**TC Control of Docker Container**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aldawsari Mohammed Florida International University malda021@fiu.edu | Nishant Maurya Florida International University nmaur009@fiu.edu | | Shuai Xue Florida International University sxue004@fiu.edu | |
|  | Yezehao Huai Florida International University yhuai001@fiu.edu | |  | |
|  |  |  | |  |

**ABSTRACT**

Docker technology makes it easier to contain an application and deploy & run container in data center or in a cloud. Native docker run command has various commands to manage container resources such as disk, CPU, memory but not network traffic control. Traffic control is necessary as it helps the user to test the application that are running on the containers. It is very important to test applications for network bandwidth control as there might come.Installing and Configuring Docker containers in the host system. Collecting real world twitter data using storm, preprocesseing and feeding it to the data analysis system in each container to run word count on it. Control the traffic by changing the bandwidth between the containers and analyze the performance changes by measuring the performance metrics such as, CPU - Memory Usage - Disk Usage and Network IO metrics.

**Abbreviations**

VM: Virtual Machines, CPU: Central Processing Unit, I/O: Input and Output

**Keywords**

Big Data, Containers, Resource Management, Storm, Container, Traffic Control.

**1 INTRODUCTION**

Big data is a term that is used to represent datasets that are so complex and large that it is hard for traditional data processing systems to deal with them. Big data are underutilized because big data technologies present many challenges, softwares/tools like Spark Streaming, MapReduce, Storm, Dryad, Graph-related systems,etc are lauded for its ability to handle huge amount of unstructured and structured data, but these softwares/tools are not very easy to manage or use. They require extensive internal hardware and software resources to maintain and understand the challenge of the big data project.

 It is crucial to accommodate for scale up and scale down on demand, with big data. If the data keeps on growing more resources needs to be allocated and most importantly the impact of network bandwidth should carefully be considered. That means for typical cluster architecture, all the infrastructure of all nodes in the cluster need to be the same specification such as memory, CPUs, operating systems and I/O bandwidth.

To overcome these problems we can make use of containers. Docker Containers could be regard as lightweight versions of virtual machines. In fact, each running container is only a process running on the host machine.They occupy small amount of space on the server, easy to create and destroy, and is really fast. You can equip the containers with the libraries and tools, code, framework needed for data analysis, eliminating the need to spend hours in installing the packages across different environments. Docker Containers provide standardization and efficiency, as well as we can create images of that container and every user can boot up an isolated standardized environment from that image. Some other advantages of using Docker Containers are: Cost Saving, Faster Configurations, Multi-Cloud Platforms, Isolation and Security.

**2 Motivation and Use Case Scenario**

**2.1 Motivation**

Nowadays, there are many cloud computing services based on Docker Containers. The Docker networking and the traffic control technology is essential. Therefore, in this project, we will focus on comparing the performance losses of four data analysis systems, namely Hadoop MapReduce, Spark and Storm, due to the bandwidth size change in a cluster of containers.

Docker technology makes it easier to contain an application, deploy and run containers. Native Docker has various options to manage container resources like CPU, Memory, Disk but there is no option to manage network resource utilization.

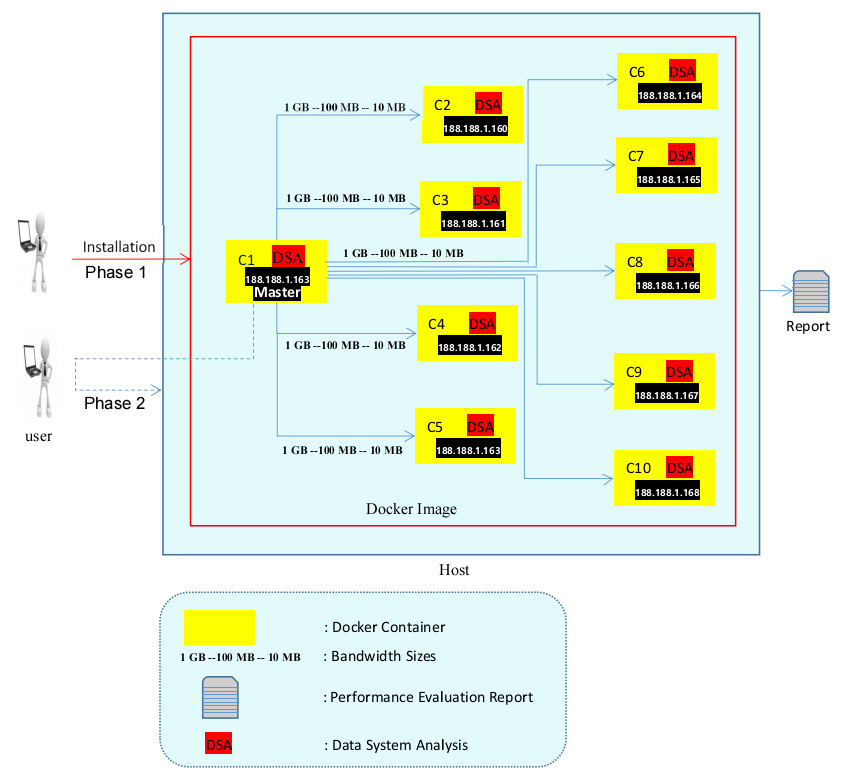
Effective network resource management will lead to better utilization of latent hardware and better performance isolation. However, to the best of our knowledge, docker does not allow per-container bandwidth control support.So, to overcome this we will implement a method to limit the container network bandwidth based on container Id dynamically.

