Lightweight Virtualization in Cloud Computing for Research

Muhamad Fitra Kacamarga, Bens Pardamean, and Hari Wijaya

[1] Bioinformatics & Data Science Research Center, Bina Nusantara University
Jakarta, Indonesia
bpardamean@binus.edu

Abstract. With the advancement of information technology and the wide adoption of the Internet, cloud computing has become one of the choices for researchers to develop their applications. Cloud computing has many advantages, particularly the ability to allocate on-demand resources without the need to build a specialized infrastructure or perform major maintenance. However, one of the problems faced by the researcher is the availability of computer tools to perform research. Docker is a lightweight virtualization for developers that can be used to build, ship, and run a range of distributed applications. This paper describes how Docker is deployed within a platform for bioinformatics computing.

Keywords: Virtualization, cloud computing, Docker, research.

1 Introduction

Computer programs are becoming more essential to many aspects of scientific research. The steps of the scientific process, from data collection, analysis, evaluations, and conclusions rely increasingly more on computational methods. With the advancement of computing technologies and the wide adoption of the Internet, computing resources have become cheaper, more powerful, and more prevalent, leading to a much higher availability than ever before [1]. One the impacts of these advancements is the presence of a new computing paradigm model called the cloud computing, in which resources (computing and storage) can be distributed as services that can be leased to customers through the Internet at an on-demand and as-needed term [2]. Within a cloud computing set-up, users utilize the resources provided by the infrastructure source as needed and pay only for those items [3].

Cloud computing has become a substantial tool of choice for researchers since it allows for the performance of complex computations and the exploration of new projects without the up-front investment in an expensive, customized infrastructure [4]. Additionally, researchers often do not have enough computational tools or time or expertise level to implement installations for a state-of-the-art data analysis application from scratch [5]. For instance, in the field of bioinformatics, state-of-the-art computational tools and algorithms for applications on biological, medical, and health data are essential for collection, sharing, and analysis [6]. Therefore, a complete computational environment that would assist collaboration between researchers is needed [7].

R. Intan et al. (Eds.): ICSIIT 2015, CCIS 516, pp. 439–445, 2015.
DOI: 10.1007/978-3-662-46742-8_40

Docker is a lightweight virtualization for developers that can be used to build, ship, and run a range of distributed applications [8]. It can be used to build a *system image* that contains state-of-the-art data analysis application, which can then be shared with researchers involved in the project. This paper describes how Docker is deployed within a bioinformatics computing platform in cloud computing.

1.1 Cloud Computing

Cloud computing is a computing model that is an elastically scalable, virtualized system with the ability for rapid provision with minimal management effort over the Internet [9]. Google, Amazon, and Microsoft are market leaders for the cloud computing industry. The emergence of cloud computing has made several compelling features that makes it attractive to users:

1. No up-front investment: building a large-scale system would need a lot of investments in information technology infrastructure. The cloud computing uses a pay-as-go pricing model so users do not need to build the infrastructure themselves.
2. Elastic Infrastructure: the infrastructure can dynamically scale up or down based on request. Users can easily expand its infrastructure to a larger scale to handle rapid increase in service demands and shrink down when demand decreases.
3. Lower operating cost: resources in a cloud environment can be easily allocated and de-allocated, allowing users to manage resources more effectively and efficiently.
4. Easy Access: services provided in the cloud are generally web-based, rendering easy access through Internet connection. Users can also manage their resources using the cloud service provider's management console that can be accessed via Internet.
5. Reducing business risks and maintenance costs: cloud computing has access availability with high guaranteed uptime. By outsourcing the information technology infrastructure to the cloud, users can reduce its business. Users also do not need to hire staff and hardware maintenance since this task is covered by the cloud service provider.

Cloud computing provider offers three service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). This example of service models can be seen in Figure 1.

1. Infrastructure as a Service (IaaS) provides services for customer to access computing resources in a virtualized environment. These computing resources include computing unit, storage, network, and other fundamental computing resources.
2. Platform as a Service (PaaS) delivers services as a computing platform and includes operating system, programming language execution environment, and other tools for designing, building, and deploying the customer's application into the cloud infrastructure.
3. Software as a Service (SaaS) provides customers with access to the hosted applications on the cloud infrastructure that is managed by the vendor or the cloud service provider.

