An Edge Computing Marketplace for Distributed Machine Learning

Susham Yerabolu*
Samuel Gomena*
Portland State University
Computer Science Department
Portland, OR, USA
{yerabolu,gomenas}@pdx.edu

Ehsan Aryafar
Portland State University
Computer Science Department
Portland, OR, USA
earyafar@pdx.edu

Carlee Joe-Wong
Carnegie Mellon University
Electrical and Computer Engineering
Silicon Valley, CA, USA
cjoewong@andrew.cmu.edu

ABSTRACT

There is an increasing demand among machine learning researchers for powerful computational resources to train their machine learning models. In order to train these models in a reasonable amount of time, the training is often distributed among multiple machines; yet paying for such machines is costly. DeepMarket attempts to reduce these costs by creating a marketplace that integrates multiple computational resources over a distributed tensorflow framework. Instead of requiring users to rent expensive resources from a third party cloud provider, DeepMarket will allow users to lend their computing resources to each other when they are available. Such a marketplace, however, requires a credit mechanism that ensures users receive resources in proportion to the resources they lend to others. Moreover, DeepMarket must respect users' needs to use their own resources and the resulting limits on when resources can be lent to others. This Demo will introduce the audience to PLUTO: DeepMarket's intuitive graphical user interface. The audience will be able to see how PLUTO in coordination with DeepMarket servers tracks the performance of each user's training jobs, matches jobs to resources made available by other users, and tracks the resulting credits that regulate the exchange of resources.

CCS CONCEPTS

• **Networks** → *Network architectures*; • **Computing methodologies** → *Machine learning*.

KEYWORDS

TensorFlow, Marketplace Design, Network Economics

ACM Reference Format:
Susham Yerabolu, Samuel Gomena, Ehsan Aryafar, and Carlee Joe-Wong. 2019. An Edge Computing Marketplace for Distributed Machine Learning. In *SIGCOMM '19: ACM SIGCOMM 2019 Conference (SIGCOMM Posters and Demos '19), August 19–23, 2019, Beijing, China.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3342280.3342299

*Both authors contributed equally to this research.

SIGCOMM Posters and Demos '19, August 19–23, 2019, Beijing, China

ACM ISBN 978-1-4503-6886-5/19/08…$15.00
https://doi.org/10.1145/3342280.3342299

1 INTRODUCTION

As deep learning proves its usefulness in an ever greater number of applications, there is a rise in demand for faster and cheaper computational resources to manage and train ever more complex learning-based models. Purchasing machines outright, however, can require significant upfront investment that is not justified by the intermittent use that many researchers require. Renting resources from cloud providers such as Amazon AWS or Microsoft Azure is also expensive since cloud providers need to construct and maintain cloud (*i.e.,* data center) infrastructure with significant operation and maintenance cost. Several recent works have attempted to intelligently exploit various types of cloud pricing to reduce their cost [5, 8, 9], but they are still fundamentally dependent on cloud providers' offering low-cost options.

One possible solution to reduce these costs is to introduce a *marketplace* for computing resources in which users can lend each other resources when they are idle, similar to Uber's or Airbnb's sharing platforms. Such a marketplace reduces upfront investment costs by allowing users to purchase fewer resources outright and supplement them with others' machines, and eliminates the need to pay a cloud provider. While prior work on volunteer computing has proposed similar ideas [6], such a computing marketplace is particularly appropriate for training deep learning models, which can easily be done in a distributed manner. Algorithms for training these models can be easily adapted to run on heterogeneous resources, e.g., both CPU and GPU servers, thus allowing a wide range of users to participate in the computing marketplace. As edge computing paradigms, which aim to exploit computational resources at the edge of the network, become popular, such edge devices may also be incorporated into a computing marketplace.

This demo introduces DeepMarket[1], an open-source application that enables a computing marketplace for deep learning. Users that lend resources on DeepMarket receive credits for doing so, which can be used to rent resources from others in the future. In this way, researchers are incentivized to contribute their idle resources to DeepMarket; and DeepMarket automatically matches available resources to pending deep learning jobs, seamlessly executing the job over dispersed, heterogeneous machines. The goal of this demo is to introduce DeepMarket with its three main parts: the user interface PLUTO; the service module, which tracks job and resource status; and the Executor module, which executes distributed training jobs. The demo will particularly showcase PLUTO's ability to

[1]We have provided additional information about our system architecture in a preliminary workshop paper [7].

track the status of users' jobs and resources, protect non-idle resources from being cannibalized by DeepMarket, and provide an accounting mechanism for users' credits. We also discuss some of the key system-level challenges in building our architecture, including containerization [2] of the software stack, scalability and robustness of the services, and system security and privacy.

2 SYSTEM OVERVIEW

The DeepMarket architecture consists of 3 main components as shown in Figure 1. PLUTO is a simple and intuitive graphical user interface developed using PyQt5. It allows users to lend their spare computational resources and list them on DeepMarket marketplace, view their submitted jobs' status, and submit jobs to the DeepMarket marketplace (i.e., borrow computational resources from others and run their jobs on them). PLUTO resides on users' machines.

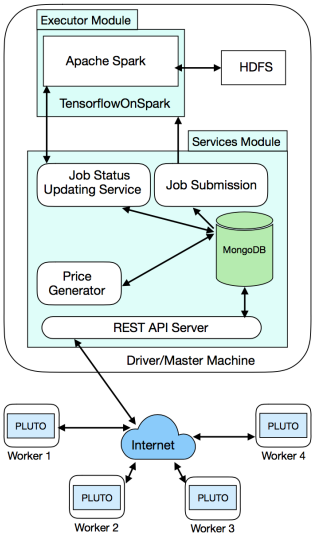

Figure 1: DeepMarket architecture. Any user machine with Internet connectivity can be added to the DeepMarket network. When the machine is used for a computing task, DeepMarket will create a container inside the user's machine. This increases the architecture scalability and provides an additional layer of security and privacy.

