,broke seven large applications into 300 microservices,with a small team maintaining each service.These services can be handled more easily and safely than the large applications on cloud.

[2] Red Hat ,IBM, Microsoft,and a couple of startups specializing in container operations,are interested in where it leads.

[3] Google search operations use containers by themselves , launching 7000 containers per second,which amounts to approximately 2 billion every week.Containers are one of the secrets to the speed and smooth operation of Google search Engine.

VI. FUTURE SCOPE

The main aim of the system will be to minimize the utilization timeand to reduce the load on system. To do the same our algorithm will analyze and replicate the containers at the back-end in case there is load on the system without disturbing the client on the front-end. Because of using cloud computing the system becomes more efficient.

VII. CONCLUSION

In this paper,we have designed a system in the DevOpscategory which provides a platform for the developersand testers where they can get the necessary softwaresfrom the cloud as per the requirement.Overcoming thedrawbacks of tedious downloads and excess of load on the system leading to loss of data system crash,we aredesigning containers which will be time and cost efficient the system. Along with this we are implementing scheduling and load balancing for the containers using a novel approach. Future studies are planned to address some of the limitations of this work, as well as to extend our proposed framework in more directions. Thecommunication mechanisms of workflows, tasks and containers will be further investigated, to provide across-layer inter-operation architecture for complex workload monitoring and prediction.

VIII. REFERENCES

[1] Bhathiya Wickremasinghe1, Rodrigo N. Calheiros2, and Rajkumar Buyya1,"Cloud Analyst :A Cloud Sim Based Visual Modeller for Analysing Cloud computing Environments and Application," 2010 24th IEEE International Conference on Advanced Information Networking and Applications.

[2]V. Venkatesa Kumar and K. Dinesh "Job scheduling using Fuzzy Network Algorithm in Cloud Environment," Bonfring International Journal of Man Machine Interface, Vol. 2, No. 1, March 2012.

[3] Md. S. Q. Zulkar Nine, Md. AbulKalam Azad, Saad Abdullah, Rashedur M Rahman, "Fuzzy Logic Based Dynamic Load Balancing in Virtualized Data Centers",10.1109@FUZZ-IEEE.2013.6622384.

[4] D. Bernstein, "Containers and cloud: From lxc to docker to kubernetes," *IEEE Cloud Computing*, pp. 81-84, 2016.

[5]X. W. Ye Tao, XiaoweiXu and Guozhu Liu, "Dynamic Resource Allocation Algorithm for Container-based Service Computing, "2017 IEEE 13th International Symposium on Autonomous Decentralized Systems.

[6]G. Sakellari and G. Loukas, "A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing," Simulation Modelling Practice & Theory, vol. 39, pp. 92-103, 2013.

[7] X. Luo, Y. Lv, R. Li, and Y. Chen, "Web Service QoS Prediction Based on Adaptive Dynamic Programming Using Fuzzy Neural Networks for Cloud Services," Access, IEEE, vol. 3, pp. 2260-2269,2015.

[8] C. Kleineweber, A. Reinefeld, and T. Schütt, "QoS-aware storage virtualization for cloud file systems," in Proceedings of the 1st ACM International Workshop on Programmable file systems, 2014,pp.19-26.

[9] X. W. Ye Tao, XiaoweiXu and Guozhu Liu, "(in press) Containeras-a-Service Architecture for Business Workflow," Int. J. Simulation and Process Modelling, 2016.

[10] A. T. Saraswathi, Y. R. A. Kalaashri, and S. Padmavathi, "Dynamic Resource Allocation Scheme in Cloud Computing," Procedia Computer Science, vol. 47, pp. 30-36, // 2015.

[11] D. Bernstein, "Containers and cloud: From lxc to docker to kubernetes," IEEE Cloud Computing, pp. 81-84,2014.

[12] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," Linux Journal, vol. 2014, p. 2, 2014.

[13]www.dockerhub.com

[14]www.edureka.com