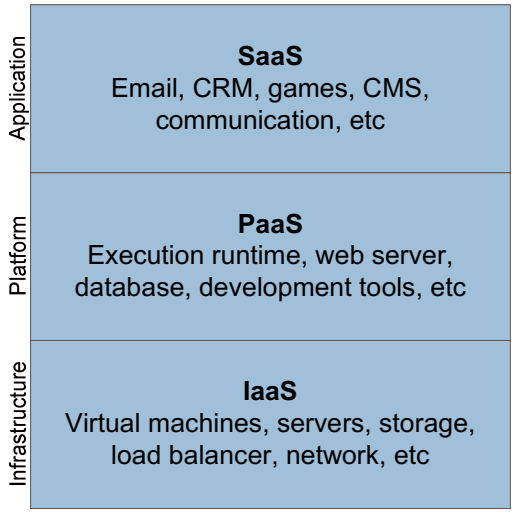

Fig. 1. An example of the cloud computing service models

Researchers can take advantage of the cloud computing service models depending on their research domain. There are many cloud computing providers from which to choose, such as Amazon Web Services, Google Cloud Platform, Microsoft Windows Azure, etc. Although each of them has unique components to provide different services, they also have several functionalities that are the same with one another. For example, Amazon EC2 provides the same service as Google Compute Engine, which provides virtual server instance. However, switching from one service provider to another is not easy in most cases due to *vendor lock-in* [10]. This is mainly due to dependencies and proprietary formats found within the underlying cloud infrastructures.

1.2 Docker

Docker is a lightweight virtualization based on Linux Containers (LXC) that can completely encapsulate an application and its dependencies within a virtual container [8]. This container can run on any Linux server enabling it to run on any premises (public cloud or private cloud) that has a Linux operating system. LXC is an operating system level virtualization technology that creates a sandbox virtual environment in Linux without the overhead of a virtual machine [5]. The overhead of a Docker's container is much smaller than that of a virtual machine because it replicates only the libraries and binaries of the application that is being virtualized [8]. Docker extends the LXC technology, leading to easier usage and simpler ways to perform versioning, distributing, and deploying. A Docker image can be transferred from one Docker host to another. Furthermore, it can be exported, archived, and run anytime in the future with the assurance of a similar computational environment. The comparison between a virtual machine and Docker architecture is shown in Figure 2.

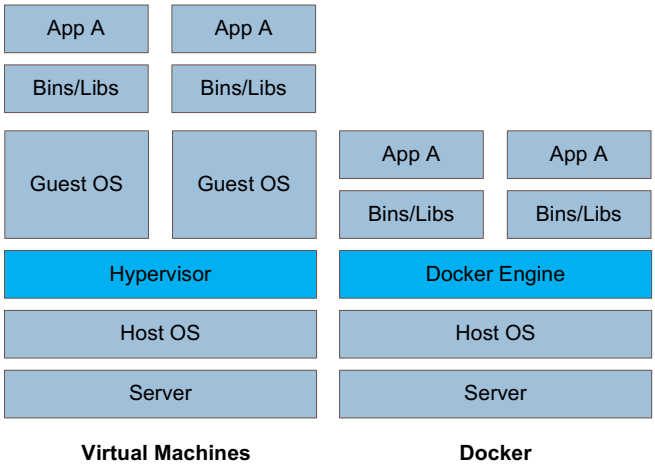

Fig. 2. Comparison between a virtual machine and the Docker architecture

Other than providing a virtualized and consistent computational environment, Docker provides many features that makes it an attractive sharing tool for research [5, 11]:

1. Docker images can be built by using text files (a Dockerfile) containing a set of instructions that commands Docker how to build its image. This approach allows Docker image to be versioned, shared, and re-built by others.
2. Docker has its own repository, similar to a Git repository. The Docker image can be easily shared with others via hosted repository, the Docker Hub.
3. The contents in a Docker container are restored to their original condition every time the container is launched. This approach makes a Docker container has a consistent computational environment.
4. Data, documentations, and files can be packaged within the Docker image, allowing for its use for sharing an entire computational experiment.
5. Docker has a large, active user base that provides a community-based information for troubleshooting.
6. Docker containers only use active resources, creating minimal overhead when running a Docker container.
7. Directories (folders) on the host's system can be easily mounted into a Docker container, rendering a seamless data sharing process between the host and the container.

