International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 4373-4378
© Research India Publications. http://www.ripublication.com

Comparitive Study of Hadoop over Containers and Hadoop Over Virtual Machine

Avanish Singh, P. Gouthaman, Shivankit Bagla and Abhishek Dey

*Department of Information Technology, SRM Institute of Science and Technology,
Kattankulathur 603203, Tamil Nadu; India.*

Abstract

Purpose - The purpose of this paper is to elaborate the concept of Containerization which enables usage of Hadoop Infrastructure on a container which can allow clusters to run rapidly fast and accessible manner and it can be used for installing, processing and monitoring/analysis of Big Data under various configurations.

Design/methodology/approach – This paper is a research-based comparative study of the VM services which deploy HADOOP framework upon them using a simple virtual machine and using container services from Docker. The analysis is done using a tool known as terasort, which has a program called Teragen which generates the random data on the HDFS cluster. Then it sorts the data using terasort services. The result is analysed upon the time taken by the Virtual Machine and Docker Container.

Findings – The result of the research proves that the time taken during benchmark testing of the cluster on the container was much lesser than that of Virtual machine, and result also proves the better efficiency of the cluster on the container. The results shown in graphical analysis clearly indicates the difference in time taken by both the services. Docker Container implementation clearly takes lesser time when processing any task into the cluster.

Originality/value – The originality in this research is the valuation and comparision of Docker Container with Virtual Machine to deploy HADOOP framework for the processing of data. The evaluation and time-based analysis is done solely by the authors using the tool known as Terasort. The value obtained during research can help in optimising the Data Centres which runs Virtual Machine to deploy their services.

Keywords - Container; Docker; Big Data; Process; Virtualization; Optimization

INTRODUCTION

HADOOP

One of the open source software, Hadoop is used for storage and processing of the big data on the large clusters of the vendible hardware. Hadoop is one of the top-level projects from Apache which is being deployed and it is used by the top global community of service providers and users. It provides scalable storage for all varities of data, enhances the processing power and provides greater agility to handle concurrent tasks virtually. Further, it can also be stipulated that the containers take less CPU and memory requirement than virtual machines. Virtual Machines can run different types of guest operating systems than containers. Considering these assumptions, it can be concluded that one has to consider few things to deploy big data application to be deployed on virtual machines.

Earlier when data was not in huge demand, these large files were considered a waste as storage was one of the costliest peripheral in the world of Computers. It was the company Yahoo that brought an entirely new concept of building a cluster of computers from the waste peripherals and brought up an entirely new innovation of Hadoop to this world.

Hadoop framework mainly focuses on the following modules - Hadoop Common: Containing libraries and utilities which are required by other Hadoop modules. HDFS (Hadoop Distributed File System): It is a type of file-system which distributes and stores the data on distributed hardware by dividing data in several blocks by default block size is kept at 128 MegaByte but it can be changed as per user requirements.

Hadoop YARN: Yarn is a software model which is installed over Hadoop map-reduce to perform the operation such as computation, managing and processing of big data.

Hadoop MapReduce: MapReduce is a programming model which is made for big data processing and computation. where the user needs to provide map-reduce programs. Hadoop modules were created in such a way that whenever any hardware failure occurs it can handle its configuration on its own without any external help. Apache Hadoop's MapReduce and Hadoop Distributed File System components are derived from Google's MapReduce project and Google File System (GFS) papers respectively.

Other Hadoop-related projects at Apache are :

Ambari: Ambari is a tool which is useful for monitoring and analyzing data for many Hadoop framework tools such as the spark, scoop, pig and many more.

It also provides a dashboard which is used to monitor the performance of cluster based on its performance and utilization of resource.

Avro: it is a data serialization system it is utilized in cloud based application.

Cassandra: Cassandra is a multi-node database cluster which has high scalability. Since it run on multi node cluster which is why is has no single point of failure.

HBase: HBase runs above hadoop cluster which is mainly used for storing data in large tables. Since it can store data in rows and columns it can be used as nor databases.

Mahout: it is a tool which can be used to run algorithms of machine learning and other data mining algorithm.

Pig: it uses high-level language i.e. pig latin for large-scale data processing.

Spark : spart is advance form of hadoop although if can also run above hadoop but it runs very faster than hadoop since it does on memory execution that is execution on ram while hadoop copy data from hard drive first then perform its operation which reduces it speed of operation. Spark can also run on multi node cluster which provides it high scalability and high availability. Spark can also run as independent cluster without hadoop as its backend. It has its own component such as Mlib which is a machine learning library used to run machine learning algorithm over spark cluster. It has also its own sql interface to run sql queries.

