Toward Portable Distributed Deep Learning

Boonrak Kaewnopparat, Wudhichart Sawangphol, Assadarat Khurat
Faculty of Information and Communication Technology,
Mahidol University, Nakhon Prathom, Thailand
boonrak.kae@student.mahidol.ac.th, wudhichart.saw@mahidol.edu,
assadarat.khu@mahidol.ac.th

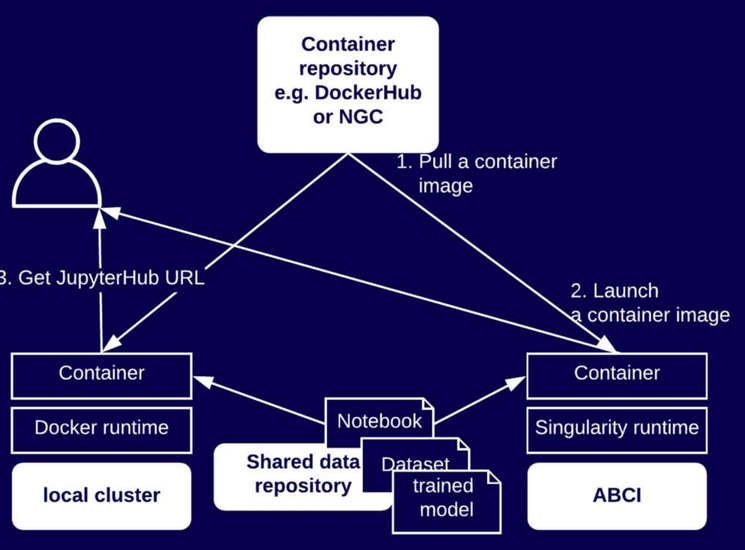
Ryousei Takano
Information Technology Research Institute, National Institute
of Advanced Industrial Science and Technology,
Tsukuba, Japan
takano-ryousei@aist.go.jp

Abstract

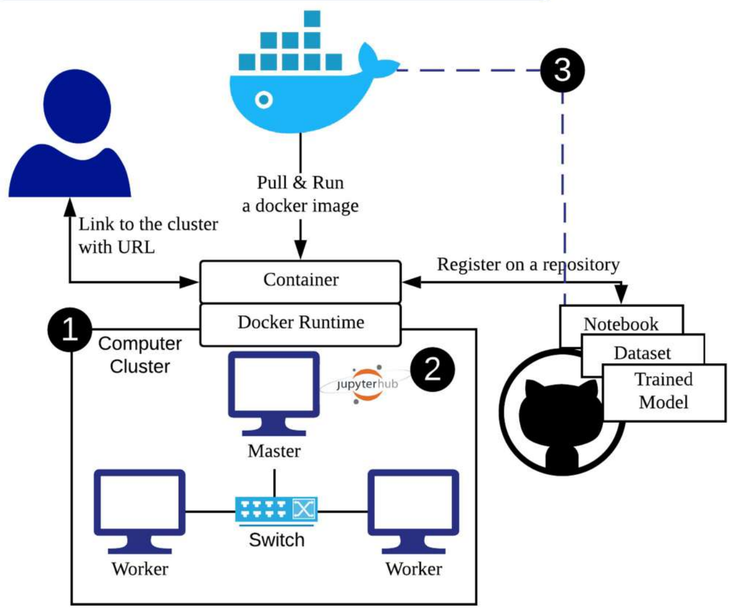
Deep learning has become a key feature in several industries as a revolution in a field including image classification, speech recognition and machine translation. Deep learning algorithms normally consume a long processing time and a huge amount of computing resources; thus, they need special environments and devices to execute, thus they need special environments and devices to execute. Consequently, distributed training comes across to solve time consumption problem of deep learning by training dataset parallelly, but it still has a minimum requirement for devices to run an application in which is fixed to be executed on the specific device. This leads to the portable distributed deep learning idea which will allow other computers to access on a computer cluster anywhere and execute deep learning applications as distributed training on the server locally so that they can shorten their processing time due to data parallelization on the cluster.