There are two ways to build Docker containers: import from *tarball* file or import from a Dockerfile. Building a container from *tarball* file allows the researcher to create a complete container without performing installation steps. This method is similar with building a Virtual Machine from a *Snapshot* image file. On the other hand, building a container from a Dockerfile requires a series of installation commands (e.g., download application, libraries, etc.). Dockerfile is a text document that contains all commands of installation to build a Docker image. This method is more preferable when the researcher wants set up their computational environment within a

container. By installing from a Dockerfile, researchers can choose specific software for installation in the container.

2 Methods

This study describes a condition in which a researcher has existing virtual server instances but still requires a wide range of bioinformatics analysis. A computing platform can be used for data analysis ([R], Python, Perl) to perform a genome-wide association study (via PLINK). The system consists of a main installation script and a set of installation files. Users can edit the main installation script in order to select the bioinformatics software and create their own customized version. The objective for this system is to simplify the following processes: software selection, automatic new container establishment with the specified software, and deployment on the researcher's existing cloud virtual server instance.

3 Results

The bioinformatics computing platform was implemented using Docker. It is based on the Debian operating system. The researcher began platform deployment through a Linux virtual server instance. Then Docker was installed on its instance, followed by the installation package download. This installation package includes a Dockerfile, as well as other installation files. The researcher was also able to edit the Dockerfile to customize which bioinformatics tools had to be included in its container. In this Dockerfile, software packages were categorized based on their installation method. A set of bioinformatics tools including [R], Python, Perl, PLINK, and Cython were installed by the *apt-get* command. On the other hand, tools like Affymetrix Power Tools, Java, and Eigensoft were installed from the installation files provided within the installation package. When the Dockerfiles were executed, the Docker performed retrieval and installation of the selected software from the repositories and installation files, building the image into a fully functional bioinformatics computing container.

Researchers also had the ability to share the edited Dockerfile in order to distribute their own customized version of a bioinformatics platform with the selected software and data. Other researchers could replicate the customized bioinformatics platform by retrieving this edited Dockerfile then executing it in their cloud. Researchers could upload data from its local desktop computer or other source to the host. To share data from the host to a Docker container, copying the data to the mounted directory allows for the subsequent data retrieval from the Docker container. There was also the option to save a container with analysis result by exporting the container into *tarball* file, which could then be shared with other researchers for collaboration. Using the *tarball* file, the collaborators could create a whole identical container on their cloud then use new data with different parameter to generate new results. These steps are depicted in Figure 3.

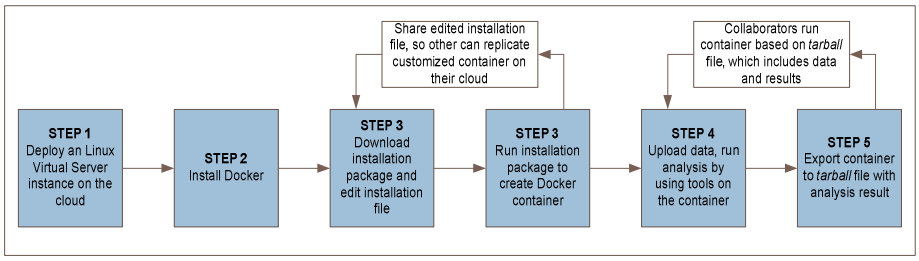

Fig. 3. Bioinformatics computing platform workflow

While research using the cloud computing environment supports reproducible research [12], many researchers worked locally, primarily with software installed on their local computer. Researchers would transform their work to become cloud-based only when collaborative tasks were performed or an increase in computational power was needed. Working locally allowed a researcher to exchange files and debugging faster. Docker is available on most major platform. Thus, a researcher could install Docker and build up the bioinformatics platform container on a local computer. The platform could also be used locally. The collaborators could also import a customized bioinformatics platform then run it locally for testing purpose.