ZooKeeper : it is used for coordinating between different nodes of cluster, it has different znodes which is used to monitor different sectors of cluster. Zookeeper also provides dashboard which is used to monitor performance.

In this paper the aim is to compare and then hence evaluate the performace of hadoop in solving a particular problem and therefore calculating and comparing the final time taken by both, a virtual machine and a container service.

DOCKER CONTAINERS

DOCKER is the only company in the world which provide lightweight, high-performance computation, container service. It has been the first company which provides free and open source access to the containers. Because of a docker, the containers services come into huge demand. The docker company works under docker community which provides it more secure and user-friendly benefits. Since it provides regular updates to the container which is why the container will not compromise with the speed of its execution.

Docker is a company which provides containerization technology that is used for the creation of Linux® containers.

Working of Docker

The Docker uses a base machine or base operating system's kernel for its operating system instance which solves the problem of too many installations of OS and also increases the speed of container. Since docker provides image-based deployment model which makes it very easy to transfer from different places. One can commit running instance into an image file which can be saved in any storage medium and can be transferred from one place to other using any storage medium. Many tools can be built on docker container which gives it high user friendliness and also provides a lightweight structure which can be used easily and effectively. Docker provides fast access to these tools.

CONTAINERS

In the case one Containers, they are so well designed that they deploy each and every OS, in the form of a containers, just like a ship carries containers during transportation over the common deck of the ship, Docker company deploys containers to run the OS over the kernel of base OS which acts as the common kernel for the usage. This simply makes the containers much faster as compared to the general Virtual Machine, it can start a cluster faster, or perform any kind of tasks assigned on it’s application. Incase of a system failure, containers will reduce the downtime before the service resumes.

Containers can virtually run anywhere, greatly easing the development and deployment.

Containers help the developers in a manners that they are isolated from all the other applications. They also include softwares which are dependent solely on the applications.

Whenever one wants to run an application, they can use containers to satisfy their needs. All of this directly or indirectly helps the developing teams, engineers in diagnosing and debugging the problems in various environment more quickly. Working with containers will reduce the chances of having bugs while it will provide maximum uptime to their clients.

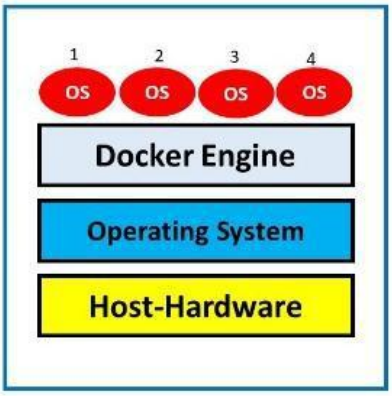


Figure 1: Docker Architecture

EXISTING SYSTEM

Workflow

A Multi-node Hadoop on large cluster of systems. All the systems are connected through network. There are two major parts of Hadoop i.e. Hadoop Distributed File System (HDFS) and Map-Reduce. On Hadoop cluster, one system is configured as client which works as intermediary between internal hadoop cluster and external environment. When some wants to process some data first client system copies the data on hdfs cluster which later can be processed using map-reduce. When a client first comes at namenode with data the namenode gives all information to the client such as block size, datanode location, number of replication to be made then according to the given block size it will be divided into blocks and copies to different datanode and then metadata information is copied on namenode.

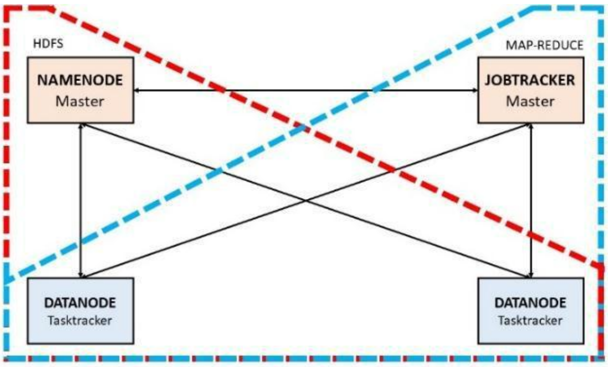


Figure 2 : Workflow of Hadoop Cluster

Virtualization

Virtualization has one of the major application in cloud computing through this technology one can install and use many Operating System on same hardware.

Hadoop is mainly installed on virtual machines. Since virtualization provides completely different environment to each virtual machines one can have completely different virtual hardware. Virtualization provides better resource utilization compared to the normal system.

