

Exploring Docker Containers in a Heterogeneous Cluster

Ying Mao
(Small Grant Proposal)

Abstract

In the last decades, we have witnessed a spectacular information explosion over the Internet. Millions of users are consuming the Internet through various services, such as mobile applications, and online games. The service providers, at the back-end side, are supported by state-of-art infrastructures. Targeting on providing the services at scale, virtualization is one of the emerging technologies used in data centers and cloud environments to improve quality of services. In this project, we aim to develop a dynamic resource management scheme based on virtual containers. It collect the runtime job progress from the running tasks and allocate the resources dynamically to improve the overall system performance.

Keywords— Container Scheduling, Virtual Machine, Resource Management, Cloud Computing, Docker Containers, Microsoft Azure Proposal

Project Summary

Virtual machines are a widely-adopted virtualization method to provide services on the cloud. It isolates various resources, such as CPU, memory, so that machine can be shared by multiple users. In a large-scale system, however, providing services through virtual machines means running many duplicated instances of the same OS and redundant boot volumes. To address the limitations, containers are designed for deploying and running distributed applications without launching entire virtual machines. Instead, multiple isolated containers are sharing the host operating system and physical resources.

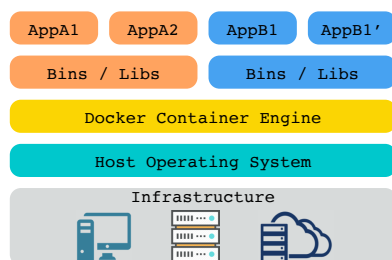


Figure 1: Docker Containers

Container is not a new concept. Unix-like operating systems have leveraged the technology for over a decade. However, new containerization platforms, such as Docker, make it enter the mainstream of application development. Docker introduces a way of simplifying the tooling required to create and manage containers. Fig 1 illustrates the structure of a physical machine that is hosting four Docker containers for two applications. *AppA* includes two services that are provided by *AppA1* and *AppA2*. *AppB* contains one service which is provided by two Docker containers—*AppB1* and *AppB1'*.

When developers deploy the applications into a production environment, it is difficult to achieve resilience and scalability on a single host. Typically, a multi-node cluster is used to provide the infrastructures for running containers at scale. Introduced by Docker, Swarmkit is an open source toolkit for container orchestration. Constructing a cluster of Docker containers, as a first step, Swarmkit needs to determine where to place each container. The default placement strategy—Spread— attempts to schedule a task based on the number of active containers on each node, which can roughly assess the available resources on it. This assessment, however, fails to reflect various nodes in a heterogeneous cluster setting. Considering the heterogeneity, the nodes in such cluster have diverse configurations in terms of memory, CPU and bandwidth. Therefore, running the same amount of containers on these nodes result in different experiences. Additionally, different containers target on various services. For example, containers that provide a deep learning service, such as TensorFlow, may require more computation resources, and web-service containers, e.g. Tomcat and Nginx, may need more memory and bandwidth resources. Our preliminary results indicate that, in a heterogeneous cluster, the Spread placement strategy fails to achieve stable performance under the extensive workloads, leading to serious service interruption.

In this project, we will investigate characteristics of Swarmkit. It contains two major phases.

- We will design and develop a new placement strategy that takes consideration of both workloads on each individual containers and real-time resource availability on the hosts. The new strategy will reduce the starting overhead and improve the system scalability.

- The system will be evaluated through various applications. Besides traditional services that are launched by standard Docker images, we are planning to build a new emulator for Internet of Things(IoTs) that combines physical devices and clouds, in which customized Docker containers represent the IoT devices on the cloud. This phase will collaborate with faculty members from other institutions.

Broader Impact

As a fast-moving technology, containerization and its applications draw a tremendous attention in the academic community. Although lots of work have been proposed on the system-level virtualization (e.g., Docker containers), the research on clusters of Docker containers management is still under-investigated. The completion of this project will serve as a basis in the community. Other researchers can use the platform to investigate new topics. Besides academic publications, all related source codes and datasets will be open-sourced. Furthermore, the faculty member will maintain an active research blog to share the ideas and reports of Microsoft Azure and containerization.

In addition, this proposed project, if awarded, will significantly inspire the students to conduct research at primarily undergraduate institutions. It creates a great opportunity for the faculty working with research students to design, implement and evaluate their innovative ideas on an industry-ready environment.

Resources

I'm fortunate to work with five research students and produced related research publications in this field [1–7]. Currently, the project is supported by an internal grant, SOSA-TCNJ, which provides both course release and the monetary award. Moreover, NSF Cloudlab has approved us to use their scientific infrastructures for small-scale experiments.

The proposed research will utilize various Microsoft Azure services, including Container Service, Container Registry, Cloud Services, IoT Hub, Machine Learning, and App Services. The awarded credits will be used by both our group and collaborative groups from other institutions when they run experiments in our proposed system. The following table illustrates core resources that we are requesting from Microsoft Azure. Depending on the specific experiments, we may spend the credits on other Azure services.

Azure Services	Quantity
Container Services	100,000 hrs
Container Registry	500 GB - 1 TB (Storage)
Cloud Services	100,000 hrs
IoT Hub	s1 / s2 (3-6 months)

References

- [1] Xuan Chen, Long Cheng, Cong Liu, Qingzhi Liu, Jinwei Liu, Ying Mao, and John Murphy. A woa-based optimization approach for task scheduling in cloud computing systems. *IEEE Systems Journal*, 2020.
- [2] Yuqi Fu, Shaolun Zhang, Jose Terrero, Ying Mao, Guangya Liu, Sheng Li, and Dingwen Tao. Progress-based container scheduling for short-lived applications in a kubernetes cluster. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 278–287. IEEE, 2019.
- [3] Wenjia Zheng, Yun Song, Zihao Guo, Yongchen Cui, Suwen Gu, Ying Mao, and Long Cheng. Target-based resource allocation for deep learning applications in a multi-tenancy system. In *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2019.
- [4] Wenjia Zheng, Michael Tynes, Henry Gorelick, Ying Mao, Long Cheng, and Yantian Hou. Flowcon: Elastic flow configuration for containerized deep learning applications. In *Proceedings of the 48th International Conference on Parallel Processing*, pages 1–10, 2019.
- [5] Ying Mao, Jenna Oak, Anthony Pompili, Daniel Beer, Tao Han, and Peizhao Hu. Draps: Dynamic and resource-aware placement scheme for docker containers in a heterogeneous cluster. In *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2017.
- [6] Hank H Harvey, Ying Mao, Yantian Hou, and Bo Sheng. Edos: Edge assisted offloading system for mobile devices. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2017.

- [7] Anil Acharya, Yantian Hou, Ying Mao, Min Xian, and Jiawei Yuan. Workload-aware task placement in edge-assisted human re-identification. In 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pages 1–9. IEEE, 2019.