

# DeCUVE: Deep Learning Cloud Unified Virtual Environment

Bo-Seon Kang

Visual Information Processing  
Korea University  
Seoul, Republic of Korea  
masati@korea.ac.kr

Chang-Sung Jeong

Department of Electrical Engineering  
Korea University  
Seoul, Republic of Korea  
csjeong@korea.ac.kr

**Abstract**—Recently, with advancement in deep learning technology in CCTV, drone and other various fields, large scale image processing environment becomes crucial and essential for fast real-time processing. In this paper, we shall present a distributed processing environment DeCUVE(Deep Learning Cloud Unified Virtual Environment) which provides scalability and provisioning for the deep learning inference. We shall show that deep learning training inference can be executed fast on our environment.

**Keywords**— *Deep learning, Distributed Environment, Image Processing;*

## I. INTRODUCTION

Deep learning is a core technology in machine learning and AI field, and can be used in various fields such as object detection, image processing, video processing and speech recognition[1]. When a system receives a large-scale image as input, and performs vast amounts of image processing computation, it takes a lot of delay time with a single computer, with performance degradation especially in deep learning applications.

There are many studies on how to use Deep learning in distributed environment[2]. However, there arises several problems on distributed environment using many computing resources. One of them is scalability. In a distributed environment using Deep learning, computing resources for huge demand must be provided[3]. Google's TensorFlow also conducting research on distributed processing. Recently, distributed processing has been added to TensorFlow, but some complications in placement of tasks on the Distributed TensorFlow[4]. Therefore, it is important to resolve placement issue which properly assigns tasks to available nodes to use for distributed deep learning. Also, there is a difficulty to configure the assigned physical nodes adapted for distributed deep learning environment.

In this paper, we shall present a new distributed processing platform which provides scalability and provisioning for the deep learning inference using DeCUVE(Deep Learning Cloud Unified Virtual Environment). It can easily build a cluster by assigning available nodes in cloud environment, and automatically provisioning various kind of physical nodes such as GPUs. We shall show that deep learning training inference

can be executed fast for object detection application in various image input devices.

The rest of this paper is organized as follows: Section II presents a related works, and section III presents a system architecture for the deep learning inference, and section IV presents the experimental result on our distributed deep learning environment. Finally, section V concludes the paper.

## II. RELATED WORKS

### A. Deep learning Inference

There are two major types of deep learning: deep learning training and inference. They have different objective. The former is the process of learning a model through a very large dataset. The latter is deep learning test which uses a trained model by receiving a smaller data set than training one, and using training trained finished model and when it received smaller than training used data set but various data, The process of perform provides available to user such as object tracking, image processing[5]. It is not possible to produce output only by the training process. Deep learning inference is essential for user to benefit and use.

We experiment using SSD algorithms based on VGG 16 suitable for large scale deep learning. SSD is available for classification of objects, and extracts feature maps very quickly through the multi-box detector technology. SSD network model is faster than Faster R-CNN, but slower than YOLO[9]. This problem is trying to solve by using distributed deep learning.

### B. Distributed Processing Environment

Google has released a Distributed TensorFlow that can be used in a distributed environment. Distributed TensorFlow basically constructs a cluster which consists of a number of tasks, a master and workers. Master is a Remote Procedure Call Service that creates a session to order commands[4]. We use DeCUVE to create distributed processing environment and to manage master and workers of distributed TensorFlow. DeCUVE can launch applications on various data processing models DeCUVE has advantages that it achieves optimal performance by estimating the optimal number of resources, and launches applications using the VM resource registered in

the cluster through auto resource provisioning function[6]. DeCUVE can be efficiently exploited as a distributed coordinator which can facilitate the deep learning inference on distributed TensorFlow.

### C. Container Virtualization

Virtualization service is designed to enable users to use and deploy services quickly and easily. Advent of container virtual technique leads to improve the limit of the previous virtual service, and provide more effective virtual environment. Existing virtualization service were wasteful of resources, since they simultaneously operated the Host OS and the Guest OS. Container virtualization is a lightweight operating system, and run on host system inside without using the Guest OS. It does not let the user feel necessity of just-in-time compilation. Various communication techniques such as standard IPC mechanism, pipes and socket can be used when communicating between containers. Using container virtualization instead of common virtualization can benefit from communication, security, and performance[7]. Use of Docker makes it easier to use images and some additional functions than the existing Law container[8].

## III. SYSTEM ARCHITECTURE

### A. System Architecture

In this section, we present a distributed processing environment for distributed deep learning. Its architecture is shown in Fig 1.