The other two modules (the services module and the executor module) reside on our servers at Portland State University. In the Services Module, we have developed a secure REST API Service that saves the job submission and available resources information submitted via PLUTO. We have also developed a pricing algorithm to generate prices for each time slot based on the previous jobs' execution times, current available resources and frequency of resource utilization; these prices would be designed to ensure that the "value" of users' contributed resources matches the resources they borrow from others. The Executor Module is responsible for data file management, scheduling of resources, and execution of distributed tensorflow programs. We use a combination of Apache Spark [1], TensorFlowOnSpark [4], and HDFS (Hadoop Distributed File System) [3] to achieve these goals. HDFS is used to provide large

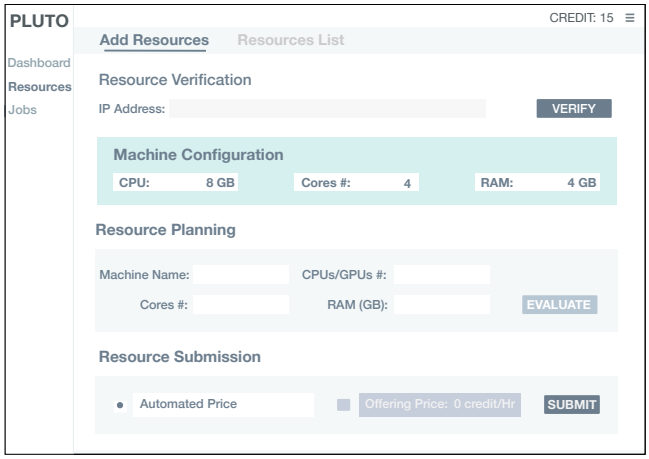

Figure 2: PLUTO - Resource Enlistment Tab

amounts of data to each worker machine in a fault-tolerant manner. Apache Spark along with TensorflowOnSpark enable distributed deep learning on a cluster of GPU and CPUs, and support all types of TensorFlow programs, including asynchronous/synchronous training and inferencing.

3 DEMO OVERVIEW

This demo will show the complete end-to-end design and implementation of DeepMarket on both the users and server. We will bring three laptops with pre-installed PLUTO to represent user machines. The audience will be able to experiment with the application, create an account on DeepMarket, list the computer as a resource on DeepMarket, or submit distributed ML jobs and retrieve the results.

Through PLUTO's resources tab (Figure 2), users can lend their current machines and add them to the pool of resources at DeepMarket by providing the machines' IP addresses. A user can specify the fraction of resources that she is willing to lend, *e.g.,* a user may only lend half of the total cores or RAM, and use the remaining cores and RAM for local use. In our current implementation, the credit that a user earns by lending her machines is automatically generated by our system. Our future updates to PLUTO would enable a user to specify a minimum lending price.

PLUTO's jobs tab (omitted due to page limitations) provides several functionalities: (i) it displays the current price of running jobs per CPU/GPU/RAM/Disk unit (*e.g.,* 1 Credit/Hr for 1GB of RAM). The prices are denoted over four six-hour time slots, beginning at 12 AM at our server location. The variability of prices at different time slots allows users with low credits to schedule their jobs at cheaper times. (ii) The "Jobs Tab" also provides an interface for users to submit jobs and run distributed tensorflow programs. In addition to the desired job running time, the user specifies the desired number of workers, cores and RAM per worker, and the "HDFS path" to the source files and data files. These are all needed when submitting a machine learning job.

ACKNOWLEDGMENTS

This research was supported in part by NSF grant CNS-1751075.

REFERENCES

[1] [n.d.]. Apache Spark. https://spark.apache.org/

[2] [n.d.]. Container: A Standardized Unit of Software. https://www.docker.com/resources/what-container

[3] [n.d.]. HDFS: Hadoop Distributed File System. https://hadoop.apache.org/

[4] [n.d.]. TensorFlowOnSpark. https://github.com/yahoo/TensorFlowOnSpark

[5] M. Khodak, L. Zheng, A. S. Lan, C. Joe-Wong, and M. Chiang. 2018. Learning Cloud Dynamics to Optimize Spot Instance Bidding Strategies. In *Proceedings of IEEE INFOCOM.*

[6] E. Lavoie and L. Hendren. 2018. Personal Volunteer Computing. https://arxiv.org/pdf/1804.01482.pdf.

[7] S. Yerabolu, S. Kim, S. Gomena, X. Li, R. Patel, S. Bhise, and E. Aryafar. 2019. DeepMarket: An Edge Computing Marketplace with Distributed TensorFlow Execution Capability. In *Proceedings of IEEE ECOnomics of Fog, Edge, and Cloud Computing (ECOFEC) Workshop, Paris, France.*

[8] L. Zheng, C. Joe-Wong, C. Brinton, C. W. Tanand S. Ha, and M. Chiang. 2016. On the viability of a cloud virtual service provider. *ACM SIGMETRICS Performance Evaluation Review* 44, 1 (2016), 235–248.

[9] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, and X. Wang. 2015. How to bid the cloud. In *ACM SIGCOMM Computer Communication Review*, Vol. 45. 71–84.