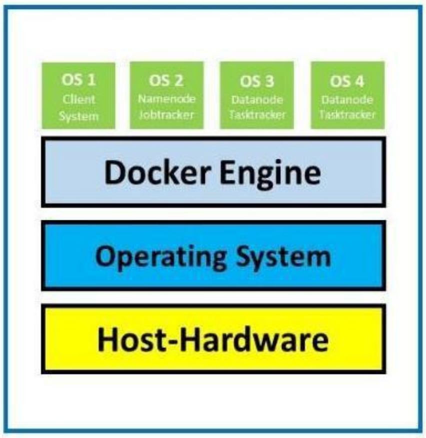


Figure 3: Hadoop Architecture with Docker Platform

Problem statement

Now days almost every Hadoop cluster runs on virtual machine. all big companies such as Yahoo, Facebook uses Virtual machines for Hadoop cluster formation. When one install os on virtual machine one need to install complete package of it which include's kernel, some basic application and so on. Sometimes these things could be very heavy to run multiple clusters. Let's assume a user wants to create 1000 node cluster than it could take lots of time, hardware and effort. All 1000 nodes will be running 1000 operating above it which will consume too much ram and will occupy lots of hard-drive. And when Hadoop cluster will run on a virtual machine the data in Hadoop distributed file system will be stored on hard-drive which is again slow and cause a problem when fast processing is needed.

Benchmark Testing

In this experiment, Terasort is used for benchmark testing which is one of the most widely used programs for benchmark testing. Hadoop already provides program written for this testing. Terasort benchmarking is done in three phases:-

- Terragen - this stage consist of generation of data where random raw data is generated.
- Terasort - in this stage actual processing of data takes place i.e. the sorting of data takes place at this stage only.
- Teravalidate - the sorted output is validated at this stage.

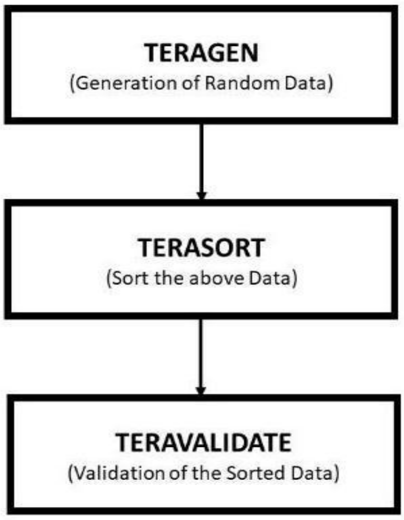


Figure 4: Terasort Benchmark Testing Flowchart

EXPERIMENT

In the experiment, the tool terasort is used. Which performs different types of benchmark testing. The Hadoop is run in 2 different environments, first in a virtual machine and then in containers. The performance and the time are taken by Hadoop while computing the problem statement in both the environments is compared.

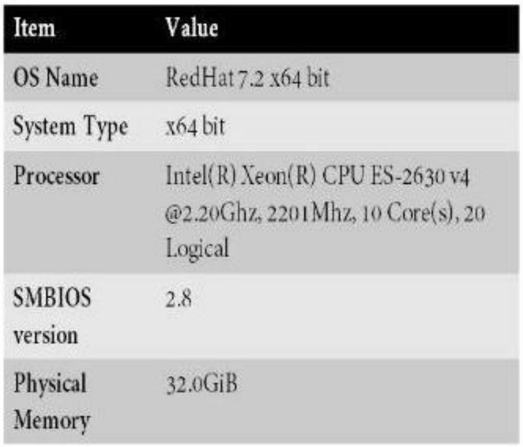
Experimental Setup

In this experiment benchmark testing has been done on two different setups:-

1 On Virtual machine:- total of five virtual machines have been used in this case. In which there are two datanodes and tasktracker(the same system was used for both datanode's and tasktrackers) , one namenode, one resource manager and one client. From client system, all the operation was performed. First data was generated on client system using teragen program which comes along with Hadoop.