The platform for distributed deep learning consists of three layers: distributed deep learning layer(DDL), distributed parallel processing layer(DPL) and cloud infrastructure management layer(CIL). DDL consists of deep learning models and preprocessing manager. Deep learning models consists of various models for deep learning, and preprocessing manager processes image resize, request analysis and builder. Preprocessing Manager manages the following tasks. Image resize module is carried out to make the image a square. Request Analyzer looks into the path of the build file, image file when the request build carries out build, or analyzes the requests of the user application. And the parameter server information and slave node data are also included for the use of distributed TensorFlow. Once the analysis is finished, the Request Builder builds the task by including the data sent by the request analyzer. DPL consists of parallel models and resource agent manager, task manager. Parallel models consists of various models for parallel processing such as mapreduce, message passing. And resource agent manager generates the resource agent controller. The resource agent controller distributes the task requested by the user to the agent. When the resource agent controller is directly connected with the agent, the basic preparation to provide resources is complete.

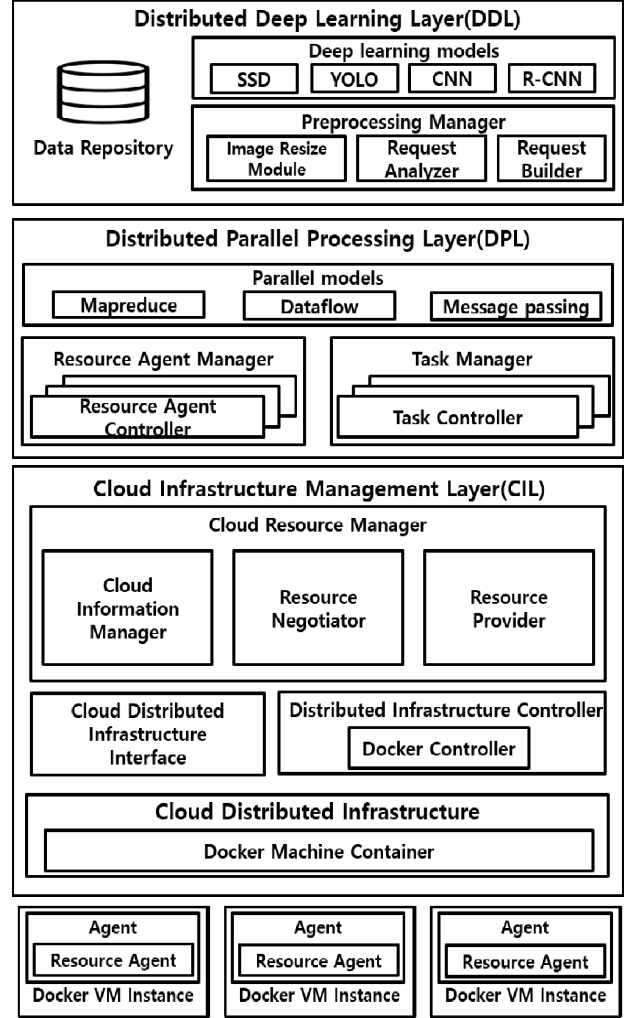


Fig. 1. System Architecture of DeCUVE

Cloud resource manager in CIL provides resources requested by user. To provide the requested resources, the cloud infrastructure information manager consistently manages the status of resources included in the cloud. When the node in the clustered DeCUVE has ended or an error has occurred, leading to the operation to be stopped, it must be removed from the resource pool so that it is not included in the provided resources. When appropriate resources are selected based on the information of resources managed by the resource negotiator, the resource provider provides the information of resources to be provided. Lastly, the distributed system infrastructure controller controls the docker controller that must be used by the system. The docker controller verifies and controls the status of the virtual machine generated through the container virtualization. All docker machine containers are controlled by the docker controller. The docker generates a container virtualization based virtual machine from the slave node and sets the quantity of the resources.

### B. System Features

There are several features in our system. These features make it possible for efficient use of the distributed deep learning inference. First, user sends a deep learning command to the master node via the application level interface, and the master node distributes the task for performing deep learning command to the slave nodes which in turn processes deep learning inference in parallel. Second, the user can select a distributed parallel model optimized for each deep learning model through the API. Third, our system can achieve the scalability, and solve placement issue. The placement issue which is the problem can be solved by the provisioning function of DeCUVE, and can append slave nodes without additional settings for clustering.

## IV. EXPERIMENT

PETS dataset is used for distributed deep learning inference to evaluate our system. It comprises has multiple pedestrians and a number of objects, making it appropriate to conduct an object tracking test.

### A. Experiment Environment

We make use of Google cloud platform as cloud infrastructure. Resource allocation and provisioning for distributed parallel processing model adapted for a given deep learning model is carried out automatically on cloud by DeCUVE.

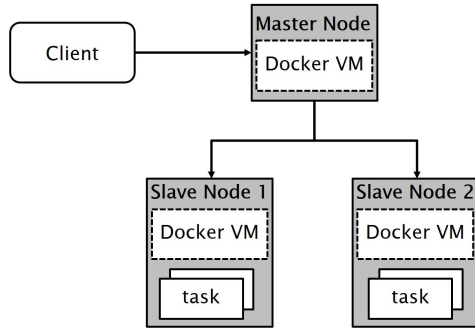


Fig. 2. System Structure for Distributed Deep Learning

On the Google cloud platform, each of three instances is generated by NVIDIA docker on a node with four vCPU(Intel(R) Xeon(R) CPU @ 2.50GHz) and memory of 15GB as well as Tesla K80 GPU by NVIDIA for experiment. One instance is used as a master for executing distributed deep learning model in orders, while the other two's for processing major tasks which are distributed by master node. We only consider the time taken for deep learning inference but not for image resizing.

