Docker in Support of Big Data Applications and Analytics

Anand Sriramulu
Indiana University
107 S Indiana Ave
Bloomington, Indiana, USA 47405
asriram@iu.edu

ABSTRACT

Different uses cases on how docker can improve the performance of Big Data applications.

KEYWORDS

i523, hid338, Data Science, Docker, Containers, Big Data Analytics, Cloud Computing

1 INTRODUCTION

The rapid acquisition of big data and development of computationally intensive analysis has led to need for novel approaches to software deployment. Containers are a solution which allows the software to run on any environment, whether it could be a developer's machine or a testing environment or servers running in data center or cloud environment.

Docker is a newer type of container technology and an open source platform for developers and sysadmins to build, ship, and run applications.

As per the article stated in [?], "Docker has the same concept as a container used for cargo ships: a standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery. In between, containers can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another."

Docker provides lightweight environment that isolates the piece of software and the resources such as CPU, memory, disk, etc and make it protable and self-contained.

2 DOCKER BENEFITS

Docker is a widely used container and it's being very matured compared to the others. I'll outline the top five benefits of using the ever-growing platform. [?]

2.1 Continuous Deployment and Testing

Continuous deployment is a DevOps process in which the applications or features from continuous integration and deployed to production environment. Docker helps both development and devops and make sure it's consistency across environments. Docker containers are configured to maintain all configurations and dependencies internally, so the same container developed and tested in the development environment can be used in production without any manual intervention.

The developer need not worry about the environment as he can develop or test any product upgrades, maintenance releases, new features in a docker container and release the images to different production servers. This is a great advantage as it saves lot of time with no errors due to the deployment issues.[?]

2.2 Multi-Cloud Platforms

Docker being widely used container solution, all the cloud computing provides supports it, so the portability being greatest strength with docker. That means, the docker image running on AWS can be switched Azure in which the docker solution is free from Platform As a Service vendor lock and provides the level of abstraction from the infrastructure layer. List of hosting provides supporting docker includes AWS, Microsoft Azure, Digital Ocean, Exoscale, Google Compute Engine, OpenStack, Rackspace, IBM Softlayer, etc. [?]

2.3 Environment Standardization and Version Control

Docker support version control as like GIT or TFS repositories. The docker images can be version controlled and if there any issues in the deployment, it can be roll-backed to the previous version. The process of rollback is quick and easy when compared to VM backup and image creation processes.[?]

2.4 Isolation

Since docker is a container, it's isolated from other containers and resources in the same host. Docker also make sure each container has it's own resource been allocated and isolated from the other containers. This gives the benefit on each container can run on its own application stack and be managed. So if an application is not needed, it can be removed by deleting the container and it won't leave any temporary or container related files on the host system. As mentioned earlier, each container has assigned with allocated resource, the docker make sure it won't be exceeded and hence there will no issues related to the performance or down time of the other applications in the same host.[?]

2.5 Security

As the containers are isolated, Docker make sure that the applications that are running on containers have control only within their container. So no container can look into the processes of other container. Each container will have its own resources ranging from processing to network stacks which is great benefit as if there any impact to an application related to security it won't impact the other applications.

[?]

3 BIGDATA AND DOCKER

Big Data is one of the big trends in IT of recent years. Majority of CIOs are investing more time in collecting and managing data for the business needs.

CIOs and IT Operations have a common goal: prepping their IT infrastructure to manage overflowing data and growing revenue by making better use of the data they collect.

It's huge struggle for them to find a right system to get the relevant information needed for the business managers to make important decisions.

Without big data, they find it's difficult task to arm the organization with the technology stack, skilled professional and resources for the business intelligence to manage the data deluge.

3.1 Use Docker To Avoid Dependency Issues

Each developer might have different set of big data tools and not to mention all the dependencies required, which then must be distributed to each machine in a cluster.

Companies assume this situation is manageable, but get enough developers on the same cluster and it doesn't take long for one tools requirements to break another. This will cause all the dependencies issues.

In this situation, there are two choices - get an entire development team to standardize on a common toolset, or use Docker. Docker allows each tool to be self contained, along with all of its dependencies. This means that a application can have different jobs use different versions of the same tool without a conflict.

This frees up your DevOps team to use the best tools for the data processing job, or set up entirely new systems and drive incredible scale and efficiency.

3.2 Reduce Reliance On MapReduce Experts With Pachyderm

For sysadmins that have a large amount of data to analyze, the go-to method has typically been to run MapReduce queries on Hadoop. This typically requires specialist programmers who specialize in writing MapReduce jobs, or hiring a third party such as Cloudera.

This typically means that Big Data initiatives require a lot of co-ordination internally and require resources that are beyond the reach of even large enterprises who do not have that kind of expertise on tap.

