Dynamic Resource Allocation Algorithm Using Containers

Akshay Patil[1], Hrishikesh Jadhav[2], Swaraj Mankar[3], Arti Jaware[4], Prof. Bhushan Pawar[5]

[1,2,3,4]Students, Assistant Professor[5], Department of Computer Engineering
Savitribai Phule Pune University, India

Abstract - *High processing data and dynamic work load on servers results in resource depletion or resource exhausting. To overcome such situations, basic response is scaling the available resources, but rather than scaling the resources, we should use available resources optimally*

Due to high processing cost of data most of data processing tasks becomes hard to maintain. But rather than scaling the available resources we can focus on better utilization of resources available to us and reducing the wastage of resources.In server farms, most of the maintaining cost is due to cooling system and power consumption of servers in ideal state of servers. In past few years, data scientist and researcher developed many algorithms and models to allocate resource dynamically as an alternative for virtualization which is hypervisor based and requires static allocations of resources. Virtualized system holds resources even when it is ideal, this results in unfair resource allocation.

This paper uses Dynamic Node Selection Algorithm for Container Deployment for building a portal which will use Docker and Hadoop in combine for better resource utilization and increase processing speed.

Key Words: *Big data, Map-reduce, Containerization, Docker, Fuzzy interface system.*

1.INTRODUCTION

Energy consumption rate of the server farms is way more than the actual energy required for the processing. Huge part of the energy is wasted on running the servers ideal and on cooling system of the server farm. This can be avoided by better utilization of servers for every situation in dynamic load distribution. Currently there are some framework which are used to share resources in cluster for workload management. For cloud computing environment various runtime parameter set statistically which lead to unbalanced resource allocation in cluster. By adding and removing resources at run time we can make cloud platform flexible and cost efficient. For this purpose, we use containerization. Nowadays, containerization gained popularity by various major vendors like Microsoft, google, amazon as an alternative to virtual machine.

To assign the load on proper nodes firstly, determine the status of each active node in the network, then pass it to the fuzzy interface system(FIS) which dynamically calculates workload on each node and then node selection algorithm for container deployment(NSCD) is used to deploy containers on the best node in the cluster.

2. Theoretical background

[1] According to the proposal, for unreliable nature of distributed systems which is used by cloud providers should have another alternative. So author proposed XtreemFS, a file system for cheap scale-out solution for storage.

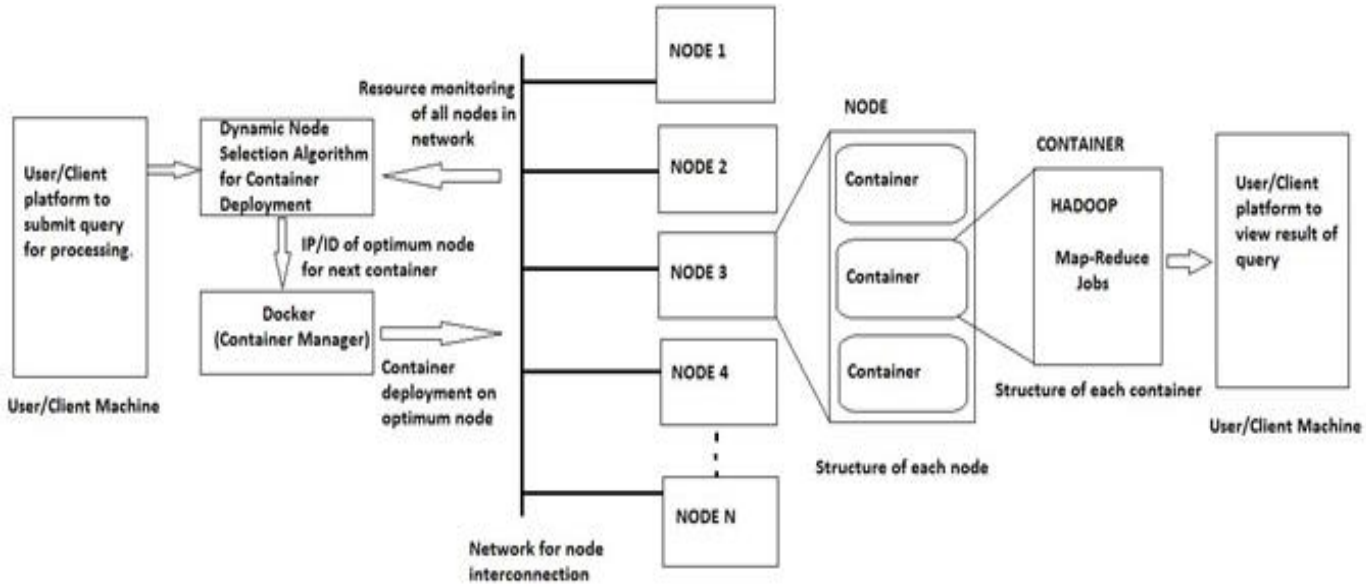
[2] Proposed that, rather than providing statically configured computing resources, system should provide computing resources on demand. This encourages cost efficient usage of computing resources.