Docker Container traffic control project will provide the means to control network traffic based on container id. This project will also enable users to limit the usage of network bandwidth and monitor containers computation processes and identifying containers that might cause the performance lose in terms of the bandwidth size.

**2.2 Related Work**

Wes Felter [1] use a suite of workloads that stress the CPU, memory, storage and networking resources. They show that containers result in equal or better performance than VM in almost all cases. Ayush Dusia[2] present an extension to the presently limited Docker’s networks to guarantee quality of service on the network. Their work ensures that time-sensitive and critical applications hosted in high-priority containers, can get a better share of network bandwidth, without starving other Containers. Janki Bhimani [3] show that Apache Spark using Docker can obtain speed up of over 10 times when compared to using VM. To the best of our knowledge, there is no work have been done to address the impact of bandwidth change on the performance of different data analysis systems in a cluster of docker containers.

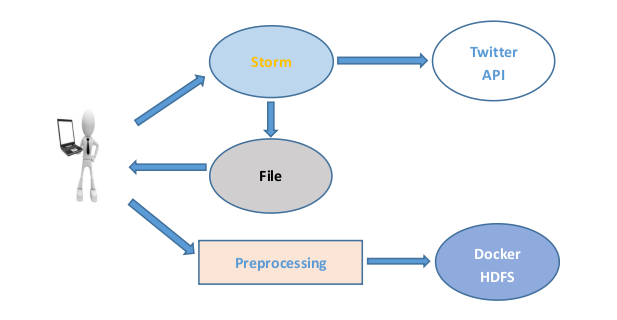
**2.3 Use Case Scenario**



**Figure 1:** *Shows the use case scenario of the following phases of our project where in the first phase the docker platform initially installed on the host machine. Then a docker image will be created that contains all needed libraries as well as the preferred operating system. After that as shown in the diagram 2 in blue, a cluster of containers (specifically 10 containers) will be created containing all the required data analysis systems for the project. Then A network  will be established between the aforementioned containers.*

*In the second phase, a word count application will be run on each data analysis systems using the aforementioned cluster with different bandwidth size.*

*In the final phase, a performance report will be created for each iteration in the second phase.*



**Figure 2:** *Shows the process of collecting the real world data from Twitter API using an open source distributed system called Storm and storing it in Docker HDFS.*

**3 Project Goals and Deliverables**

**3.1 Goals**

Make use of Docker Containers by distributing the computations required for Word Count application on a self collected real world Twitter data using some data analysis systems such as Hadoop MapReduce, Spark and Storm. Compare the performance losses or gain by controlling (increase or reduce) the traffic of the bandwidth between containers and most importantly identifying the causes.

Monitor containers computation processes and identifying containers that might cause the performance lose in terms of the bandwidth size.

**3.2 Deliverables**

* Project Proposal
* Final Report
* Code
  + Technical installation instruction
* Powerpoint Presentation

**4 Scope**

Docker Container traffic control project will provide the means to control network traffic based on container id. This project will also enable users to limit the usage of network bandwidth and monitor containers computation processes and identifying containers that might cause the performance lose in terms of the bandwidth size.

**45 Experimental Evaluation**

Number of Machines that will be used: 1 machine with 10 docker containers

Input: large text file( a self collected Twitter data)

Output:

* Word Count
* Performance measure with different bandwidth size for each data analysis system.

Metrics to be Used: CPU - Memory Usage - Disk Usage and Network IO metrics.

**REFERENCES**

|  |  |
| --- | --- |
| [1] | Wes Felter, Alexandre Ferreira, Ram Rajamony An updated performance comparison of virtual machines and Linux containers , March 2015. |
| [2] | Ayush Dusia, Yang Yang, Michela Taufer Network Quality of Service in Docker Containers, Sept. 2015. |
| [3] | Janki Bhimani, Zhengyu Yang, Miriam Leeser, and Ningfang Mi, Accelerating Big Data Applications Using Lightweight Virtualization Framework on Enterprise Cloud, 2015. |
| [4] | [Docker-Network-Bandwidth. https://github.com/shivacherukuri/Docker-Network-Bandwidth](https://github.com/shivacherukuri/Docker-Network-Bandwidth) |
| [5] | Gubbiotti, G. 2000. *Publication quality tables*. <http://www.ctan.org/pkg/> booktabs. |
| [6] | Ben Youssef, J. and Layadi, A. 2010. *J. Appl. Phys.* *108*, 053–913. |
| [7] | Helmer, A., and Chappert, C. 2010. *Phys. Rev. B* *81*, 094416. |
| [8] | American Mathematical Society. 2015. *Using the amsthm Package*. April. http://www.ctan.org/pkg/amsthm. |