Alternatively, Pachyderm is a tool that allows programmers to implement a http server inside a Docker container, then use Pachyderm to distribute the job. This has the potential to allow sysadmins to run large scale MapReduce jobs quickly and easily to make product level decisions, without knowing anything about MapReduce.[?]

Pachyderm has the ambition of replacing Hadoop entirely - whether it achieves that remains to be seen, but it certainly looks like it will be a significant player in the next generation of data processing

3.3 Run Scheduled Analytics Using Containers With Chronos

By reading the above section, it's evident that the containers are a great way of deploying services at scale and giving isolation to services that run on the same host and improving utilization, but Docker can also be used for batch processing as well.

The latest release of the Chronos job scheduler for Mesos allows you to launch Docker instances into a Mesos cluster. This provides developers and sysadmins with the ability to run scheduled analytics jobs using containers.[?]

Chronos allows you to schedule Docker containers to run ETL, batch and analytics applications without manual setup on your cluster nodes. One of the neat features of Chronos is that it will also produce a dependency graph between scheduled jobs that depend on each other, so they only run if the previous job is successful.

Chronos and Marathon combine really nicely to provide orchestration for a container infrastructure.

3.4 Provision A Big Data Dev Environment Using Ferry

Ferry allows you to create big data clusters on the local machine (and AWS). The beauty of Ferry is that it allows anyone to define a big data stack using YAML, and then share it with other developers using a Dockerfile. As per the article [?], "Setting up a Hadoop cluster is as simple as:

```
backend:
- storage
personality: 'hadoop'
instances: 2
layers:
- 'hive'
Connectors:
- personality: 'hadoop-client'
```

Get started by typing

```
ferry start hadoop"
```

This will create a two node Hadoop cluster and a single Linux client. This can be customized at runtime or defined using a Dockerfile. Ferry is great for developers who want to get up and running with a big data environment using a test AWS box, developers that need a local big data dev environment, or users that want to share Big Data applications.

Running Ferry on AWS also has several advantages over something like Elastic MapReduce, such as not tying you to a single cluster of a single type (such as Hadoop).

3.5 Run Big Data As Microservices With Coho

When we talk to enterprise customers about Big Data processing, there are one or two recurring themes. For example, in healthcare, there are frequent workflows where new data triggers a new action.

Taking transcoding as an example. When a new image is pushed to the storage system, a transcoding workflow will take place, reading the data back to a client machine or VM, transcoding it, and then writing the results back to storage. This can mean that the data has to cross the network three times!

In Big Data environments, data might be pushed out to a separate HDFS-based analytics system, only to be pushed back to the enterprise system when the job has been run.[?]

Coho has worked on a storage-integrated tool that allows developers and DevOps teams to think specifically about workflows as operations on data, and for them to be embedded in the storage system.

These resulting extensions can then run efficiently and transparently at scale as the system grows. This theoretically allows presentation layers to be built on top of existing data, for the system to be extended with audit and compliance functionality and for complex, environment based access controls to be built.

4 CONCLUSIONS

The complex nature of big data and the tools used to analyze these data sets makes efficient processing difficult with standard environments. As noted above, the use of emerging technologies such as Docker in combination with automated workflows may significantly improve the efficiency of data processing in data analytics. With the growing number of open data projects, use of these techniques will be necessary to take advantage of available computational resources.

paraphrase and cite

While performance and pipeline efficiency were key components of this implementation, Docker containers also allow for application isolation from the host operating system. Since many big data tools have complex sets of dependencies and are difficult to build from source, the ability to deploy containers with different operating systems and dependency versions to the same host decreases the amount of effort needed to being analysis. For example, the cgDownload utility is distributed as a compiled binary for use on CentOS 6.7, but can only be deployed on CentOS 7 when built from source, which requires a significant amount of manual configuration. With the use of containers allowed the deployment of each utility on its natively supported operating system, which improves stability and decreases the potential for dependency conflicts among software applications.

paraphrase and cite

Several other tools exist for the orchestration of containerized applications, such as Kubernetes and Docker Swarm. For complex platforms, these tools can be used to deploy containers across hardware clusters and to integrate networking and storage resources between containers. However, these applications work strictly at the container level and do not inherently provide application-level workflows as presented here. Additional implementation experience about the use of these tools within high-performance clusters may provide valuable insights about the scalability of these tools within data analytics workflows.

Because of the subsequent increase in analysis throughput, use of these tools means that big data analyses can be done even with limited local computational capacity. Finally, use of container technology can improve pipeline and experimental reproducibility since preconfigured applications can be readily deployed to nearly any host system. While many factors can impact reproducibility, the use of containers limits variability due to differences in software environment or application configuration when appropriately deployed. The continued use of emerging technology and novel approaches to software architecture has the potential to increase the efficiency of computational analysis in big data.

ACKNOWLEDGMENTS

The author would like to thank Dr. Gregor von Laszewski and the Teaching Assistants for their support and valuable suggestions.