Portable Distributed Deep Learning

Bring your own distributed deep learning environment anywhere by leveraging container technologies



System Architecture



1 | Computer Cluster named 'mini-cluster'
The mini-cluster is constructed with 3 computers which have 1 GPU, Nvidia Geforce RTX2080Ti, in each node, each node is connected with NETGEAR Gigabit Ethernet Switch for communicating data.

2 | JupyterHub
The JupyterHub package is installed with the virtual environment inside the mini-cluster; then, it will generate an URL for other users to link with the JupyterHub on the mini-cluster

3 | Docker with GitHub or Dockerhub repository
Docker is used to wrap up the JupyterHub package after configuration as a docker image and to register it on a repository on GitHub or Dockerhub for rebuild it other computer clusters by pulling and running the image.

Experiment

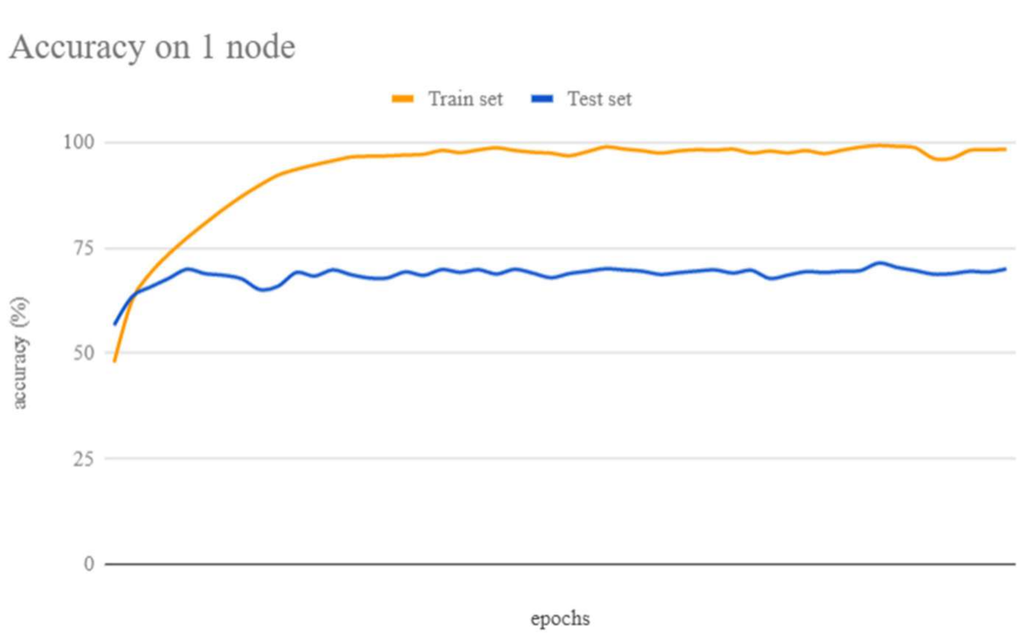
Dataset	CINIC-10
Docker Image	nvcr.io/nvidia/pytorch:19.08-py3
Deep Learning Framework	pytorch 1.2.0a0+f6aac41
Benchmark	https://github.com/ryousei/pytorch-cifar
CPU	Intel Core i5-8400
Memory	DDR4-2666 32GB
GPU	Nvidia GeForce RTX 2080Ti
OS	Ubuntu 18.04
Epoch	50
Batchsize	100
Learning rate	0.01

Result

On 1 node

Train set
Accuracy: 98.36%

Test set
Accuracy: 69.27%



References

Dataset from:
Darlow, L. and Crowley, E. (2018). CINIC-10 Is Not ImageNet or CIFAR-10. [online] Available at: https://arxiv.org/pdf/1810.03505v1.pdf

Background Information:
Petuum, Inc.. (2018). Intro to Distributed Deep Learning Systems. [online] Available at: https://medium.com/@Petuum/intro-to-distributed-deep-learning-systems-a2e45c6b8e7
Project Jupyter team, (2016). JupyterHub — JupyterHub 1.0.0 documentation. [online] Available at: https://jupyterhub.readthedocs.io/en/stable/
Docker Documentation. (2019). Docker overview. [online] Available at: https://docs.docker.com/engine/docker-overview/

Acknowledgement

This work was supported by National Institute of Advanced Industrial Science and Technology (AIST), Japan and Faculty of Information and Communication Technology, Mahidol University