2 All the five virtual machines were installed on the same system whose configuration is listed in table[1] below.

Table 1: Experimental System Configuration



Item	Value
OS Name	RedHat 7.2 x64 bit
System Type	x64 bit
Processor	Intel(R) Xeon(R) CPU ES-2630 v4 @2.20Ghz, 2201Mhz, 10 Core(s), 20 Logical
SMBIOS version	2.8
Physical Memory	32.0GiB

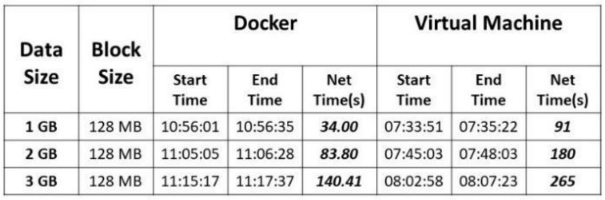
3 On Container:- Docker has been used for containers. Total of 5 containers have been utilized in this case. In which there were two datanodes and task tracker(same system was used for both datanodes and tasktrackers) , one namenode, one resource manager and one client. From client system all the operation was performed. First data was generated on client system using teragen program which comes along with Hadoop. All the five container machines were installed on same redhat OS of which system configuration is listed in table[1] above.

Tests stages:-

The test was divided into three Stages:-

1. This stage studies the time taken by system to generate different amount of data. In each case the time taken by each system has been evaluated carefully. The result of this test is given in table[2] below.

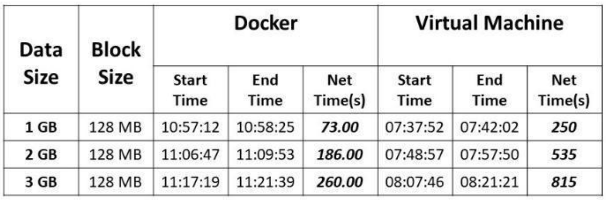
Table 2: Time Taken in Teragen Data Generation



Data Size	Block Size	Docker			Virtual Machine		
		Start Time	End Time	Net Time(s)	Start Time	End Time	Net Time(s)
1 GB	128 MB	10:56:01	10:56:35	***34.00***	07:33:51	07:35:22	***91***
2 GB	128 MB	11:05:05	11:06:28	***83.80***	07:45:03	07:48:03	***180***
3 GB	128 MB	11:15:17	11:17:37	***140.41***	08:02:58	08:07:23	***265***

2. This stage studies the time taken by system to sort different amount of data. Since this stage is main stage of processing of data so this stage takes slightly more time than other two stages. The result of this test is given in table[3] below.

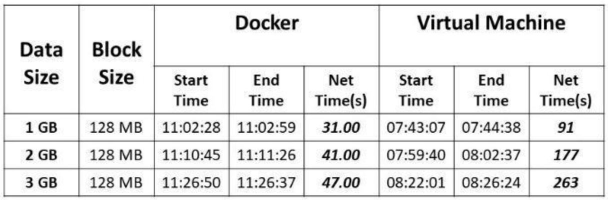
Table 3: Time Taken in Terasort Data Sorting



Data Size	Block Size	Docker			Virtual Machine		
		Start Time	End Time	Net Time(s)	Start Time	End Time	Net Time(s)
1 GB	128 MB	10:57:12	10:58:25	***73.00***	07:37:52	07:42:02	***250***
2 GB	128 MB	11:06:47	11:09:53	***186.00***	07:48:57	07:57:50	***535***
3 GB	128 MB	11:17:19	11:21:39	***260.00***	08:07:46	08:21:21	***815***

3. This was the final stage of evaluation which studies the time taken by system to validate the output generated by above stage. The result of this test is given in table[4] below.

Table 4: Time Taken in Teravalidate Data Validation



Data Size	Block Size	Docker			Virtual Machine		
		Start Time	End Time	Net Time(s)	Start Time	End Time	Net Time(s)
1 GB	128 MB	11:02:28	11:02:59	***31.00***	07:43:07	07:44:38	***91***
2 GB	128 MB	11:10:45	11:11:26	***41.00***	07:59:40	08:02:37	***177***
3 GB	128 MB	11:26:50	11:26:37	***47.00***	08:22:01	08:26:24	***263***

TEST RESULTS AND ANALYSIS

In test 1 three different random data volumes have been generated. To generate 1 GB of data the time taken by virtual machine was 91 seconds and time taken by docker was nearly one third i.e. 34 seconds. And in case of 2 and 3 GB Data generation, it nearly tripled the time taken by Dockers. There was also a huge difference of time that has been seen in the case of sorting of data. VM took 250 seconds to sort 1 GB of data while on docker this was done only in 73 seconds. And also in case of 2 GB, the time taken by virtual machine was 535 seconds which is nearly three times the time taken by Docker containers. This time difference has also been seen in the case of validation also.