[3] Authors presented a Container-as-a-Service (CaaS) framework to deploy workflow or applications automatically across multiple vendor systems. Containers allow services to run in isolated environments without the extra overhead of running entirely separate operating systems. But, the problems of how to effectively manage computing resources for containers remain open, because multiple applications sharing the same resources can result in substantial resource contention among the applications in the containers.

[4] Proposed an adaptive control scheme by serializing applications for the cases of overutilization of CPU resources, where the quality of services in the cluster was considered as a multidimensional objective function. authors proposed a solution by extending the concept of time slicing to the level of virtualization container.

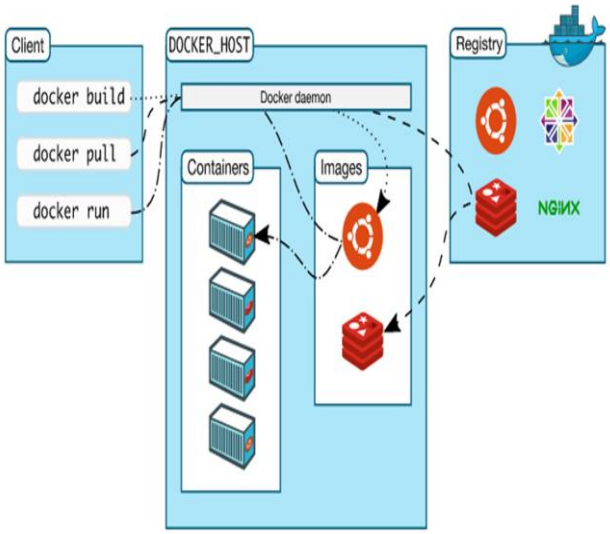
[5] Proposed "Fuzzy logic based dynamic load balancing in virtualized data centres". Authors developed a dynamic fuzzy load balancing algorithm which allows system to select next virtual machine based on resource utilization to schedule upcoming job.

[6] Described node selection algorithm for container deployment(NSCD), where a Fuzzy Inference System (FIS) is applied to dynamically predict the most proper node (server) where the selected containers will be deployed. This algorithm is used for better utilization of resources and reduction in the data processing cost. This is the basic algorithm used in our architecture. We used Hadoop for the better handling of big data and some alteration in the algorithm which reducescomplexity at some level. We are using Best Fit approach for allocation of nodes.



I. Docker

Docker is a tool for management of container. Containers provides kernel level virtualization which is light weight than virtual machine. Management of containers such as creating, deploying them becomes simplified by Docker. Application's environment and related dependencies are bundled in a single package, ready to deploy.



II. Methodology

Rate of growing data with time in this modern age is enormous. Data can be unstructured or semi-structured generated from various heterogeneous systems. To create environment that provides computation and distributed storage across cluster of resources we use Hadoop. Hadoop implicitly distributes load on clusters. Hadoop can handle data processing in the provided cluster.

For better resource utilization and eliminating of overheads of virtual machine we are using Docker. It is easy to deploy container using Docker. Hadoop is structured in container, so we can easily move processing environment for Hadoop. This helps in dynamic adaption of available resource.

Dynamic Node Selection Algorithm for Container Deployment, monitors resource availability and usage of nodes, which is useful for node selection based on container's requirements and resource availability.

The architecture is supported with the user platform which is portal for query submission and viewing result of query after processing. User is also provided with the graphical presentation of resource utilization of all the nodes available in the system.

Algorithm monitors and collects information about active nodes in network and resource utilization of active nodes. Algorithm provides IP of optimum node for processing to Docker which will deploy container on that particular node.