4 Discussion

Several studies (Krampis [13] and Dudley & Butte [12]) have proposed ways to share computational environment for research. There are two dominant approaches: workflow software and virtual machines [11]. Workflow software provides solutions to standardize the creation, representation, and sharing of computational workflow that bind diverse software tools together into a single analysis [12]. This workflow software is often adopted by well-funded collaboration research through which they receive substantial support from the communities. Most workflow software has relatively low adoption due to proprietary formats and interfaces [12].

Virtual machine (VM) offers a more direct approach. The VM approach, operating system, software tools, and databases are packaged into a single digital image that is ready to be used. This approach is used by Krampis [13] and Dudley & Butte [12] to share virtual machine images that will run on the cloud as a platform for doing research. But, there are some drawbacks using VM images such as large file size, the need for system administration knowledge, and the difficulties to track versions [5]. The implementation of VM in cloud has a serious problem when users want to switch from one service provider to another due to vendor *lock-in* [10].

5 Conclusion

This study described how Docker was deployed for a bioinformatics computing platform in cloud computing. The motivation for this study was the primary concern on

the lack of time and expertise on a researcher's part to install and implement state-of-the-art data analyses application from the scratch. Our objective was to simplify the processes of software selection, automatic building of a new container with the specific softwares, and deployment on a researcher's existing cloud virtual server instance. A script (Dockerfile) that allows bioinformatics platform to be easily reproduced and updated was developed since the script provides an exact instruction on how the image was built. Furthermore, researchers also has the option to save the container with analysis result by exporting the container into a *tarball* file. This approach is an easy way for sharing a complete computational environment for researchers.

Acknowledgements. We would like to thank James W. Baurley, PhD from BioRealm Research for sharing the bioinformatics computing platform.

References

1. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. internet Serv. Appl. 1, 7–18 (2010)
2. Marinescu, D.C.: Cloud Computing and Computer Clouds (2012)
3. Weinhardt, C., Anandasivam, W.A., Blau, B., Borissov, N., Meinl, T., Michalk, W.W., Stößer, J.: Cloud computing–a classification, business models, and research directions. Bus. Inf. Syst. Eng. 1, 391–399 (2009)
4. Rehr, J.J., Vila, F.D., Gardner, J.P., Svec, L., Prange, M.: Scientific computing in the cloud. Comput. Sci. Eng. 12, 34–43 (2010)
5. Chamberlain, R., Invenshure, L.L.C., Schommer, J.: Using Docker to support reproducible research (2014)
6. Kesh, S., Raghupathi, W.: Critical Issues in Bioinformatics and Computing. Perspect. Health Inf. Manag. 1, 9 (2004)
7. Hu, Y., Lu, F., Khan, I., Bai, G.: A Cloud Computing Solution for Sharing Healthcare Information (2012)
8. Hykes, S.: What is Docker?, `https://www.docker.com/whatisdocker/`
9. Padhy, R.P., Patra, M.R., Satapathy, S.C.: X-as-a-Service: Cloud Computing with Google App Engine, Amazon Web Services, Microsoft Azure and Force. Com. Int. J. Comput. Sci. Telecommun. 2, 8–16 (2011)
10. Kratzke, N.: A Lightweight Virtualization Cluster Reference Architecture Derived from Open Source PaaS Platforms. Open J. Mob. Comput. Cloud Comput. 1, 17–30 (2014)
11. Boettiger, C.: An introduction to Docker for reproducible research, with examples from the R environment. arXiv Prepr 0846 (2014)
12. Dudley, J.T., Butte, A.J.: Reproducible in silico research in the era of cloud computing. Nat. Biotechnol. 28, 1181 (2010)
13. Krampis, K., Booth, T., Chapman, B., Tiwari, B., Bicak, M., Field, D., Nelson, K.E.: Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community. BMC Bioinformatics 13, 42 (2012)