Docker container took 31,41 and 47 in case of 1,2 and 3 GB of data while Virtual machine took 91, 177 and 263 seconds which is a huge difference. The above report clearly suggests the process is much faster in the case of the container than that of in the case of virtual machines. To run the cluster on docker it is also less complex as compared to running the cluster on VM. Since no different os setup is required in the case of the container the same above task can be achieved in much lesser time with lesser complexity involved.

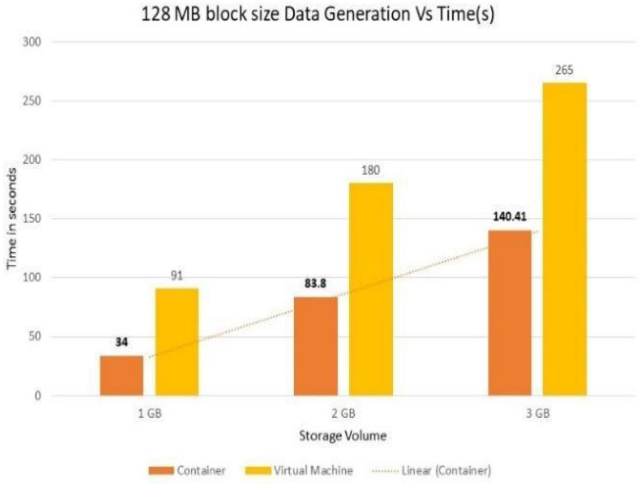


Figure 5: Data Generation plot Cotnainer Vs Virtualisation

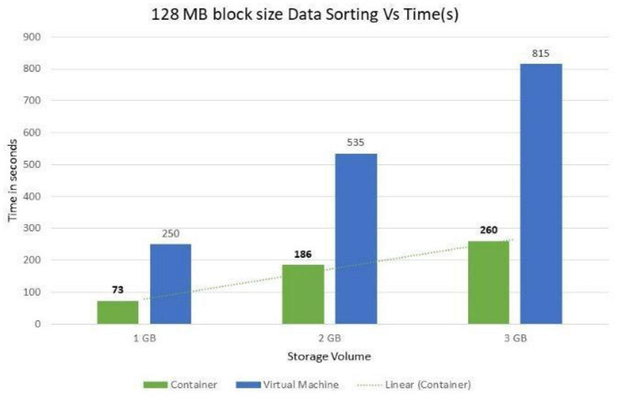


Figure 6: Data Sorting plot Cotnainer Vs Virtualisation

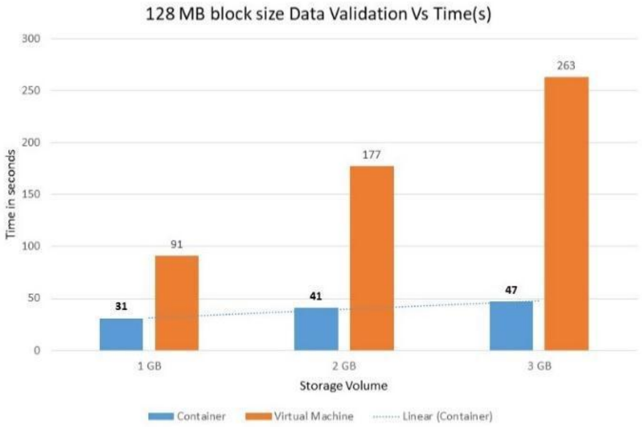


Figure 7: Data Validate plot Cotnainer Vs Virtualisation

CONCLUSION AND IMPACT

As researched and stated that container application run much faster whereas the virtual machine which runs on a indipendant OS runs slower. The current world uses the standard procedures to optimise the big data, which perhaps is the next big thing, but using Hadoop frameworks on containers, decreases the downtime of a system incase of failure or starting. The clusters gain super speed as they have a common kernel in case of a container, as the base OS launches the OS or the different VMs in form of a container which uses a common base. This results in the impact of much greater growth in IT industry, leading to great expansion of businesses which will reduce the downtime at the back-end and provide a dependable all time service to their clients.

REFERENCES

[1] Goasguen, S., 2014. Docker Cookbook: Solutions and Examples for Building Distributed ApplicationsBook. 2nd ed. USA: OREILLY.

[2] Nickoloff, J., 2016. Docker in Practice. 1st ed. USA: Manning Publications, Company.

[3] Welcome to Apache™ Hadoop®!. 2018. Welcome to Apache™ Hadoop®!. [ONLINE] Available at: http://hadoop.apache.org/. [Accessed 28 February 2018].