Each node is capable of running multiple containers according to the resources available for that node. Each container runs Hadoop image which will process the query and provides output to the user.

5. Data acquisition and algorithm

5.1 Data acquisition

Required data and metadata about the system and user query is collected before start of the algorithm. This data can be collected as follows

Consider a "Info" class with attributes as CPU utilization, Memory utilization, I/O utilization and Network utilization.

5.2 CPU utilization

It is overall CPU utilization of the system's processing power. This can be obtained by *"iostat"* which is a system monitoring utility of Linux.

5.3 Memory utilization

Memory utilization if required for the amount of memory available or used as per the requirements to find total processing load on the system. This can be obtained by the *"/proc/meminfo"* file in Linux file system.

5.4 I/O utilization

It is required for finding pending work. Percentage shows how resources are busy for pending task. It can be obtained by "iostat" utility.

5.5 Network utilization

Network utilization provides traffic on network. It can be obtained from "/proc/net/dev" file.

5.6 Optimal node selection algorithm

Input:
Available node in cluster: N = (N1, N2, N3……Nn); Container to deploy "Container"
output:
optimal node for container. It will provide IP of optimum node to Docker.

Steps:

1. Find active devices connected in the network. N = (N1, N2, N3……Nn)

Active devices can be identified by network monitoring tools or classes/API provided by language (E.g. InetAddress class in Java)

2. Find all attributes of "Info" class for each node. Info(N) = (CPU utilization, Memory utilization, I/O utilization, Network utilization).

3. Rank all the nodes according to according the load on each node in descending order.

Load = Desc(N1(load), N2(load), N3(load) …. Nn(load))

4. Find minimum requirements of the container to deploy. This can be obtained from the available Docker utilities. Container(min) = (CPU utilization, Memory utilization, I/O utilization, Network utilization).

5. For (Container(min) >= N(info))
for (Node with minimum rank)
Select optimum node for container deployment.

6. Check if next job is available.
If (Next job available)
Repeat from step1
If (Next job not available)
Stop

7. Stop

6. Experimental setup

For validating the propose framework we create small cluster of computers that used four node in which one is act as server and other three act as portal(client).this four machine are connected via Giga-Ethernet and each node equipped with 1xCPU@2GHZ and 2GB RAM. OS running on primary hardware is Ubuntu 14.04(LTS) with 3.13 kernel version (Docker support kernel version of 3.13 or more) on which Docker and all its dependencies are installed.

7. Proposed Outcomes

Outcome of the project will be the result of the user query submitted by the user for processing. The query result will be accompanied by the graphical representation of the resource utilization of active nodes in the network.

Output of the project represents the better utilization of resources compared to the virtualization and reduction in the data processing cost due to use of Docker which is light weight virtualization tool, the dynamic resource allocation approach of the algorithm which is capable of better load sharing among the available nodes and huge data handling capacity of Hadoop.

8. Conclusion

By using Docker which is container manager and the Node Selection algorithm for container deployment(NSCD) using dynamic resource allocation approach the processing speed has significant increase. Container which is light weight virtualization technique helps to achieve maximum resource utilization and minimum wastage of resources. The algorithm definitely decreases processing cost of data. This approach can help to build maximum numbers of server farms possible with minimum cost required.

References

[1] 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.

[2] 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.

[3] X. W. Ye Tao, Xiaowei Xu and Guozhu Liu, "(in press) Container-as-a-Service Architecture for Business Workflow," Int. J. Simulation and Process Modelling, 2016.

[4] J. Monsalve, A. Landwehr, and M. Taufer, "Dynamic CPU Resource Allocation in Containerized Cloud Environments," in Cluster Computing (CLUSTER), 2015 IEEE International Conference on, 2015, pp. 535-536.

[5] Z. Nine, S. Md, M. A. K. Azad, S. Abdullah, and R. M. Rahman, "Fuzzy logic based dynamic load balancing in virtualized data centres," in Fuzzy Systems (FUZZ), 2013 IEEE International Conference on, 2013, pp. 1-7.

[6] Ye Tao Xiaodong Wang, Xiaowei Xuin and Yinong Chen their research paper "Dynamic Resource Allocation Algorithm for Container-based Service Computing" published in the 2017 IEEE 13th International Symposium on Autonomous Decentralized Systems.