A Docker-based HPC architecture for RS imagery processing

Fenghua Hu, Jingliang Zhang

Geovis Technology Co., Ltd, P.O. 101399, Beijing, China.
Email: hufh2004@gmail.com

Abstract. Remote sensing imagery processing requires strong computing demand. HPC can provides powerful computing capability and has been proved as an efficient approach for acceleration. To build and deploy a HPC is costly due to its complexity and need of dedicated hardware and this restricts its application in RS field. Docker is a container technology and can simplify software deployment and improve resource utilization. This paper proposes an architecture which is based on Docker and can lower the barrier to use HPC for average user. It greatly simplifies deployment of HPC as well as improve the resource utilization by sharing with other applications.

Keywords: RS imagery processing, Docker, HPC

1. Introduction
Over past decades, Remote sensing (RS) imagery data has been exploding[1,2]. A complicate process is needed for RS imagery before application and it requires strong computing demand[3]. High performance computing(HPC) has been proven an effective approach to accelerate this process[4]. Usually HPC is implemented as dedicated system, consisting of a cluster of dedicated and connected computer servers. It takes long time for advanced IT expert to build a HPC. This makes HPC costly and unaffordable for average user therefore largely restricts HPC’s application in RS.

Recently, Docker-based container technology is becoming more and more popular By running an encapsulated application in an isolated container, Docker can not only dramatically simplify software’s deployment, but also improve resource utilization. It has been adopted by thousands of large-scale systems[5]. Employing Docker to build and deploy a HPC cluster can address the issue fore-mentioned. It will lower the technical barrier for RS imagery processing. In this paper, we will propose an architecture which leverages Docker technology to build HPC system and give a prototype.

2. Approach
Docker is a container technology and opened source in 2013 and becoming increasingly widespread due to its simplicity and flexibility. We propose an architecture to build a HPC system for RS imagery processing, based on Docker technology. In our architectural solution, the HPC cluster consists of several components: a workflow manager, a resource manager and a scheduler. Workflow manager is responsible for handling job request. Workflow manager receives the request, breaks down it into a set of sub-tasks, then forwards to resource manager. Resource manager is responsible for resource allocation. Scheduler runs on every machine to collect status info and send to resource manager. Resource manager decides which machines will execute the sub-tasks. Scheduler gets sub-task information from resource manager, then runs corresponding algorithm code.
These components are respectively encapsulated as independent Docker images. Each image contains binaries for corresponding component. Besides HPC components, the algorithms for remote sensing imagery processing are encapsulated as images as well. Each algorithm image contains algorithm executable, plus some libraries, for instance, GDAL library for basic image processing, and MPI library for parallel computing, and CUDA for GPU acceleration. An image is an independent and self-contained execution environment thus won’t be interfered by other algorithm images and compatibility issues can be avoided.

All of Docker images, including HPC component images and algorithm images, are stored in a repository called image registry. When a HPC is started, it pulls required image from the registry, starts a container instance and loads the image into it then runs. The instance is stateless and can’t store any data and status. Data and status for an instance are stored in a distributed file system which shared by all of instances. Before an instance runs, its file system path storing data and status data, is automatically mapped into instance and mounted as a volume, so that the instance can access from inside. Multiple algorithm instances can be run on same physical machine or multiple machines to implement parallel imagery processing.

When user submits a RS imagery process request, he can specify how many instances for algorithm should be run so as to get parallelism. Workflow manager and resource manager will work together to allocate requested resource and start the instances. If currently available resource can’t satisfy requirements, the request will be queued. Once the requirement is met due to releasing resource or adding new server into cluster, the request will be re-scheduled and processed. With this approach, Docker instances can added or removed in a dynamic way.

All of HPC components can be run on any machines with available resource, and deploying a HPC can be done by simply running these images containing HPC components. When RS imagery processing finishes, instances will be terminated and resources will be reclaimed then allocated to other applications. The resource utilization can be improved.

3. Result and Conclusion
It took us one day to build a 16-server cluster in tradition way with an advanced IT expert’s help. While using Docker, in order to deploy a HPC environment, what we only need to do is to write a configuration file then run a script. This work can be done in only five minutes. Moreover, IT expert isn’t a must. For RS imagery processing, if more parallelism is needed, it can be simply implemented by running more instances. The process only needs 10-15 seconds. When no RS imagery process is running, these resources can be used by other kind of applications like web and Hadoop.

HPC has been proven an effective approach for RS image processing, but it’s costly to build and deploy a HPC cluster due to need of dedicated hardware and IT specialist. We propose an architecture for building a Docker-based HPC and it demonstrated an effective way to lower the cost and barrier for deployment as well as to improve resource utilization.

References
[1] Yan Ma et al, Remote Sensing Big Data Computing: Challenges and Opportunities, Future Generation Computer Systems, 51, pp.47-60(2015)
[2] Huadong Guo et al, Big Earth Data: a new challenge and opportunity for Digital Earth’s development, International Journal of Digital Earth, ,10:1, pp.1-12(2017)
[3] Karen E. Joyce et al, A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters, Progress in Physical Geography Vol.33(2), pp.183–207(2009)
[4] Craig A. Lee et al, Recent Developments in High Performance Computing for Remote Sensing: A Review, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol.4(3), pp.508-527(2011)
[5] Carl Boettiger, An introduction to Docker for reproducible research, ACM SIGOPS Operating Systems Review, Vol.49(1), pp.71–79(2015)