### B. Experiment Result

In the experiment, we make use of the SSD300 model for object detection, and measure its speed on cloud infrastructure with two nodes for slave tasks.

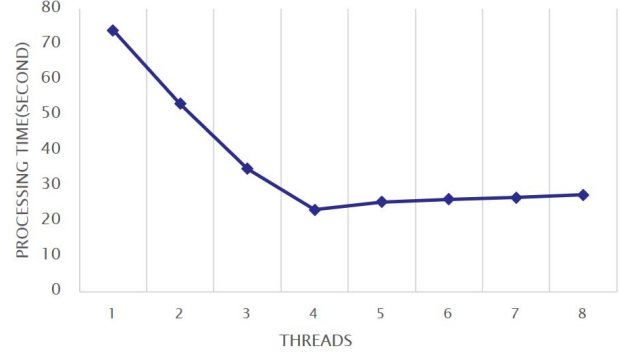


Fig. 3. Experiment Result of Distributed Deep Learning for View\_001

Fig. 3 shows the number of threads and processing time-for PETS with 248MB. For one thread, it takes approximately 0.09 seconds per image with 10.5 fps. As the number of thread increases, the time taken for object detection decreases up to 4 by 3.5 times.

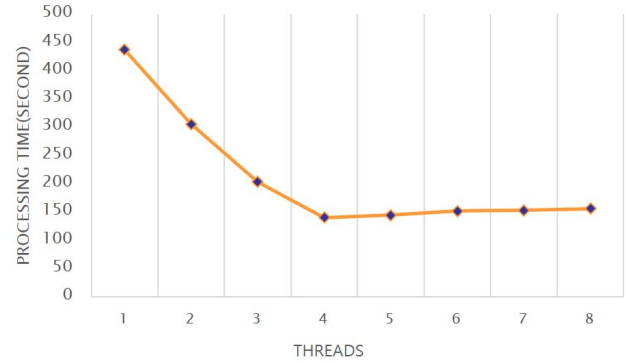


Fig. 4. Experiment Result of Distributed Deep Learning for entire View

Fig. 4 shows the number of threads and processing time for PETS data with 2GB. As the number of thread increases, the time taken for object detection decreases up to 4 by 3.5 times as in Fig. 3. However, in both experiments, with more than five threads, the speed up is slow down. That is, we have the maximum speed up for two threads on each node when using two nodes.

## V. CONCLUSION

In this paper, in this paper, we have presented a new distributed processing platform which provides scalability and provisioning for the deep learning inference using DeCUVE. It can easily build a cluster by assigning available nodes in cloud environment, and automatically provisioning various kind of physical nodes such as GPUs. We have shown that deep learning training inference can be executed fast for object detection application. Experiment results have shown an increase in the number of thread can bring about further improvement in speed. Use of Deuce for distributed deep learning is effective for deep learning study. We expect to save money and time in building distributed deep learning environment when using DeCUVE. Our distributed processing

environment for deep learning will be very helpful for artificial intelligence in real life.

Our future work is to construct an environment for real-time image processing with kind of processing elements such as multicores and GPU. Many users are unable to use high-end system because expensive. DeCUVE can solve this problem.

#### ACKNOWLEDGMENT

This work was supported by the Brain Korea 21 Plus Project in 2017 and this research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2017R1D1A1B03035461).

#### REFERENCES

- [1] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *Nature* 521.7553 (2015): 436-444.
- [2] Dean, Jeffrey, et al. "Large scale distributed deep networks." *Advances in neural information processing systems*. 2012.
- [3] Wang, Wei, et al. "Deep learning at scale and at ease." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 12.4s (2016): 69.
- [4] Abadi, Martin, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." *arXiv preprint arXiv:1603.04467* (2016).
- [5] "GPU-Based Deep Learning Inference: A Performance and Power Analysis." *Nvidia Whitepaper* (2015).
- [6] In-Yong Jung. "Cloud unified virtual environment for distributed parallel computing." *Doctor thesis. Korea University* (2015).
- [7] Dua, Rajdeep, A. Reddy Raja, and Dharmesh Kakadia. "Virtualization vs containerization to support paas." *Cloud Engineering (IC2E)*, 2014 IEEE International Conference on. IEEE, 2014.
- [8] Felter, Wes, et al. "An updated performance comparison of virtual machines and linux containers." *Performance Analysis of Systems and Software (ISPASS)*, 2015 IEEE International Symposium On. IEEE, 2015.
- [9] Liu, Wei, et al. "Ssd: Single shot multibox detector." *European conference on computer vision*. Springer, Cham, 2016.