[4] White, T., 2015. Hadoop: The Definitive Guide. 4th ed. USA: O' Reilly.

[5] Karanth, S., 2015. Hadoop: The Definitive Guide. 1st ed. India: Packt Publishing - ebooks Account.

[6] Bashari Rad, B., 2015. An Introduction to Docker and Analysis of its Performance. 1st ed. Kuala Lumpur, Malaysia: Asia Pacific University of Technology and Innovation Technology Park Malaysia.

[7] Bholebawa, Idris Zoher, 2016. Design and Performance Analysis of OpenFlow-Enabled Network Topologies Using Mininet. International Journal of Computer and Communication Engineering, 5, 419-429.

[8] Xueyuan Wang, Brian Lee, Yuansong Qiao,2017, Experimental Evaluation of Memory Configurations of Hadoop in Docker Environments.IEEE,5

[9] Janki Bhimani, Zhengyu Yang, Miriam Leeser, and Ningfang Mi; Dept. of Electrical & Computer Engineering,NortheasternUniversity,360 Huntington Ave., Boston, MA 02115; bhimani@ece.neu.edu,

[10] yangzy1988@coe.neu.edu, mel@coe.neu.edu, and ningfang@ece.neu.edu Accelerating Big Data Applications Using Lightweight Virtualization Framework on Enterprise Cloud.

[11] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” Communications of the ACM, vol. 51, no. 1, pp. 107– 113, 2008.

[12] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on. IEEE, 2010, pp. 1–10.

[13] A. G. Shoro and T. R. Soomro, “Big data analysis: Apache Spark perspective,” Global Journal of Computer Science and Technology, vol. 15, no. 1, 2015.

[14] D. Merkel, “Docker: lightweight linux containers for consistent development and deployment,” Linux Journal, vol. 2014, no. 239, p. 2, 2014.

[15] J. Bhimani, J. Yang, Z. Yang, N. Mi, Q. Xu, M. Awasthi, R. Pandurangan, and V. Balakrishnan, “Understanding Performance of I/O Intensive Containerized Applications for NVMe SSDs,” in 35th IEEE International Performance Computing and Communications Conference (IPCCC). IEEE, 2016.

[16] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, “An updated performance comparison of virtual machines and linux containers,” in Performance Analysis of Systems and Software (ISPASS), 2015

IEEE International Symposium on. IEEE, 2015, pp. 171–172.

[17] R. Dua, A. R. Raja, and D. Kakadia, “Virtualization vs containerization to support PaaS,” in Cloud Engineering (IC2E), 2014 IEEE International Conference on. IEEE, 2014, pp. 610–614.

[18] R. Bosagh Zadeh, X. Meng, A. Ulanov, B. Yavuz, L. Pu, S. Venkataraman, E. Sparks, A. Staple, and M. Zaharia, “Matrix computations and optimization in apache spark,” in Proceedings of the 22nd ACM SIGKDD

[19] International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 31–38.

[20] H. Gao, Z. Yang, J. Bhimani, T. Wang, J. Wang, B. Sheng, and N. Mi, “AutoPath: Harnessing Parallel Execution Paths for Efficient Resource Allocation in Multi-Stage Big Data Frameworks,” in 26th International Conference on Computer Communications and Networks (ICCCN). IEEE, 2017.

[21] K.-T. Seo, H.-S. Hwang, I.-Y. Moon, O.-Y. Kwon, and B.-J. Kim, “Performance comparison analysis of linux container and virtual machine for building cloud,” Advanced Science and Technology Letters, vol. 66, no. 105-111, p. 2, 2014.

[22] D. Bernstein, “Containers and cloud: From lxc to docker to kubernetes,” IEEE Cloud Computing, vol. 1, no. 3, pp. 81–84, 2014. [12] T. P. (BlueData), “Lessons learned from running spark on docker.”

[23] Z. Lei, H. Du, S. Chen, C. Zhu, and X. Liu, “Dcspark: Virtualizing spark using docker containers,” in Audio, Language and Image Processing (ICALIP), 2016 International Conference on. IEEE, 2016, pp. 13–18.

[24] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster computing with working sets.” HotCloud, vol. 10, no. 10-10, p. 95, 2010.

[25] G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, and C. A. De Rose, “Performance evaluation of container-based virtualization for high performance computing environments,” in Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on. IEEE, 2013, pp. 233–240.

[26] N. Regola and J.-C. Ducom, “Recommendations for virtualization technologies in high performance computing,” in Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on. IEEE, 2010, pp. 409–416.